SparseEMG: Computational Design of Sparse EMG Layouts for Sensing Gestures

Anand Kumar
University of Calgary
Calgary, Canada
anand.kumar1@ucalgary.ca

Antony Albert Raj Irudayaraj
University of Calgary
Calgary, Canada
antony.irudayaraj@ucalgary.ca

Ishita Chandra
University of Calgary
Calgary, Canada
ishita.chandra@ucalgary.ca

Adwait Sharma
University of Bath
Bath, United Kingdom
as5339@bath.ac.uk

Aditya Shekhar Nittala
University of Calgary
Calgary, Canada
anittala@ucalgary.ca

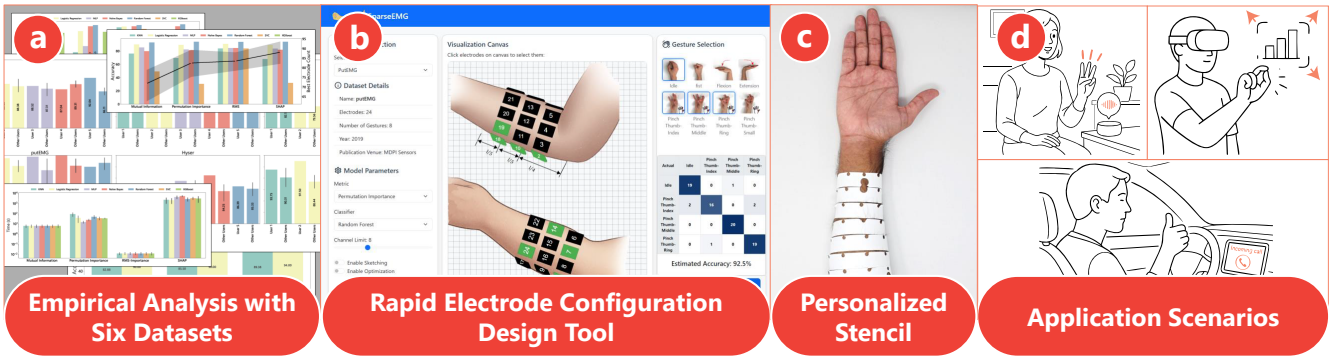


Figure 1: SparseEMG is a computational tool designed for rapid prototyping of sparse EMG electrode layouts for gesture recognition. Overview of our method (a) Empirical analysis across six datasets to identify optimal electrode selection strategies, (b) design tool that recommends sparse electrode configurations with high-level input, (c) and generates personalized stencils for user-specific electrode placement, and (d) evaluation across diverse real-world scenarios to demonstrate the tool's applicability.

ABSTRACT

Gesture recognition with electromyography (EMG) is a complex problem influenced by gesture sets, electrode count and placement, and machine learning parameters (e.g., features, classifiers). Most existing toolkits focus on streamlining model development but overlook the impact of electrode selection on classification accuracy. In this work, we present the first data-driven analysis of how electrode selection and classifier choice affect both accuracy and sparsity. Through a systematic evaluation of 28 combinations (4 selection schemes, 7 classifiers), across six datasets, we identify an approach that minimizes electrode count without compromising accuracy. The results show that Permutation Importance (selection scheme) with Random Forest (classifier) reduces the number of electrodes by 53.5%. Based on these findings, we introduce SparseEMG, a design tool that generates sparse electrode layouts based on user-selected gesture sets, electrode constraints, and ML parameters while also predicting classification performance. SparseEMG supports 50+ unique gestures and is validated in three real-world applications using different hardware setups. Results from our multi-dataset evaluation show that the layouts generated from the SparseEMG design tool are transferable across users with only minimal variation in gesture recognition performance.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interaction techniques**; **Gestural input**.

KEYWORDS

Electromyography, Gesture Recognition, Design tools

ACM Reference Format:
Anand Kumar, Antony Albert Raj Irudayaraj, Ishita Chandra, Adwait Sharma, and Aditya Shekhar Nittala. 2025. SparseEMG: Computational Design of Sparse EMG Layouts for Sensing Gestures. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28-October 1, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3746059.3747614

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '25, September 28-October 1, 2025, Busan, Republic of Korea

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-2037-6/2025/09…$15.00
https://doi.org/10.1145/3746059.3747614

1 INTRODUCTION

Surface Electromyography (sEMG) is a technique to record muscular activity by measuring the electrical field generated by the contractions of muscle fibres through electrodes placed on the skin. While extensively used in neuroscience, biomedical engineering, prosthetic control and rehabilitation studies, this technique is also used in HCI for gesture recognition. Seminal work by Saponas et al. [78–80] and Constanza et al. [10, 11] demonstrated the use of sEMG signals for gesture recognition, coining the term 'muscle-computer interfaces.' Subsequently, plenty of follow-up work has studied gesture recognition with sEMG [22, 37, 40], and even consumer-grade EMG devices are being actively researched [1].

However, building EMG systems from scratch is complex and requires expertise in numerous domains, such as sensing technology, signal processing, machine learning, and interaction design. Sensing with EMG signals also requires placing the electrodes at precise locations for high-quality gesture recognition. This placement of electrodes becomes even more challenging when designing an optimal layout for recognizing a given set of gestures.

Recent research has contributed open-source toolkits to facilitate the development of myoelectric control systems for non-domain experts. These toolkits abstract many challenges, including data processing, hardware interfacing, feature extraction/selection, classification, postprocessing, and evaluation [22, 37]. While impressive, these toolkits focus on developing interactive ML systems for a given set of electrodes and gestures, i.e. they assume that a given set of electrodes are appropriately placed and then allow users to train a set of gestures. *They do not focus on finding a sparse layout of electrodes for sensing a given set of gestures.*

Recent research in HCI has proposed the use of optimal sensor placement for sensing fine-grained microgestures [85]. There have been similar efforts which have shown that identifying the optimal location for sensor placement can boost the sensing accuracy [26, 51, 87] or it can reduce the number of sensors required for detecting a given set of microgestures [85]. Similarly, Lin et al. [51] suggested that the minimum accuracy of 70.8% can be increased to 95.8% for an identified optimal location. In a broader context, a large body of work has used compressed sensing and data-driven techniques to demonstrate that human body pose can be reconstructed by a significantly reduced number of sensors [3, 21, 28, 82]. However, all these efforts have been focused on sensors such as IMUs [28, 85], strain gauges [51], or optical markers [82] and do not generalize to sEMG sensing.

Biomedical and physical sciences research has used data-driven and signal-processing techniques to analyse sEMG signals for identifying the innervation zones [6]. This was typically done by placing an extensive array of sEMG electrodes on the target muscle group and collecting the signals across the array. There have been similar efforts to identify the optimal placement of electrodes [31, 59, 74, 94]. However, these focus on rehabilitation, prosthesis control or for determining the major muscle groups that are responsible for specific muscular contractions (e.g. Flexor Carpi Radialis flexes abducts the hand at the wrist [95]). These techniques do not generalize to gestures because each gesture typically involves the contraction of multiple muscle groups, and no established mapping exists between a given gesture and a muscle group.

[1] https://tech.facebook.com/reality-labs/2021/3/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computing-platform/

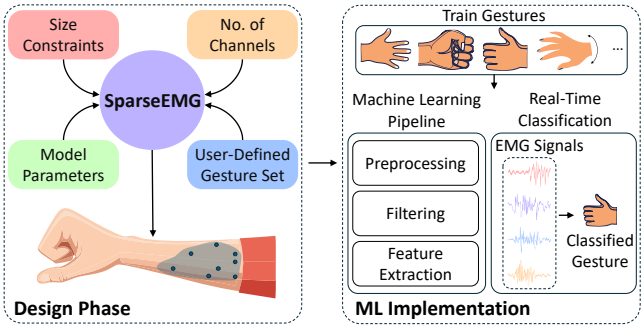


Figure 2: Existing EMG toolkits in HCI majorly focus on creating a machine learning pipeline for recognizing predefined gestures and easing this process [22, 37] (which falls in the ML implementation phase of EMG gesture recognition). In contrast to these approaches, our design tool helps designers to rapidly select sparse EMG layouts for a desired set of gestures and other constraints specific to EMG and machine learning (which usually occurs during the gesture and device design phase).

This paper addresses the following research questions, which, to the best of our knowledge, have not been explored in prior work which focused on EMG gesture recognition.

- How can we objectively quantify the involvement of specific electrodes for a given set of gestures?
- For a given set of constraints (number of channels, given gesture set, ML parameters), can we identify an electrode layout that can balance sparsity without compromising the recognition accuracy?
- How well do the solutions generated by our tool scale across widely available EMG hardware platforms, and for practical real-world scenarios?

Exploring these research questions offers several benefits. Firstly, by determining the optimal number of electrodes needed for a specific set of gestures, we can simplify the hardware acquisition process and reduce costs. For instance, decreasing the number of electrodes from 16 to 6 can lower the price from several thousand dollars to just a few hundred by transitioning from professional-grade acquisition setups to low-cost Arduino-based EMG systems. Secondly, in an Arduino-based setup, having fewer channels allows for a higher frame rate, which can enhance performance. Finally, investigating these questions facilitates the creation of a computational tool that simplifies the complexities of machine learning and EMG. This empowers users to generate gesture-specific layouts with just a few clicks.

Overall, our paper makes the following contributions:

(1) **Series of Empirical Analysis**: We contribute a systematic data-driven analysis to understand how electrode selection and sparisty affect gesture recognition. Through a systematic evaluation of 28 combinations (4 selection schemes, 7

classifiers) across six datasets to understand the best combination of electrode-selection metrics and ML classifiers for a given set of electrodes. (Figure 1(a))

(2) **Computational Design Tool**: We contribute a computational design tool that allows designers to identify an optimal set of electrodes for a given gesture set and a desired number of electrodes. The tool abstracts the designer from the lower-level technical aspects such as the anatomical relation between gestures and muscles, machine learning classifiers and sEMG signal characteristics. (Figure 1(b))

(3) **Application Scenarios:** We contribute application scenarios from diverse and representative domains to illustrate how designers and engineers can leverage the potential of our approach for concrete design tasks. Our application examples demonstrate that the computational approach is scalable to different hardware setups. (Figure 1(d))

2 RELATED WORK

Our work falls at the intersection of EMG sensing, computational design tools and toolkits for gesture sensing.

2.1 EMG Gesture Recognition in HCI

The success of using electromyography (EMG) for prosthesis control led to its exploration by researchers in Human-Computer Interaction (HCI) in the early 2000s [10, 11, 78–80]. Prior to this, surface EMG devices were expensive, tethered, and mainly used for medical purposes. The introduction of commercially available wearable devices, such as the Myo Armband in 2014 [75], opened new possibilities for the broader research community to explore this technology for novel hands-free interactions.

Since then, myoelectric control has been applied to various general-purpose applications, including piano augmentation [36], drone control [14], gaming [59, 90], hands-free input [39], and mixed-reality interactions [96]. At the same time, the HCI community has made significant contributions, applying their expertise to enhance prosthetics-related research through training tools [1, 91], approaches to alleviate phantom limb pain [71], and improved prosthesis designs [86]. There is also increasing attention to EMG sensing from ML and computer vision communities [40, 53, 77]. However, these works focus on user-independent models, pose estimation from a commercial EMG hardware band (e.g. MyoBand) and not on generating spare EMG layouts for a given set of gestures and other user-defined constraints. Despite the growing interest and adoption of this technology, both commercially [30] and academically, the widespread use of myoelectric control for general applications remains limited [23]. Recent work by Eddy et al. [23] supports the notion that designing gesture recognition systems with EMG requires EMG-specific domain expertise, machine learning and interaction design. While there have been efforts in building toolkits to ease this process [22, 37], these focus on building ML pipelines and don't focus on the interaction design of gestures and generating a sparse EMG layout for a set of user-selected gestures. Our toolkit fills this gap in EMG toolkit research by abstracting the designers from EMG-specific domain expertise and instead focusing on gesture design for specific application scenarios.

2.2 EMG in Biomedical and Physical Sciences

EMG sensing has been extensively used in biomedical engineering and physical science research communities. Very early work in biomedical engineering used sEMG for understanding the mapping between limb movements and the muscles [6]. More recently, using sEMG and machine learning for prosthesis control has received more attention in the biomedical research community, contributing datasets [5, 34], classifier-finetuning [15, 35, 62] and open-source toolkits [20, 22, 56]. BioPatRec is one of the earliest efforts in prosthesis control research that provides a modular, customizable platform that abstracts the practitioners from domain-specific knowledge in signal processing, feature selection, extraction, classification and real-time control [64]. Follow-up works have investigated the performance of various classifiers and contributed fine-tuned classifier topologies for improving the accuracy for prosthetic control [65, 72]. Similar efforts have contributed open-source datasets and benchmarking toolkits for analyzing the sEMG signals [32, 41, 81]. There has been work on optimizing the number of sEMG channels for gesture recognition [52, 73]. However, these works focus on a specific dataset and for a pre-defined set of gestures and do not support the interactive exploration of variations in gesture set and user-defined electrode constraints. A more detailed review of feature extraction techniques, deep learning approaches for sEMG in the context of prosthetic control and clinical evaluation of neuromusculoskeletal systems can be found here [88]. While there is vast literature in biomedical and physical sciences research on using sEMG signals, they primarily focus on improving classification accuracy or incorporating deep-learning approaches or hardware/software toolkits for analyzing sEMG signals. To the best of our knowledge, we are yet to see a systematic analysis and design tool that enables interactive exploration of the various parameters that can influence sEMG signals.

2.3 Computational Tools and Toolkits in HCI

One solution to improve accessibility for complex areas of research is through the development of toolkits. Toolkits provide easy access to complex algorithms, enable fast prototyping of software and hardware interfaces, and/or enable creative exploration of design spaces [48]. Research in HCI has contributed toolkits [37, 58], fabrication techniques [61, 66] and design tools for various applications, such as for electrical muscle stimulation [38, 46], physiological sensing [60], actuation [57, 97] and haptics [16, 24, 58, 89, 92]. Similarly, several design tools and elicitation studies have been presented in the HCI literature for the design of various gestures [12, 83, 84]. These include works that allow the designer to compare a gesture with a corpus of everyday activity data for false positive testing [4, 45]. EventHurdle [43], M.Gesture [42], and Mogeste [67] enable users to compose custom gestures on mobile devices. Gesture Coder [54] is a tool to help developers add multi-touch gestures by demonstrating them on a tablet's touchscreen. While existing frameworks and platforms enable prototyping and debugging various classifiers and implementing custom machine learning pipelines [27, 68], they are targeted at programmers and do not consider aspects of interaction design. On the other hand, recent technological advances have enabled novice users to train and classify custom ML models without the need for programming

Related Work	Context	Gesture Variability	Hardware Independent	Optimized Layouts	Dataset Scalability
libEMG [22]	Toolkit for EMG gesture recognition	✗	✓	✗	✓.
EMBody[37]	Toolkit for EMG gesture recognition	✗	✗	✗	✗.
SparseIMU [85]	Tool for Sparse IMU generation	✓	✓	✓	✗.
Nittala et al. [59]	Tool for optimized EMG layouts	✗	✓	✓	✗.
Li et al. [50]	Optimizing the feature set and electrode configuration for EMG recognition	✗	✗	✓	✗.
BioPatRec [64]	Software for development of algorithms in prosthetic control	✗	✗	✗	✓.
Luis et al. [69]	Analyzes the impact of electrode configuration to reduce channel count	✗	✗	✓	✗.
Prakash et al. [73]	Optimizing electrode configuration for wrist-worn hand gesture recognition.	✗	✗	✓	✗.
Fang et al. [25]	Electrode configuration framework to improve the operational effectiveness	✓	✗	✓	✗.
Zhao et al. [98]	Electrode optimization for prosthetics	✗	✗	✓	✗.
Chaplot et al. [9]	Electrode optimization with five placement configurations	✗	✗	✓	✗.
This work	Interactive exploration of gestures, electrode configurations and accuracy.	✓	✓	✓	✓.

Table 1: Comparison of related work across HCI, biomedical and prosthesis control that has contributed to the analysis of sEMG signals and the design of toolkits.

expertise [19]. However, these majorly address image or audio classification problems. SparseIMU [85] contributes a design tool for generating sparse IMU layouts for gesture recognition. However, it does not scale to EMG sensing which is a completely different sensing mechanism. Secondly, the number of IMUs in SparseIMU is 17, while EMG arrays typically involve a much denser array of electrodes ranging from 10 to 200. Finally, the placement of EMG is more complex (electrodes need to be placed precisely with a target muscle). In contrast, IMU placement is simpler and involves placing the sensors on the finger phalanges. Our main goal behind this work is to use machine learning as a design material [18] and enable designers to prototype custom microgestures without the need for having expertise in ML and programming.

2.4 How does this work differ from prior EMG toolkits in HCI?

LibEMG [22] and EMBody [37] are EMG toolkits that enable novice users to develop EMG gesture recognition applications. Although an excellent starting point, EMBody's [37] contribution stands primarily in its open-access hardware design and goal of enabling HCI researchers to explore EMG as an input modality. EMBody's [37] limitation is that it was created with a predefined and rigid myoelectric control system. For example, by default, EMBody only supports a single classifier (SVM) and feature (root mean square). LibEMG [22] aims to address these limitations of EMBody [37] by providing users access to a richer set for feature selection and ML model parameters. The authors also tested this on multiple hardware platforms. However, these are limited to commercial EMG bands that do not offer the choice of optimizing the electrode placements or channels.

In contrast with these exemplary prior works, we aim to understand how electrode selection schemes and classifiers can influence the accuracy of a given set of gestures. While the existing toolkits focus on easing the process of building ML models for a desired set of gestures and number of electrodes, our approach aims to identify a set of electrodes balancing sparsity and accuracy for a given set of gestures and desired set of electrodes.

3 DESIGN GOALS

Many toolkits and datasets exist for EMG gesture recognition across diverse disciplines, including biomedical engineering, prosthesis controls, HCI, and rehabilitation. Hence, the toolkits are often designed for specific tasks in a given discipline. For example, BioPatRec [64] is one of the very early toolkits designed in Matlab that enables researchers to interactively test and validate various ML features and algorithms suitable for prosthesis control. On the other hand, toolkits in HCI, such as LibEMG [22] and EMBody [37] focus on enabling novice users to quickly train an EMG recognition model with a custom gesture set defined by users.

3.1 Gesture Variability

While existing EMG toolkits enable novice practitioners to create machine learning models for sEMG gesture recognition, they do not adequately support interaction designers in quickly prototyping and assessing the accuracy of specific gestures. To aid in gesture design prototyping, a primary goal of our design tool is to allow designers to choose from a set of gestures and to select and refine a subset of those gestures for effective recognition. This is a crucial requirement because, for a given set of n gestures, there are $2^n - 1$ subsets possible and being able to select and deselect gestures for given application scenarios rapidly helps in rapid design iterations. The design tool should generate a sparse EMG layout based on user-defined electrode constraints for a given set of gestures.

3.2 Supporting User-Defined Electrode Placement

One of our primary design objectives is to support user-defined electrode placement. This feature enables researchers and practitioners to customize sEMG sensor layouts according to their specific experimental needs, anatomical constraints, or application requirements. Unlike rigid, predefined electrode configurations, our approach allows designers to specify their preferred electrode locations while ensuring that the resulting layout maintains both signal quality and spatial efficiency.

3.3 Hardware Independence

Another fundamental design objective which we consider is to ensure hardware independence, allowing solutions to scale seamlessly across various sEMG acquisition systems (e.g. EMG bands like Myo or traditional gel-electrode based sensing systems) available to researchers and practitioners. Given the diversity of EMG hardware, including differences in amplifier sensitivity, sampling rates, and electrode interfaces, the solutions generated should scale across multiple hardware configurations without fine-tuning hardware parameters or other technical modifications.

3.4 Scalability Across Datasets

EMG is one of the most active research areas explored by several research communities. Each paper has a new dataset with different gesture sets, hardware, model parameters, and participants. One of the design goals is to ensure that the tool can import a wide range of datasets, understand their semantics and gestures, and then allow users to explore all possible combinations with that dataset interactively. Additionally, the tool should also be able to allow for generating an optimal solution for a given set of parameters (e.g. gesture set, number of electrodes, specific muscles to choose from, ML parameters etc) that the user can pick from a chosen dataset.

4 DATA-DRIVEN ANALYSIS FOR IDENTIFYING SPARSE CONFIGURATIONS

Electrode placement is crucial for robust gesture recognition, as different gestures activate distinct muscle groups [8, 29]. While uniform distribution around the forearm is common [17, 34, 70], it often leads to suboptimal performance. Identifying the optimal configuration is challenging, especially with high-density arrays, due to combinatorial complexity $\binom{n}{k}$. To address this, we treat electrode selection as a feature subset selection problem, aiming to efficiently identify a minimal set of electrodes that maximizes accuracy while maintaining a sparse, practical layout.

The goal of the data analysis is to identify the optimal combination of electrode selection schemes (ESS) and classifiers (CL) that balance accuracy, sparsity, and computational efficiency. This informs our design tool in generating sparse electrode layouts with high classification accuracy in real time.

4.1 Method

We systematically analyze electrode selection schemes and classifier combinations to identify the minimal subset of electrodes and evaluate their generalizability across six open-source datasets. Before diving into the analysis, we present the underlying methodology by introducing the electrode selection schemes, datasets, and the preprocessing pipeline.

4.1.1 Experimental Datasets. After conducting a literature survey based on three criteria: (1) forearm muscle coverage, (2) diversity of HCI-relevant gestures, and (3) open-access availability, we identified six relevant datasets, summarized in Table 3. For more details, readers can refer to the respective publications.

We begin our analysis with the CSL-HDEMG dataset [2], published at CHI 2015, for three reasons: (1) it includes 192 electrodes, offering extensive forearm coverage; (2) it aligns with HCI applications through single-handed gestures; and (3) it features 27 diverse hand gestures, grouped into tap, bend, and multi-finger sets.

The dataset was recorded from five users (five sessions/user). Each gesture was repeated 10 times per session, resulting in a total of 27 (# gestures)×10 (# trials)×5 (# users)×5 (# sessions) = 6,750 gesture trials. Each gesture was performed over a 3-second interval, and the signals were recorded at a sampling rate of 2048 Hz.

To empirically assess the generalizability of the electrode selection schemes, we also evaluate their performance across five additional datasets in Section 4.5.

4.1.2 Electrode Selection Schemes. The feature subset selection problem aims to identify the most informative set of features, in this case, sEMG channels, that maximizes model accuracy. Broadly, subset selection methods fall into two categories: filter-based methods (model-independent, relying on data statistics) and wrapper-based methods (model-dependent, evaluating feature impact on model performance) [44]. In this paper, we evaluate four methods for selecting the most informative electrodes:

(1) **Mutual Information (MI)**: Uses entropy-based calculations to measure the amount of information shared between features and target variables. By selecting electrodes based on their informational contribution, MI helps minimize redundant or irrelevant electrode placements. [25, 47, 93].
(2) **Permutation Importance (PI)**: It provides insight into the importance of each electrode by showing how much performance is affected when an electrode is removed. This can help pinpoint the minimum number of electrodes necessary to maintain high accuracy while ensuring computational efficiency [7].
(3) **Root Mean Square-based Importance (RMS-I)**: Quantifies muscular activity using RMS, which highly correlates with muscle force. [13].
(4) **SHapley Additive exPlanations (SHAP)**: Uses game theory to assign importance scores based on feature contributions to predictions. It offers a robust explanation of feature importance and ensures that the most influential electrodes are selected, helping build a sparse yet highly accurate EMG layout [55].

4.1.3 Classifiers. Rapid prototyping in HCI demands tools that balance speed, interpretability, and practical deployability. To systematically test our subset selection methods, we evaluate them in combination with six classifiers that are commonly used and well-established in EMG gesture recognition workflows: Support Vector Classifier (SVC), Logistic Regression (LR), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB), XGBoost (XGB), and Random

Forest (RF) with $max_depth = 30$. All classifiers are used with their default hyperparameters. Additionally, we train a Multi-Layer Perceptron (MLP) pipeline as described in [52] to explore the performance of neural networks in this context. We evaluated every electrode selection scheme and classifier combination, resulting in a total of 4 (#ESS) × 7 (#CL) = 28 combinations.

Each gesture trial is divided into three non-overlapping windows of equal length. For each window, we compute the Root Mean Square (RMS) value for each channel and concatenate the results, producing an $n \times 3$ feature vector per trial, where n is the number of channels. This feature vector is used as input to each classifier.

Other Reduction Strategies and Classification Techniques. SparseEMG prioritizes electrode reduction over general dimensionality reduction. Our primary objective is to identify and eliminate non-essential electrodes while preserving classification accuracy. Techniques like PCA and LDA are widely used for reducing data dimensions, but they don't focus on electrode reduction, which is crucial for optimizing hardware efficiency. Similarly, methods like LASSO and autoencoders help with dimensionality reduction but don't directly address which electrodes can be discarded without sacrificing performance.

We included a simple neural network (MLP) in our evaluation to investigate the potential of deep learning. However, we deliberately excluded more complex architectures such as CNNs and LSTMs due to their high computational overhead, reliance on larger datasets, and reduced interpretability. SparseEMG emphasizes speed and interpretability, making simpler models like Random Forest and MLP more suitable for rapid prototyping and exploratory analysis.

While this paper focuses on applying existing machine learning techniques to electrode reduction, future work could explore the development of novel methods tailored to this challenge. Though beyond the scope of this study, these advancements represent an exciting direction for future research.

4.2 Evaluating Selection Scheme and Classifier Combination

RQ1: What is the minimum number of electrodes required for each ESS-CL combination to achieve maximum accuracy, and at what point does the accuracy plateau for each combination?

Previous works [2, 69] have shown that not all electrodes contribute significantly to achieving maximum accuracy, and beyond a certain point, adding more electrodes results in minimal to no improvement. We evaluate each ESS-CL combination to determine the minimum number of electrodes required to reach its maximum possible accuracy, i.e., the point at which accuracy plateaus.

For this analysis, we included all the gestures from the dataset. Starting from the top two ranked electrodes, we progressively added more electrodes, up to a maximum of 100. This incremental strategy allowed us to determine the minimum number of electrodes required to achieve near-maximal classification performance. We included all five users from the dataset to evaluate generalization across different individuals. To account for variability across trials and sessions, we use a total of 20 trials across sessions per user, with 27 gestures, resulting in 540 trials per user. A 4-fold cross-validation is performed for each electrode selection scheme and classifier combination. During each cross-validation fold, the dataset is stratified to ensure a balanced distribution of gestures across the training and testing sets. In each fold, the test set contains 135 trials (25%), with the remaining trials used for training. We repeat the entire cross-validation process across five randomly selected users from the dataset to evaluate the generalizability of our findings across different individuals.

4.2.1 ***Findings.*** : The results in Table 2 highlight the trade-off between the number of selected electrodes and the classification accuracy of different subset selection methods and classifiers. The PI + RF combination achieves the highest accuracy (94.04%) with a relatively low electrode count (76), making it one of the most efficient combinations. Similarly, MI + RF (92.81% accuracy, 78 electrodes) balances performance and efficiency, while MI + LR prioritizes minimal electrodes (74) at the cost of reduced accuracy (89.93%). In contrast, methods like RMS-I and SHAP often demand significantly more electrodes for marginal gains. For example, SHAP + RF achieves only a 0.04% accuracy increase over PI + RF (94.00% vs. 94.04%) but requires 94.40 electrodes, 18 more than PI + RF. Moreover, RMS-I + MLP achieves the highest overall accuracy (95.20%) but at the expense of practicality, using 94.40 electrodes.

Classifier	MI		PI		RMS-I		SHAP	
	# Elec.	Acc.	# Elec.	Acc.	# Elec.	Acc.	# Elec.	Acc.
KNN	68	76.00	72	69.41	82	83.81	75	67.89
LR	74	89.93	91	89.33	82	90.78	92	89.07
MLP	75	85.26	89	82.33	94	84.41	95	83.63
NB	85	79.78	93	81.33	71	84.52	88	81.93
RF	78	92.81	76	94.04	86	93.78	94	94.00
SVC	48	49.19	90	30.41	85	83.74	96	32.04
XGB	73	84.30	66	86.07	85	84.67	76	86.30

Table 2: Comparison of subset selection methods and classifiers for gesture recognition. For each method, the average number of selected electrodes (rounded off to the nearest integer) (# Elec.) and best classification accuracy (Acc.) are reported.

4.3 Optimizing Electrode Count for Practical Usability

RQ2: How well do ESS-CL combinations perform when constrained to a practical electrode limit (e.g., 20 electrodes), and which combinations offer the best balance between accuracy and real-world hardware feasibility?

Real-world wearable systems face tight hardware constraints, typically supporting no more than 16 electrodes[2]. Earlier, we found that PI and MI paired with RF and LR achieved high accuracy but required 75 to 80 electrodes, which is not practical for deployment. We impose a constraint on the number of electrodes used during training to improve practicality. Anticipating future advancements,

[2] https://tech.facebook.com/reality-labs/2021/3/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computing-platform/

we set a slightly higher limit to accommodate hardware advancements, restricting our analysis to top 20 electrodes.

4.3.1 ***Findings.*** : Figure 3 shows the accuracy trends across five users, averaged for each electrode count, and highlights the plateau behaviour with a limited number of electrodes. In particular, RF performs well across all selection methods, achieving 82%–88% accuracy within 20 electrodes. Among these, the PI + RF combination achieves the highest accuracy, reaching 88.33% with 19 electrodes. In contrast, SVC struggles to achieve high accuracy, even when using all 20 electrodes across various selection methods.

4.4 Balancing Sparsity and Accuracy

RQ3: Which combinations offer the best balance between classification accuracy and sparsity?

Our previous analysis (Figure 3) showed how each combination performed across users, gestures and sessions with a constrained set of 20 electrodes. Figure 13 shows the raw data of the accuracies for each of the combinations across each electrode count. Based on this data, selecting the best combination solely based on the lowest electrode count and highest accuracy may not be optimal, as some combinations exhibit early plateaus. For instance, SHAP and RMS-I show diminishing returns, where adding electrodes beyond a certain threshold provides negligible improvements. In these cases, the trade-off between additional electrodes and minimal accuracy gains becomes unfavorable. A notable example is MI + SVC, which achieves its peak accuracy using only 13 electrodes. However, the accuracy remains low at 37.78%. This illustrates the risks of prioritizing electrode reduction without considering classifier compatibility and emphasizes the importance of analyzing the trade-off curve rather than focusing solely on extreme values.

To address this, we develop a selection metric called "**Sparsity Score**', a metric designed to evaluate and balance the trade-off between the number of electrodes used and the classification accuracy achieved by a specific ESS-CL combination. It quantifies the point at which adding more electrodes provides diminishing returns in accuracy, penalizing combinations that require significantly more electrodes to achieve only marginal improvements in performance. The "**Sparsity Score**' is calculated as:

$Sparsity\ Score(E) = (w_1 * (100 - Accuracy(E)) + (w_2 * E)$, where, $(w_1 + w_2 = 1)$

We used a linear weighted scheme to balance the accuracy drop and the number of electrodes; however, an exponential scheme could also be used to assign steeper penalties for accuracy drop or electrode count. A lower sparsity score indicates a better balance between sparsity and accuracy. For this specific analysis, we set the weights equally (i.e $w_1 = w_2 = 0.5$). We determine the plateau point for each combination by identifying the electrode count E that minimizes this score. The scores are then averaged across the five users, and the combination with the lowest overall score is selected.

4.4.1 ***Findings.*** : The annotated triangles in Figure 3 demonstrate the sparsity scores for a given electrode count and accuracy. For e.g. for PI + RF, the best accuracy is 88.33%, but the best sparsity score is achieved at 9 electrodes, achieving an accuracy of 84.04%. Similarly, for PI + SVC, the sparsity scores are always higher because the accuracies are low. Despite this, the Sparsity Score enables us to identify a solution with a 70% drop in electrode count while having only a ~4.5% drop in accuracy. This approach ensures a more practical and efficient electrode selection strategy. A key finding from this analysis is that PI + RF is the best combination for this dataset. However, it is essential to note that this conclusion is based on results from a single dataset and cannot be generalized across multiple datasets.

4.5 Generalizability of the Selected Method Across Datasets

RQ4: SHAP + RF and PI + RF offer the best balance between classification accuracy and sparsity. But how well do they generalize across datasets?

To ensure the robustness of the proposed method, the selected combination must generalize well across diverse datasets, including potential custom datasets. To evaluate this generalizability, we extend the analysis from Section 4.4, where PI + RF combination demonstrated the best balance between sparsity and accuracy, to the additional datasets listed in Table 3.

Dataset	Channels	Gestures	Gesture Types
CSL-HDEMG [2]	192	29	Single-finger, Multi-finger
Hyser [33]	256	34	Single-finger, Multi-finger, Wrist
DELTA [17]	64	6	Multi-finger, Wrist
Nizamis et al. [63]	64	7	Single-finger, Wrist
GrabMyo [70]	28	17	Single-finger, Multi-finger, Wrist
putEMG [34]	24	8	Multi-finger, Wrist

Table 3: Datasets used for identifying the most optimal combination of subset selection method and classifier

Each dataset varies regarding the number of users, gestures, electrode hardware, sessions, and placement strategies. To ensure consistency across datasets, we took the following steps: (1) we selected the number of users that was the least common denominator across all datasets, and (2) we chose 20 trials per dataset, spread across sessions, to account for session variability.

We trained all combinations of subset selection methods and classifiers using data from five randomly selected users per dataset. Each combination was evaluated using 4-fold cross-validation. For each fold, the data was split in a stratified manner into training (75%) and testing (25%) sets, with the following splits: Hyser [33] (306 train, 102 test), DELTA [33] (90 train, 30 test), Nizamis et al. [63] (105 train, 35 test), GrabMyo [70] (255 train, 85 test), and putEMG [34] (105 train, 35 test).

For each user in a dataset, the Sparsity Score is computed for all combinations, and the average score across five users is then calculated. These scores are further averaged across datasets to determine the final score for each combination.

4.5.1 ***Findings.*** Figure 4 presents the averaged Sparsity Score across all datasets. The results show that the SHAP + RF combination performs the best, with the lowest score of 10.2. Interestingly,

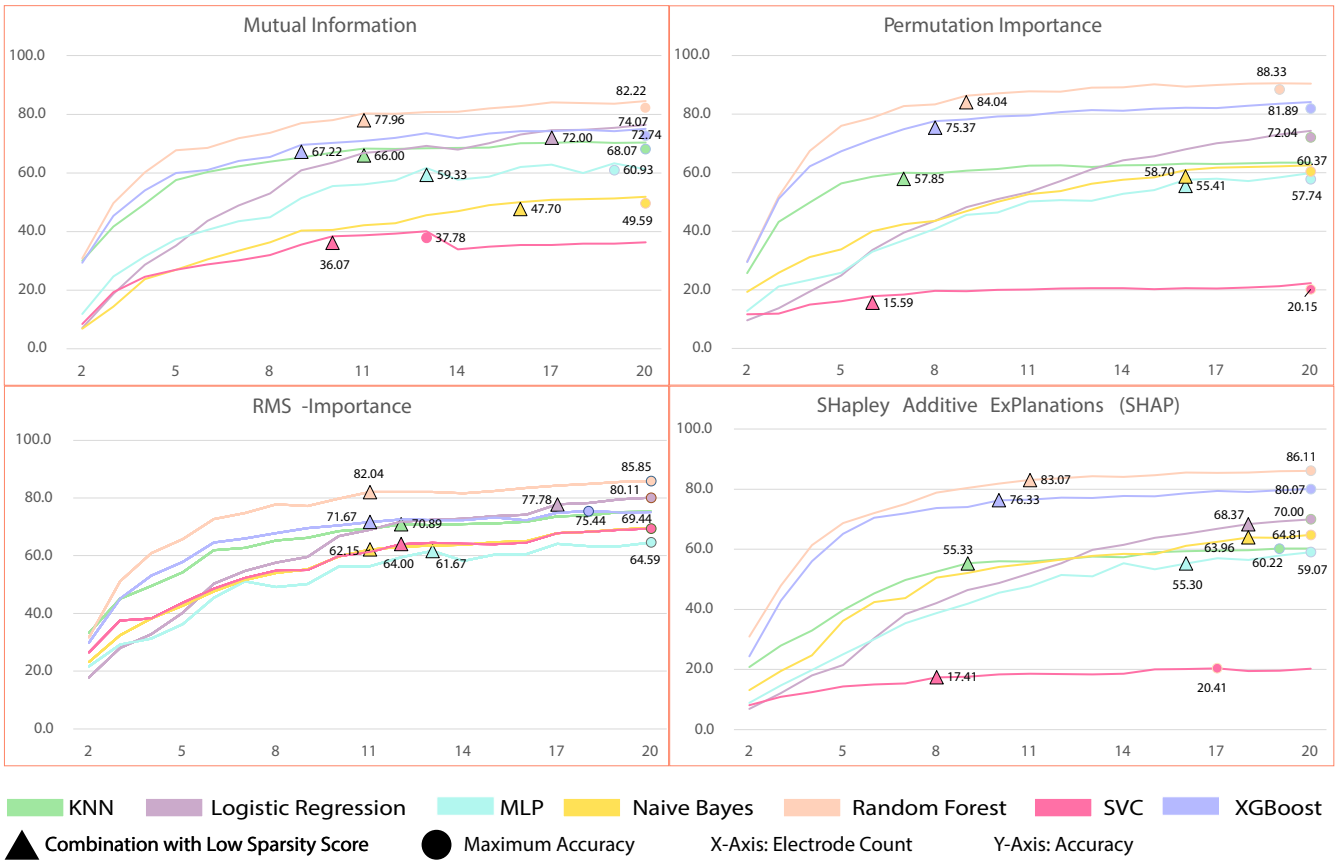


Figure 3: Comparison of Electrode selection and classifier combinations for a constraint of 20 electrodes across five users and 29 gestures. For each of the combinations, the maximum accuracy and the optimal configuration derived from sparsity scores are highlighted respectively.

the PI + RF combination also achieves a similar score of 10.6. These are followed by RMS + RF (12.5), MI + RF (13.8), RMS + NB (15.1), and RMS + LR (15.2), all demonstrating relatively strong performance.

In contrast, SVC and MLP consistently emerge as the weakest classifiers with the added constraint on the number of electrodes, leading to higher sparsity scores across their respective combinations.

4.6 Computation Time

RQ5: What is the variation in computation time across different electrode selection methods and classifiers?

One key criterion for the SparseEMG design tool is its ability to generate sparse layouts in near real-time, enabling rapid prototyping and iteration for gesture design. Since SHAP and Permutation Importance (PI) achieved the best Sparsity Scores across all datasets, it is essential to evaluate their suitability for real-time computation. For this, we analyzed the computation times for each electrode selection schemes (PI, MI, RMS-I, and SHAP) on the first dataset (CSL-HDEMG [2]), across users and for a randomly chosen session

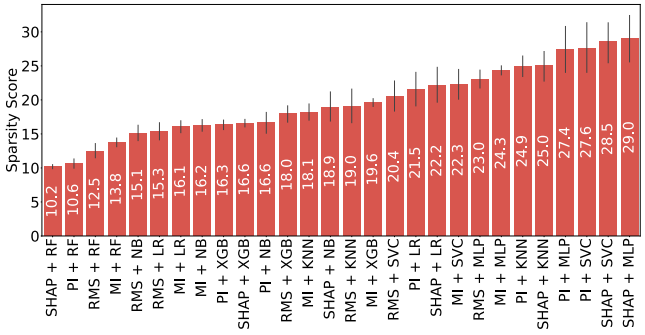


Figure 4: Average combination scores of subset selection methods with classifiers across all six datasets listed in Table 3 and five random users. Lower scores indicate better performance, with Permutation Importance (PI) + Random Forest (RF) achieving the lowest scores

with 27 gestures. MI and RMS-I are filter-based, model-independent methods, so their computation times were consistent across users.

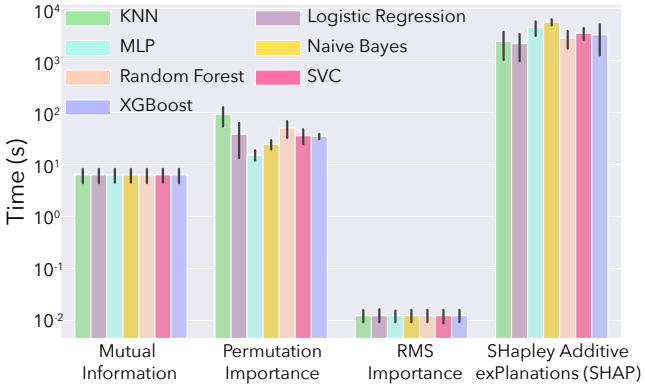


Figure 5: Computation times for the four electrode selection schemes.

Computation times were recorded on a desktop machine (Intel Core i7 [13th Generation], 32GB RAM, and NVIDIA GTX-4060 GPU).

4.6.1 ***Findings.*** Figure 5 illustrates the computation times for each selection method across all classifiers and users. SHAP took the longest time, averaging 3322.48 ± 1097.4 seconds, while RMS-Importance was the fastest, taking only 0.012 ± 0.003 seconds. Given the significant computational cost of SHAP, we excluded it from the design tool and selected the next best combination, Permutation Importance + Random Forest, as the primary electrode selection-classifier combination.

4.7 Summary of Findings

The key takeaways from the above in-depth analyses are:

(1) PI + RF offers the best trade-off between accuracy and sparsity, achieving near-peak performance (94.04%) with fewer electrodes (#76). In contrast, methods like RMS-I and SHAP require substantially more electrodes for only marginal accuracy gains.
(2) PI + RF delivers the highest accuracy (88.33%) even with a constraint of 20 electrodes, demonstrating strong performance under hardware-constrained conditions, while SVC consistently underperforms even with the full set of 20 electrodes.
(3) SHAP + RF and PI + RF consistently perform well in providing an optimal balance between sparsity and accuracy.
(4) SHAP's high computation time led to choosing PI + RF.

5 COMPUTATIONAL DESIGN TOOL FOR RAPID SELECTION OF CUSTOM SPARSE LAYOUTS

Based on our analysis, we contribute a computational design tool. It assists designers in the following tasks.

- *Finding a sparse EMG layout that balances high gesture recognition accuracy and sparsity:* Using the designer's specifications, the tool selects optimal designs in near real time and indicates the expected recognition accuracy. This also allows the designer to quickly understand how well a desired set of gestures can be recognized with a specified set of electrodes. The design tool assists designers in locating the electrode locations and
- *Exploring location alternatives:* Considerations of ergonomic wearability or aspects inherent to certain application cases may restrict the space where sEMG electrodes can be deployed on the user's forearm. For instance, an EMG band (e.g. MYO) with a pre-defined set of channels can only be placed on the upper region of the forearm. Similarly, an application case involving forearm actions may benefit from electrodes placed on the middle or upper region of the forearm rather than close to the fingertips. The tool allows the designer to select what locations can be augmented with sEMG electrodes and to explore alternatives quickly (Figure 6(c)).
- *Fine-tuning the Gesture Set for a given Electrode Configuration*: One key functionality of the design tool is to provide a visual representation that depicts the performance of the individual gestures (Figure 6(e)). This enables the designer to quickly inspect which gestures perform well and which do not and choose the most compatible gestures with high recognition accuracy.
- *Assisting in Electrode Placement:* One of the key challenges in EMG gesture recognition is the placement of electrodes. Replicating a given electrode placement is hard and can lead to significant performance issues [37, 59]. To assist the designers in replicating electrode placement for a given dataset, our design tool generates a placement stencil based on the details provided in the dataset.

5.1 Walkthrough

A screenshot of the design tool is shown in Figure 6. The designer first selects a dataset from a drop-down list. Once selected, the dataset details are loaded, including the gestures in the dataset, illustrations showing electrode placement. Next, she chooses the set of gestures that shall be recognized and selects the electrodes. The electrodes can be chosen by either directly selecting the electrodes or by sketching a region on the canvas. Then, the designer can place additional constraints for electrode placement. For instance, the designer selects the desired number of electrodes using the slider in the 'Model Parameters' panel. Additionally, the user can choose the electrode selection mechanism and the desired classifier. The parameters are sent to the backend with the click of a button. To visually present the recognition accuracy of chosen gestures, the tool displays a confusion matrix and the location of the individual electrodes on the canvas. If the designer is unsatisfied with the Tool's recommendation, she can quickly explore options iteratively. For instance, she may fine-tune the gestures or explore alternative locations for placing electrodes. We provide the source code to use our design tool [3].

5.2 Implementation

Notably, our tool differs from a conventional lookup table, which would require more than 4.2 billion entries for just one dataset (24 electrodes and 8 gestures) to cover the various combinations of electrodes and subsets of gestures. The front end of the design tool is implemented in HTML/JavaScript and uses the Snap.svg library.

[3] https://github.com/difflab-ucalgary/SparseEMG

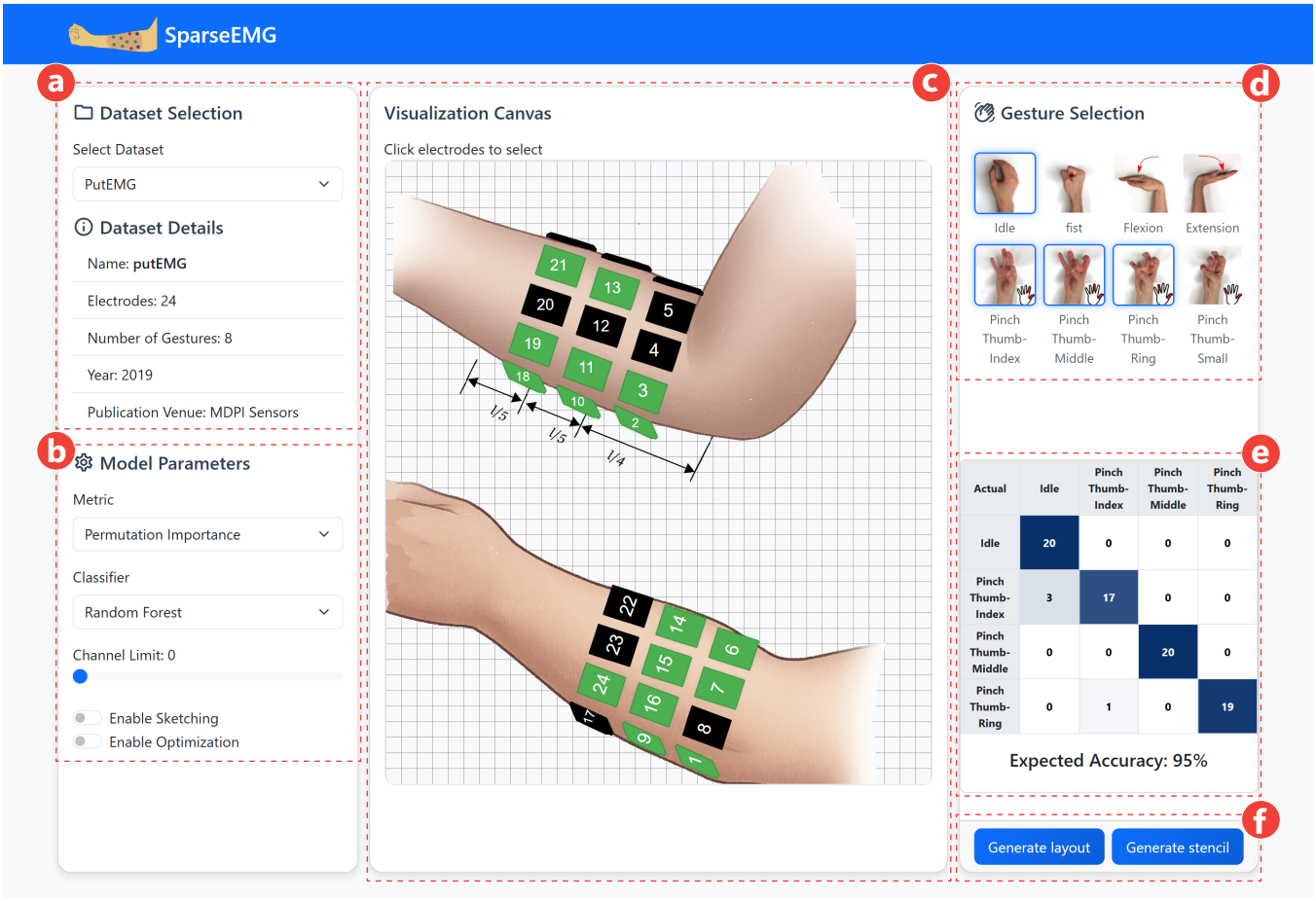


Figure 6: Screenshot of the SparseEMG Computational Design Tool

5.2.1 Dataset Scalability. To ensure easy and seamless integration of the datasets, we meticulously curated JSON files to add details of the dataset, such as the number of gestures, illustrations of gestures, electrode placement and other metadata (e.g. authors, electrode dimensions, spacing, etc.). For each dataset, we recreated the placements based on the instructions and descriptions from the publications. To allow the selection of electrodes within the tool, we created SVG files of the electrode placement and labeled them with their IDs. Once the user selects an electrode, we retrieve its ID through the Snap.svg library and update the global list, which consists of data about all the electrodes, their locations, and their selection status. We employ a similar approach for gesture selection as well.

5.2.2 Solution Generation. The backend of the tool is built using Python and the FastAPI framework. It receives information such as the dataset name, selected channels, maximum number of channels, gestures to recognize, subset selection method, and classifier type from the frontend via a WebSocket API call. The backend processes this information and begins training models using the top 2 to a maximum of 20 electrodes (or using the maximum number of channels, or only the selected channels). It performs 4-fold cross-validation for each electrode count to find the optimal sparse electrode configuration. After identifying the sparse configuration, the backend calculates the expected accuracy and confusion matrix for the selected gestures and sends the model file, accuracy, and confusion matrix back to the frontend.

Finally, the front-end receives this information, highlights the sparse electrode configuration, plots the confusion matrix, and displays the predicted accuracy.

5.3 Stencil Generation

The tool also offers the flexibility to generate a stencil based on the user's arm measurements (e.g., the arm's circumference at various levels as described in the original paper of the selected dataset). The placement stencil is created based on the electrode size and spacing dimensions reported in the chosen dataset. This stencil can be passed to any cutting machine. Once cut, the electrodes can be easily positioned on the user's arm according to the suggested sparse configuration.

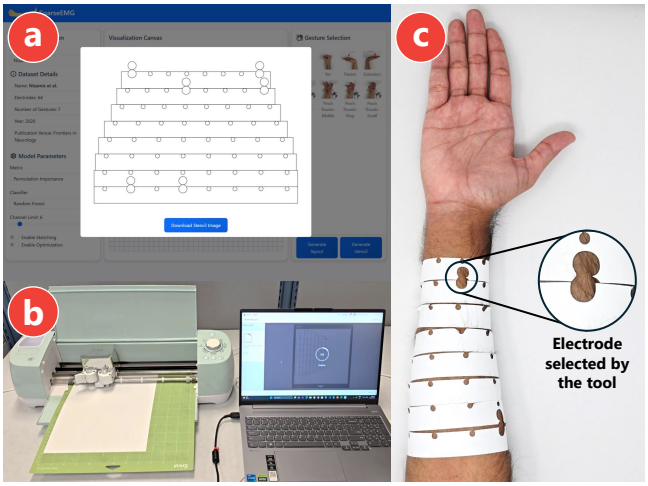


Figure 7: Applying electrode layout stencils generated by SparseEMG: (a) User interface for downloading the suggested sparse layout; (b) cutting the stencil using a paper cutter; (c) applying the stencil to the user's forearm for consistent and repeatable electrode placement.

6 DOES THE PLACEMENT SUGGESTED BY SparseEMG SCALE ACROSS USERS?

One of the main goals of the tool is to reduce the number of electrodes without needing to individually optimize placement for each new user. For practical deployment, it is essential that the electrode configurations identified by the tool generalize well across different users. If a sparse configuration derived from one user can be applied to others with minimal or no loss in performance, it indicates that the layouts generated by the tool are scalable across unseen users.

6.1 Method

We evaluated the tool using cross-user validation within each dataset. For each dataset, we selected one user to determine the optimal sparse electrode configuration based on their data. We then applied this configuration to the remaining users in the dataset and measured classification accuracy. This process was repeated for all the users, ensuring each served as the source-user once.

We applied this protocol across all six datasets to assess consistency and evaluate how well the tool's suggested placements generalized to other users. This validation helped determine if the tool identifies electrode placements that are robust and transferable across datasets.

6.2 Results

Figure 8 displays the results of the cross-user validation across all six datasets. Each plot shows a bar representing the accuracy for each user using their own electrode placement, alongside a bar showing the averaged accuracy when this user's layout is applied to others.

The tool-generated layouts demonstrated strong generalizability. The key takeaway is not the absolute accuracy but the consistency of accuracy when a layout from one user is applied to others. Ideally, if accuracy remains stable across users, it suggests that the tool generates configurations that generalize well. Datasets like CSL-HDEMG, DELTA, and Hyser exhibit minimal accuracy variation (2–6%), while putEMG shows slightly higher variability, with performance differences reaching 6–8% depending on the source layout.

These findings confirm the tool's ability to identify robust, transferable electrode layouts that scale without requiring per-user calibration. However, since this analysis used users within the same dataset with consistent conditions, we conduct an application-driven analysis in section 8 to test the effectiveness of the tool's layouts across different users and hardware setups.

7 SparseEMG VS. BAND FORM-FACTOR

Commercial EMG systems often employ fixed, evenly-spaced band configurations for convenience. In contrast, SparseEMG identifies performance-optimal sparse electrode subsets without geometric constraints. This section compares the classification accuracy of SparseEMG-generated layouts against typical fixed-band layouts, using identical electrode counts to assess the performance benefits of data-driven placement for interactive systems.

7.1 Method

We evaluated three standard band form-factor layouts (16, 8, and 4 electrodes), mimicking commercial systems by selecting equally spaced electrodes. These were directly compared against SparseEMG-derived layouts with the same electrode counts. The 16-electrode band was applied only to CSL-HDEMG [2] and Hyser [33] datasets due to their suitable circular electrode arrangements. To generate the band-form factor designs, we used our computational tool (Section 5) to manually select the electrodes that form a band around the upper forearm. For the SpaseEMG layouts, we used our design tool. We set the channel count to 16, 8 or 4 appropriately to generate the optimized electrode placement. Both the configurations (Band form-factor and SparseEMG layouts) used the same classifier (Random Forest, default parameters) and train-test split (75-25%).

7.2 Results

As shown in Figure 9, SparseEMG consistently outperforms fixed band layouts across all datasets and electrode counts. On average, SparseEMG improves classification accuracy by 9 percentage points at 16 electrodes, 3 points at 8 electrodes, and 1 point at 4 electrodes. These gains are particularly pronounced in low-electrode settings, where the band layout often falls below 60% accuracy, while SparseEMG maintains significantly higher performance. For instance, on the CSL-HDEMG [2] dataset, accuracy with four band-placed electrodes drops to 58.74%, whereas SparseEMG retains 65.19%, yielding a 7-point improvement. This trend holds across nearly all datasets, highlighting SparseEMG's robustness under sensor constraints and its potential to substantially enhance performance in wearable EMG systems where electrode count is limited.

8 APPLICATION-DRIVEN VALIDATION

A thorough usability evaluation of our computational design tool is challenging because of the multi-dimensional space involving

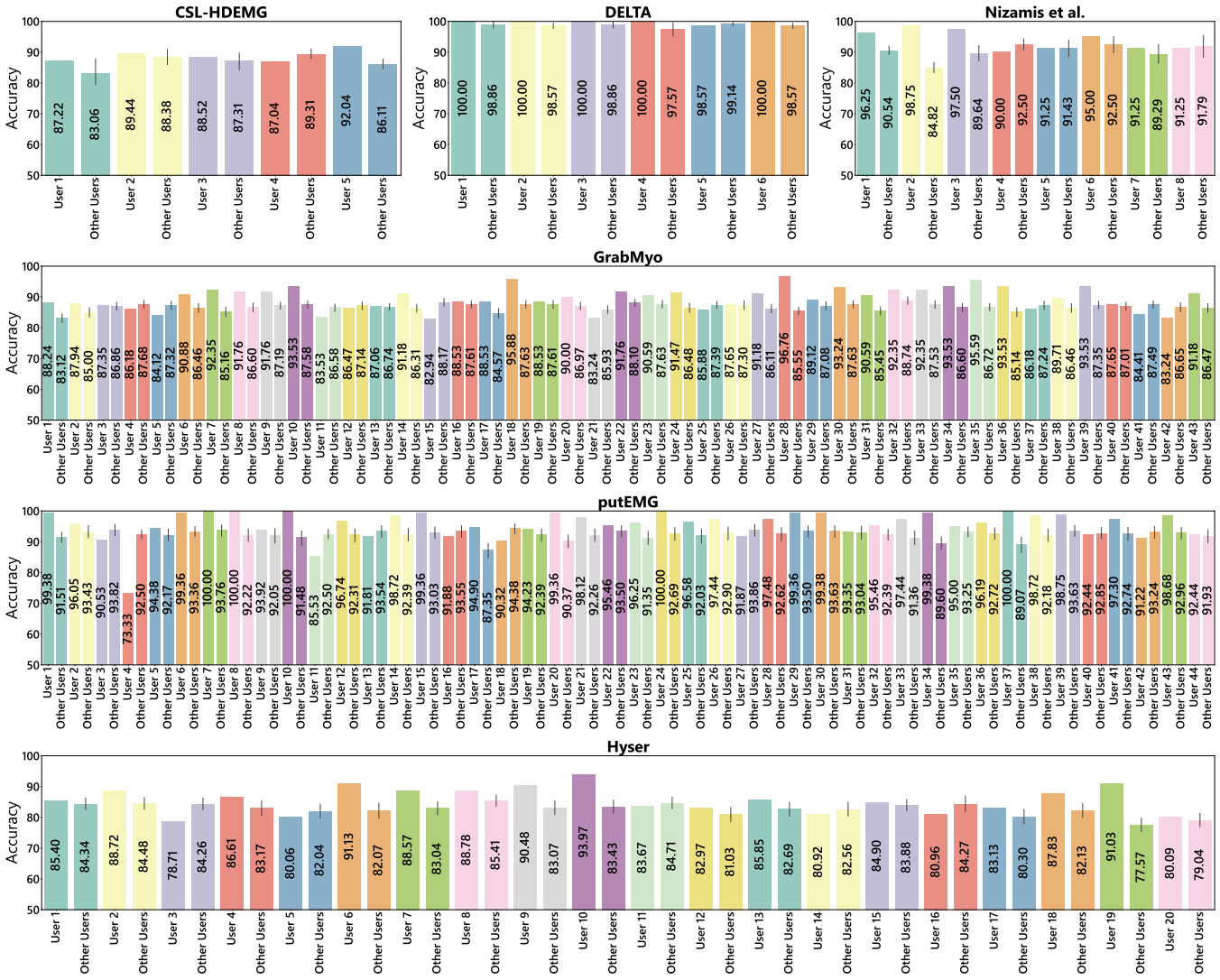


Figure 8: Cross-user validation accuracy for six datasets. Each sub-plot shows the accuracy for individual users using their own electrode layout (left bar) and the averaged accuracy when applying the user's electrode layout to other users (right bar)

electrode count, placement, datasets, gestures, and ML parameters resulting in combinatorial complexity. As recommended by Ledo et al., we adopt application-driven validation [49] to evaluate the practical usefulness of our tool. This approach has also been commonly employed in prior gesture work.

Our approach draws inspiration from prior work that contributed computational design tools [76, 85]. We collected three datasets based on different application scenarios with varying hardware configurations and participants to assess the tool's practical usefulness and generalizability to real-world applications. This section compares the accuracy achieved using the channels specified in the layouts suggested by the tool for the original dataset, as well as when these layouts are applied to participant recordings.

8.1 Acquisition Setup

We chose widely available recording hardware to make the setup more practical. Since Arduino-based hardware is commonly used in EMG gesture recognition, we selected the Olimex EMG Shield for Arduino-based boards to collect data for two datasets. This hardware allows stacking multiple shields and supports data collection from up to six channels with a single Arduino. We used six channels to record the data. The sampling rate is 256 Hz, the maximum supported by the board.

Since EMG signals are also commonly recorded using a band form factor, we recorded the third dataset with an EMG band [4] that has 8 EMG channels and a sampling rate of 500 Hz. The band also provides additional data, such as accelerometer and gyroscope

[4] http://www.oymotion.com/en/product32/149

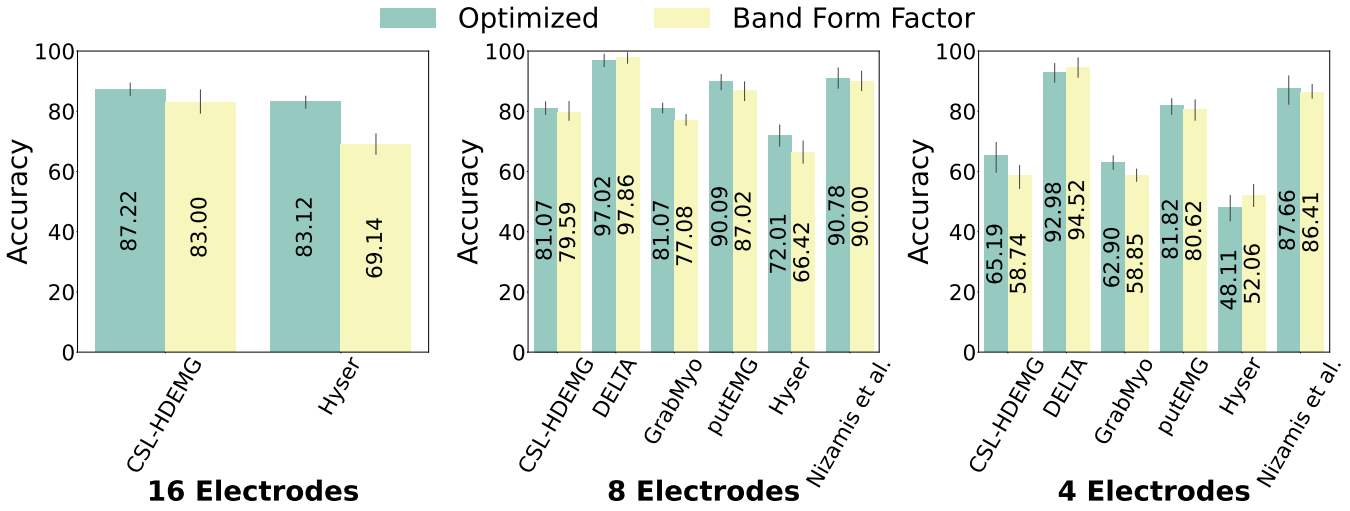


Figure 9: Comparison of classification accuracy using SparseEMG-selected electrodes (left bars) versus standard band form-factor layouts (right bars) across three electrode configurations (16, 8, and 4 electrodes) and six datasets. For each configuration, the number of electrodes in the band layout was matched to that of SparseEMG for fair comparison. The 16-electrode configuration was evaluated only on the CSL-HDEMG [2] and Hyser [33] datasets, as these were the only datasets with sufficient electrodes to support a circular band layout of that size.

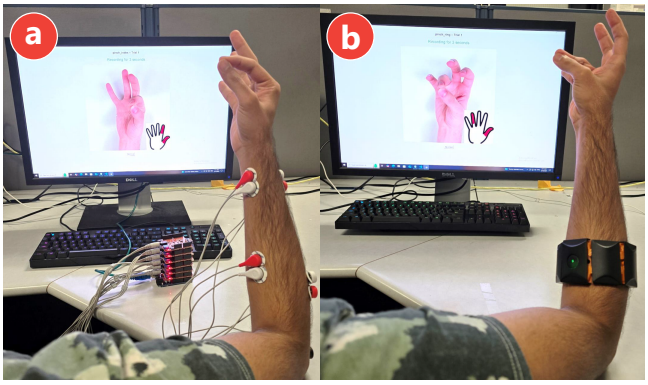


Figure 10: Acquisition setups used for acquiring EMG signals from study participants. (a) shows the Olimex EMG shield with six channels; (b) shows an armband similar to the Myo armband with eight channels.

values, but these were not used in the analysis and were, therefore, discarded.

We recorded the first two datasets using bipolar wet electrodes of 10 mm size (Covidien, H124SG), which are most commonly used for EMG signal acquisition.

8.2 Application Scenarios

8.2.1 Scenario 1: Assistive Technology for Daily Tasks. We envision our computational design tool assisting a designer, Ben, in creating a gesture recognition system for accessibility. Ben needs to design finger-based gestures that allow elderly users to perform daily tasks intuitively. He selects six gestures from the GrabMyo dataset [70], which are mapped to actions such as controlling smart home devices, initiating calls, sending messages, opening/closing apps, selecting actions, and triggering emergency alerts. Figure 11 illustrates the gestures and their corresponding functions.

8.2.2 Scenario 2: Smart Car Control. Similarly, another scenario involves Aisha, a designer focused on improving in-vehicle interaction through gesture-based controls. Her goal is to reduce driver distraction by enabling hands-free operation of essential car functions. Using the dataset from Nizamis et al.[63], she selects seven expressive hand and wrist gestures—such as wrist flexion, thumb extension, and fist—to represent commands like starting the engine, adjusting climate or audio settings, and answering phone calls. These gestures are integrated into the vehicle's interface to support intuitive, safe, and responsive interaction. Figure 11 depicts each gesture and its corresponding function.

8.2.3 Scenario 3: Virtual Reality Interaction. In this scenario, Ravi, a designer working on immersive interaction systems, explores the use of hand gestures to navigate and interact within a Virtual Reality (VR) environment. His objective is to enhance the intuitiveness of VR experiences by replacing traditional input methods with natural gesture-based controls. Drawing from the putEMG dataset [34], Ravi selects seven gestures that are well-suited for tasks such as object manipulation, spatial navigation, and engaging with virtual menus. These gestures are seamlessly integrated into the VR interface to support a more immersive and fluid user experience. Figure 11 illustrates the gesture-action mappings.

8.3 Participants

We recruited 8 participants for each scenario, with some participants overlapping between scenarios. The first scenario had 8

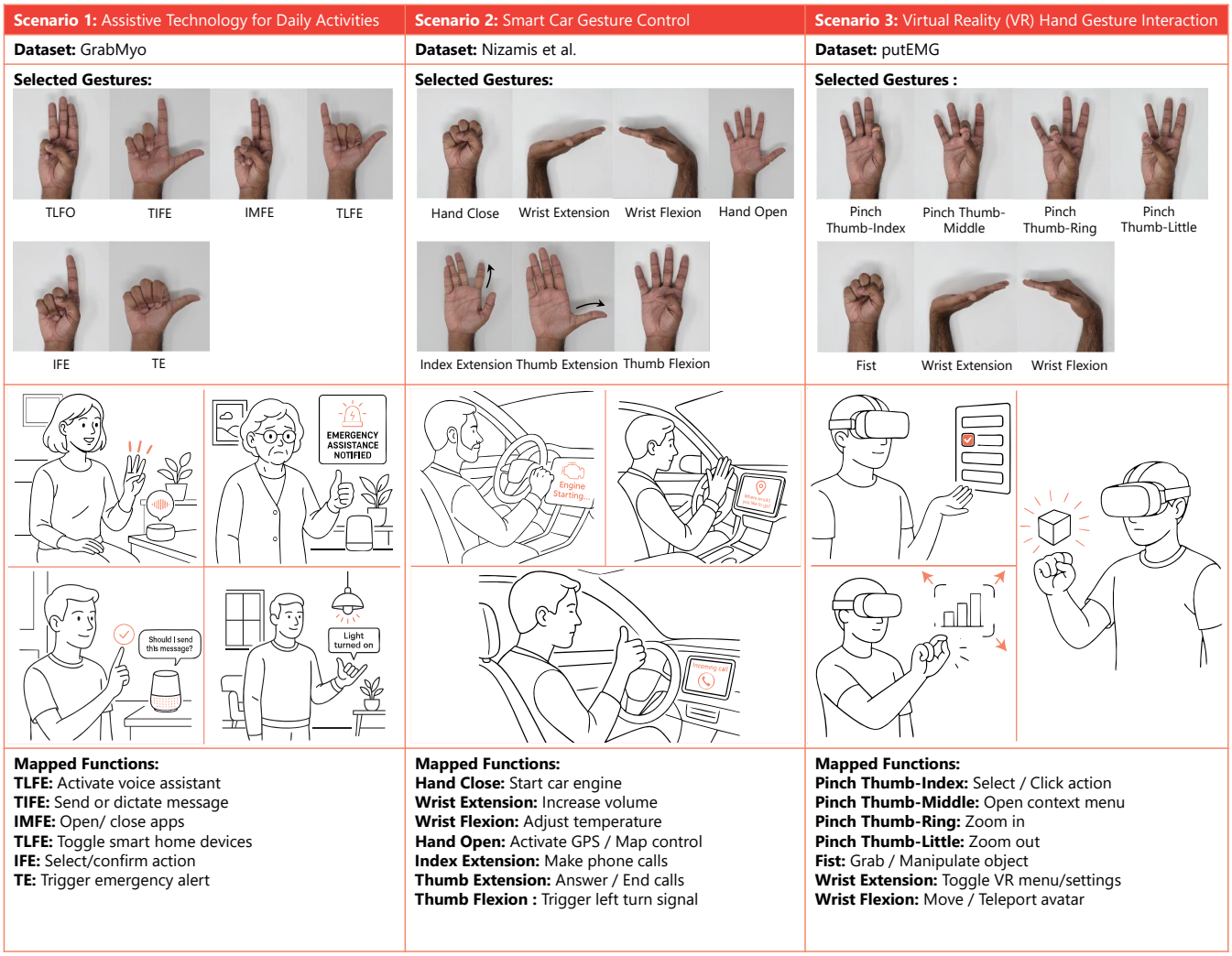


Figure 11: Overview of the three application scenarios used in the study: (1) Assistive Technology for Daily Activities using the GrabMyo [70] dataset, (2) Smart Car Gesture Control using the Nizamis et al. [63] dataset, and (3) Virtual Reality (VR) Hand Gesture Interaction using the putEMG [34] dataset. Each scenario highlights selected hand and wrist gestures along with their respective mapped functions, demonstrating diverse use-case applications of gesture recognition systems.

participants (4 M, 4 F, mean age: 25.5; SD: 5.1) with an average forearm length of 26.71 cm (SD: 2.02 cm) and an average forearm circumference of 24.57 cm (SD: 3.56 cm). The second scenario had 8 participants (5 M, 3 F, mean age: 26.13; SD: 5.46) with an average forearm length of 27.1 cm (SD: 2.47 cm) and an average forearm circumference of 25.47 cm (SD: 3.5 cm). The third scenario also had 8 participants (4 M, 4 F, mean age: 26; SD: 5.42), with an average forearm length of 28.9 cm (SD: 1.24 cm) and an average forearm circumference of 26.28 cm (SD: 3.5 cm).

8.4 Procedure

We first collected participant-specific measurements (forearm length and circumference at various points as required by each scenario) to generate a customized stencil based on the electrode layout suggested by the tool. This stencil was then sent to a paper cutter to produce a precise paper cutout. The placement of the stencil on the forearm was determined by the guidelines mentioned in the original dataset of the respective scenario. Once aligned according to the measurements, the stencil was carefully applied to the participant's forearm. Electrode positions were then marked directly onto the skin through the stencil using a highlighter, after which the stencil was removed. After removing the stencil, the electrodes were placed on the user's skin.

Initially, the participant performed a practice block with each gesture to help them become familiar with the gestures. Each block has one trial per gesture, and 10 blocks were recorded for each participant. After performing each gesture, the participants were

asked to bring their hands back to the rest state. This time between two gestures is recorded as a rest (non-gesture) gesture for the classifier. The order of the gestures was randomized within each block. After each block, participants were allowed to take a break and continue when ready. The full data collection session took approximately one hour per participant for each dataset.

Overall, our dataset comprises a total of 2,080 trials (10 trials × 8 participants × 25 gestures), with gestures drawn from three different datasets as well as a rest state.

8.5 Feature Extraction and Classification Comparison

In order to do a systematic comparison, we first evaluated the classification accuracy of the selected gestures using data from the original datasets (Table 3). For each dataset, we selected the top electrodes based on the rankings generated by the Permutation Importance + Random Forest (PI + RF) combination (six electrodes for scenario one and two, and eight electrodes for scenario three) using data from a random user in the dataset. This layout of top electrodes was then used for comparison between the original datasets and the datasets we recorded in our study.

For the original dataset, we used 10 trials per gesture from a single session and computed the classification accuracy using 4-fold cross-validation with a 75% training and 25% testing split for all the users in the dataset. The same feature vector described in Section 4.1.3 was used as input to the classifier.

We then evaluated classification accuracy using the datasets recorded in our study, following the same 4-fold cross-validation strategy and using the same feature vector to ensure consistency and fair comparison. For each recorded dataset, we computed the average accuracy across all 8 participants and compared them with the average accuracies calculated using the original datasets across all users.

8.6 Results

Figure 12 shows the classification accuracies obtained from the original datasets and the recorded datasets using the selected gesture subsets and electrode layouts generated by the tool.

For Scenario 1, the average accuracy with the original dataset using the top six electrodes was 84.57% (SD: 0.07), while the recorded dataset achieved an average accuracy of 82.88% (SD: 0.08). For Scenario 2, accuracy was 88.91% (SD: 0.06) for the original dataset, compared to 85.5% (SD: 0.08) for the recorded dataset. In Scenario 3, which used eight electrodes, accuracy was 87.02% (SD: 0.11) for the original data and 89.38% (SD: 0.05) for the recorded data which was higher than the original dataset.

Overall, the recorded datasets achieved accuracy levels comparable to the original datasets, with a drop of 2-4%. This indicates that the electrode layouts generated by the SparseEMG design tool generalized well to real-world hardware and participant variations. These results demonstrate the practical applicability and reliability of the tool for sparse EMG layout generation in real-world settings.

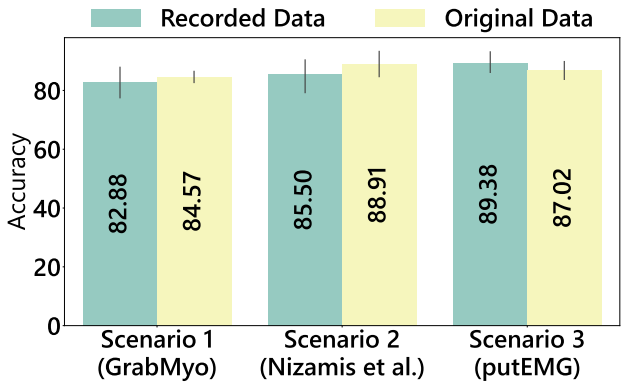


Figure 12: Comparison of accuracy between the recorded data and the data from the original datasets.

9 DISCUSSION, LIMITATIONS AND FUTURE WORK

SparseEMG introduces a novel approach to EMG-based interaction by optimizing electrode layouts for sparsity without significantly compromising gesture recognition accuracy. A key contribution is demonstrating that the combination of Permutation Importance (PI) and Random Forest (RF) provides a strong trade-off between these factors across diverse datasets, outperforming other techniques by maintaining a high classification accuracy (94.04%) while reducing the number of electrodes by 53.5%. This balance is critical for real-world applications where hardware limitations and user comfort are paramount. Unlike prior EMG toolkits, SparseEMG emphasizes sparse electrode configurations in achieving practical, scalable myoelectric systems, abstracting the complexities of electrode selection to empower designers to focus on gesture creation rather than underlying technical challenges, which could accelerate the adoption of myoelectric interfaces in fields such as assistive technologies, gaming, and augmented reality.

Benefits and Practical Usability. : SparseEMG provides major benefits by optimizing the electrode count as follows:

- Cost and Electrode Reduction: Many low-cost EMG systems (e.g. Arduino-based) support up to 6 channels for ~$100. Scaling to 8 channels raises costs to ~$500–$1.5K, while 16- and 64-channel lab-grade setups start at ~$8K and ~$10K–$25K respectively (excluding amplifiers/electrodes) and require specialized training. Stacking Arduinos is possible but impractical due to time sync and frame rate issues. Reducing channel count helps avoid these challenges. Since each channel uses 3 electrodes, even reducing from 8 to 6 channels removes 6 electrodes. As shown in Figure 3 via Sparsity Score, our tool can reduce total channels from 192 to 11 with only ~7-8% drop in accuracy.
- Boosting Frame Rate: For Arduino-based setups, having fewer channels means higher frame rate. With a standard 10-bit ADC the Arduino takes ~100-110 ms to sample each channel. Hence, if the number of channels is reduced from 6 to 4, the frame rate can be boosted from ~240 fps to ~360 fps (assuming a baud rate of 115200).

- Simplified & Scalable Electrode Placement: Electrode placement is critical for EMG signal acquisition but often requires expertise in anatomy, biomedical engineering, and consideration of the gesture set and ML model. Our tool abstracts these complexities, enabling users to generate gesture-specific layouts with just a few clicks. Scaling across users is difficult due to anatomical differences, so prior work relied on one-size-fits-all solutions like arrays or bands. Our stencil-generation feature enables reproducible and user-specific placements, which, to our knowledge, has not been explored in prior EMG classification work.

Scalablility Across Users and Hardware Platforms. : The generalizability of SparseEMG's recommendations across datasets and users further highlights its robustness. Our evaluation reveals that sparse layouts derived from one user or dataset perform consistently well when applied to others, with accuracy drops typically within 2–4%. This suggests that the tool generates electrode configurations that are not overly dependent on individual-specific or dataset-specific variations. This scalability across hardware platforms, from high-density arrays to consumer-grade devices, showcases its adaptability for both research and practical applications, ensuring its relevance across academic studies to commercial products. Moreover, SparseEMG positions machine learning as a design material, enabling interaction designers to iteratively prototype gesture sets and electrode configurations. By providing real-time feedback on estimated gesture recognition accuracy, the tool supports the rapid exploration of design alternatives, such as refining gesture sets or optimizing electrode placement for ergonomic and functional constraints, aligning with the broader HCI goal of leveraging computational tools to augment the creative process.

Improving Cross-User Performance. : While SparseEMG offers significant advancements, several limitations warrant discussion. First, our focus was on user-dependent models, and we did not evaluate whether the selected combination performs effectively in user-independent scenarios due to the limited number of users in the datasets. Building a robust user-independent model would require more diverse data, including samples from children, elderly individuals, and other underrepresented groups. Future work could explore integrating user-independent classifiers or transfer learning techniques to improve cross-user performance. Second, the tool currently supports gesture selection within a single dataset, which restricts its ability to handle gestures aggregated from multiple datasets. This limitation stems from variations in electrode placements, hardware configurations, and recording protocols across datasets. A standardized framework for integrating heterogeneous datasets could address this challenge and further enhance the tool's versatility. Third, while the tool abstracts the complexities of electrode placement, it relies on user-provided measurements for stencil generation. Variations in forearm dimensions and electrode placement guidelines across datasets introduce potential inconsistencies that may impact performance. Incorporating automated calibration methods or adaptive placement strategies could mitigate these issues.

Integrating Deep Learning Architectures. : SparseEMG opens several exciting avenues for future research. First, incorporating deep learning architectures, such as convolutional or recurrent neural networks, could enhance the tool's ability to model complex spatiotemporal dependencies in EMG signals. While our current approach prioritizes computational efficiency, advancements in hardware capabilities may enable the integration of more sophisticated models without compromising real-time performance. Second, expanding the tool's functionality to support dataset-independent modeling would significantly broaden its applicability. This could involve developing meta-learning algorithms that generalize across datasets or incorporating domain adaptation techniques to handle variations in hardware and recording conditions.

Applications beyond EMG. :This work focuses on EMG gesture recognition, but the approach can be applied more broadly within the general category of Muscle-Computer Interfaces as introduced by Saponas et al. [80]. Additionally, it can be adapted for use with various other sensing technologies, such as inertial measurement units (IMUs), electrical sensing, and mm-wave radars. For example, while our primary focus is on gesture recognition, the placement of electrodes is also crucial for providing haptic feedback through electrical muscle stimulation (EMS).

Moreover, the design tool can be adjusted for other applications, including muscle fatigue assessment, rehabilitation, movement analysis, and prosthesis control. Finally, the SparseEMG framework can be extended to incorporate other biosignal modalities, such as electroencephalography (EEG) and IMU sensors. By generalizing the design principles of our tool, we envision a wider array of computational tools that will empower designers to create interactive systems across multiple sensing technologies.

10 CONCLUSION

In this work, we systematically explored the relationship between electrode selection strategies and machine learning classifiers in the context of EMG-based gesture recognition. By evaluating 28 combinations, comprising four electrode selection schemes and seven classifiers, across six diverse datasets, we demonstrated that electrode count can be reduced by up to 53% without a significant drop in classification accuracy. Our results highlight the effectiveness of using the Permutation Importance method for electrode selection, paired with a Random Forest classifier. Building on these insights, we introduced SparseEMG, a novel design tool that enables users to generate sparse electrode layouts tailored to specific gesture sets, hardware constraints, and machine learning configurations. SparseEMG supports over 50 unique gestures, and our validation experiments confirm that: (1) the electrode layouts generated by the tool scale effectively across users, and (2) the tool demonstrates strong practical utility and generalizability across three real-world application scenarios. This work advances the field by foregrounding the value of principled electrode selection and contributes a data-driven, user-centered tool to accelerate the development of efficient EMG-based interaction systems. By enabling rapid iteration and informed trade-offs, SparseEMG addresses a critical bottleneck in EMG interface design and lays the groundwork for broader deployment of EMG in everyday interactive technologies.

ACKNOWLEDGMENTS

This project received funding from the National Science and Engineering Research Council (NSERC) Canada (RGPIN - 2023-03608), NFRF (New Frontiers in Research Fund (NFRFE/00256-2022), A-MEDICO(Alberta Medical Devices Innovation Consortium) and Alberta Innovates Postdoctoral Fellowship.

REFERENCES

[1] Adel Al-Jumaily and Ricardo A. Olivares. 2009. Electromyogram (EMG) driven system based virtual reality for prosthetic and rehabilitation devices. In *Proceedings of the 11th International Conference on Information Integration and Web-Based Applications & Services* (Kuala Lumpur, Malaysia) *(iiWAS '09)*. Association for Computing Machinery, New York, NY, USA, 582–586. https://doi.org/10.1145/1806338.1806448

[2] Christoph Amma, Thomas Krings, Jonas Böer, and Tanja Schultz. 2015. Advancing Muscle-Computer Interfaces with High-Density Electromyography. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 929–938. https://doi.org/10.1145/2702123.2702501

[3] Sheldon Andrews, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell. 2016. Real-time Physics-based Motion Capture with Sparse Sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016)* (London, United Kingdom) *(CVMP '16)*. Association for Computing Machinery, New York, NY, USA, Article 5, 10 pages. https://doi.org/10.1145/2998559.2998564

[4] Daniel Ashbrook and Thad Starner. 2010. MAGIC: a motion gesture design tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2159–2168. https://doi.org/10.1145/1753326.1753653

[5] Manfredo Atzori, Arjan Gijsberts, Simone Heynen, Anne-Gabrielle Mittaz Hager, Olivier Deriaz, Patrick van der Smagt, Claudio Castellini, Barbara Caputo, and Henning Müller. 2012. Building the Ninapro database: A resource for the biorobotics community. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 1258–1265. https://doi.org/10.1109/BioRob.2012.6290287

[6] Marco Barbero, Roberto Merletti, and Alberto Rainoldi. 2012. *Atlas of muscle innervation zones: understanding surface electromyography and its applications*. Springer Science & Business Media.

[7] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[8] Nicolò Celadon, Strahinja Dosen, Marco Paleari, Dario Farina, and Paolo Ariano. 2015. Individual finger classification from surface EMG: Influence of electrode set. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, New York, NY, USA, 7284–7287. https://doi.org/10.1109/EMBC.2015.7320073

[9] Lakshya Chaplot, Sara Houshmand, Karla Beltran Martinez, John Andersen, and Hossein Rouhani. 2024. Optimizing Sensor Placement and Machine Learning Techniques for Accurate Hand Gesture Classification. *Electronics* 13, 15 (2024). https://doi.org/10.3390/electronics13153072

[10] Enrico Costanza, Samuel A. Inverso, and Rebecca Allen. 2005. Toward subtle intimate interfaces for mobile devices using an EMG controller. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 481–489. https://doi.org/10.1145/1054972.1055039

[11] Enrico Costanza, Samuel A. Inverso, Rebecca Allen, and Pattie Maes. 2007. Intimate interfaces in action: assessing the usability and subtlety of emg-based motionless gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 819–828. https://doi.org/10.1145/1240624.1240747

[12] Kurtis Danyluk, Simon Klueber, Aditya Shekhar Nittala, and Wesley Willett. 2024. Understanding Gesture and Microgesture Inputs for Augmented Reality Maps. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 409–423. https://doi.org/10.1145/3643834.3661630

[13] Carlo J De Luca. 1997. The use of surface electromyography in biomechanics. *Journal of applied biomechanics* 13, 2 (1997), 135–163.

[14] Joseph DelPreto and Daniela Rus. 2020. Plug-and-Play Gesture Control Using Muscle and Motion Sensors. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction* (Cambridge, United Kingdom) *(HRI '20)*. Association for Computing Machinery, New York, NY, USA, 439–448. https://doi.org/10.1145/3319502.3374823

[15] Jia Deng, Jian Niu, Kun Wang, Li Xie, and Geng Yang. 2018. Discriminant analysis based EMG pattern recognition for hand function rehabilitation. In *Wireless Mobile Communication and Healthcare: 7th International Conference, MobiHealth 2017, Vienna, Austria, November 14–15, 2017, Proceedings 7*. Springer, 207–214.

[16] Aritra Dhar, Aditya Nittala, and Kuldeep Yadav. 2016. TactBack: VibroTactile braille output using smartphone and smartwatch for visually impaired. In *Proceedings of the 13th International Web for All Conference* (Montreal, Canada) *(W4A '16)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. https://doi.org/10.1145/2899475.2899514

[17] Dario Di Domenico, Nicolò Boccardo, Andrea Marinelli, Michele Canepa, Emanuele Gruppioni, Matteo Laffranchi, and Raffaello Camoriano. 2024. Long-Term Upper-Limb Prosthesis Myocontrol via High-Density sEMG and Incremental Learning. *IEEE Robotics and Automation Letters* 9, 11 (2024), 9938–9945. https://doi.org/10.1109/LRA.2024.3451388

[18] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 278–288. https://doi.org/10.1145/3025453.3025739

[19] Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu Liu, Ahmed Sabie, Sergio Orts-Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal. 2023. Rapsai: Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 125, 23 pages. https://doi.org/10.1145/3544548.3581338

[20] Eric J. Earley, Nathaly Sánchez Chan, Autumn A. Naber, Enzo Mastinu, Minh Tat Nhat Truong, and Max Ortiz-Catalan. 2024. Low-Cost, Wireless Bioelectric Signal Acquisition and Classification Platform. *IEEE Access* 12 (2024), 69350–69358. https://doi.org/10.1109/ACCESS.2024.3397909

[21] Karsten Eckhoff, Manon Kok, Sergio Lucia, and Thomas Seel. 2020. Sparse magnetometer-free inertial motion tracking–a condition for observability in double hinge joint systems. *IFAC-PapersOnLine* 53, 2 (2020), 16023–16030.

[22] Ethan Eddy, Evan Campbell, Angkoon Phinyomark, Scott Bateman, and Erik Scheme. 2023. LibEMG: An Open Source Library to Facilitate the Exploration of Myoelectric Control. *IEEE Access* 11 (2023), 87380–87397. https://doi.org/10.1109/ACCESS.2023.3304544

[23] Ethan Eddy, Erik J Scheme, and Scott Bateman. 2023. A Framework and Call to Action for the Future Development of EMG-Based Input in HCI. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 145, 23 pages. https://doi.org/10.1145/3544548.3580962

[24] Shreyosi Endow, Hedieh Moradi, Anvay Srivastava, Esau G Noya, and Cesar Torres. 2021. Compressables: A Haptic Prototyping Toolkit for Wearable Compression-based Interfaces. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference* (Virtual Event, USA) *(DIS '21)*. Association for Computing Machinery, New York, NY, USA, 1101–1114. https://doi.org/10.1145/3461778.3462057

[25] Yun Fang, Weichao Guo, and Xinjun Sheng. 2023. An Electrode Configuration Framework for sEMG-Based Gesture Recognition With Muscle-Matched Layouts. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–10. https://doi.org/10.1109/TIM.2023.3311054

[26] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 12. https://doi.org/10.1145/3332165.3347947

[27] G. Holmes, A. Donkin, and I.H. Witten. 1994. WEKA: a machine learning workbench. In *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference*. 357–361. https://doi.org/10.1109/ANZIIS.1994.396988

[28] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Trans. Graph.* 37, 6, Article 185 (Dec. 2018), 15 pages. https://doi.org/10.1145/3272127.3275108

[29] Han-Jeong Hwang, Janne Mathias Hahne, and Klaus-Robert Müller. 2014. Channel selection for simultaneous and proportional myoelectric prosthesis control of multiple degrees-of-freedom. *Journal of neural engineering* 11, 5 (2014), 056008.

[30] Lisa Brown Jaloza. 2021. Inside Facebook Reality Labs: wrist-based interaction for the next computing platform. *Tech at Meta* 18 (2021).

[31] C Jensen, O Vasseljen Jr, and RH Westgaard. 1996. Estimating maximal EMG amplitude for the trapezius muscle: on the optimization of experimental procedure and electrode placement for improved reliability and increased signal amplitude. *Journal of Electromyography and Kinesiology* 6, 1 (1996), 51–58.

[32] Xinyu Jiang, Xiangyu Liu, Jiahao Fan, Xinming Ye, Chenyun Dai, Edward A. Clancy, Metin Akay, and Wei Chen. 2021. Open Access Dataset, Toolbox and Benchmark Processing Results of High-Density Surface Electromyogram Recordings. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), 1035–1046. https://doi.org/10.1109/TNSRE.2021.3082551

[33] Xinyu Jiang, Xiangyu Liu, Jiahao Fan, Xinming Ye, Chenyun Dai, Edward A. Clancy, Metin Akay, and Wei Chen. 2021. Open Access Dataset, Toolbox and

Benchmark Processing Results of High-Density Surface Electromyogram Recordings. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), 1035–1046. https://doi.org/10.1109/TNSRE.2021.3082551

[34] Piotr Kaczmarek, Tomasz Mańkowski, and Jakub Tomczyński. 2019. putEMG—A Surface Electromyography Hand Gesture Recognition Dataset. *Sensors* 19, 16 (2019). https://doi.org/10.3390/s19163548

[35] Naveen Kumar Karnam, Shiv Ram Dubey, Anish Chand Turlapaty, and Balakrishna Gokaraju. 2022. EMGHandNet: A hybrid CNN and Bi-LSTM architecture for hand activity classification using surface EMG signals. *Biocybernetics and biomedical engineering* 42, 1 (2022), 325–340.

[36] Jakob Karolus, Annika Kilian, Thomas Kosch, Albrecht Schmidt, and Paweł W Wozniak. 2020. Hit the Thumb Jack! Using Electromyography to Augment the Piano Keyboard. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 429–440. https://doi.org/10.1145/3357236.3395500

[37] Jakob Karolus, Francisco Kiss, Caroline Eckerth, Nicolas Viot, Felix Bachmann, Albrecht Schmidt, and Pawel W. Wozniak. 2021. EMBody: A Data-Centric Toolkit for EMG-Based Interface Prototyping and Experimentation. *Proceedings of the ACM on Human-Computer Interaction* 5, EICS, Article 195 (May 2021), 29 pages. https://doi.org/10.1145/3457142

[38] Manami Katoh, Narihiro Nishimura, Maki Yokoyama, Taku Hachisu, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto. 2013. Optimal selection of electrodes for muscle electrical stimulation using twitching motion measurement. In *Proceedings of the 4th Augmented Human International Conference* (Stuttgart, Germany) *(AH '13)*. Association for Computing Machinery, New York, NY, USA, 237–238. https://doi.org/10.1145/2459236.2459279

[39] Frederic Kerber, Pascal Lessel, and Antonio Krüger. 2015. Same-side Hand Interactions with Arm-placed Devices Using EMG. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. Association for Computing Machinery, New York, NY, USA, 1367–1372. https://doi.org/10.1145/2702613.2732895

[40] Frederic Kerber, Michael Puhl, and Antonio Krüger. 2017. User-independent real-time hand gesture recognition based on surface electromyography. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, Article 36, 7 pages. https://doi.org/10.1145/3098279.3098553

[41] Rami N Khushaba, Sarath Kodagoda, Maen Takruri, and Gamini Dissanayake. 2012. Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals. *Expert Systems with Applications* 39, 12 (2012), 10731–10738.

[42] Ju-Whan Kim, Han-Jong Kim, and Tek-Jin Nam. 2016. M.Gesture: An Acceleration-Based Gesture Authoring System on Multiple Handheld and Wearable Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2307–2318. https://doi.org/10.1145/2858036.2858358

[43] Ju-Whan Kim and Tek-Jin Nam. 2013. EventHurdle: supporting designers' exploratory interaction prototyping with gesture-based sensors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 267–276. https://doi.org/10.1145/2470654.2470691

[44] Ron Kohavi and George H John. 1997. Wrappers for feature subset selection. *Artificial intelligence* 97, 1-2 (1997), 273–324.

[45] Daniel Kohlsdorf, Thad Starner, and Daniel Ashbrook. 2011. MAGIC 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. 1–6. https://doi.org/10.1109/FG.2011.5771412

[46] Michinari Kono, Yoshio Ishiguro, Takashi Miyaki, and Jun Rekimoto. 2018. Design and Study of a Multi-Channel Electrical Muscle Stimulation Toolkit for Human Augmentation. In *Proceedings of the 9th Augmented Human International Conference* (Seoul, Republic of Korea) *(AH '18)*. Association for Computing Machinery, New York, NY, USA, Article 11, 8 pages. https://doi.org/10.1145/3174910.3174913

[47] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69, 6 (2004), 066138.

[48] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3173574.3173610

[49] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3173574.3173610

[50] Jianfeng Li, Xinyu Jiang, Xiangyu Liu, Fumin Jia, and Chenyun Dai. 2024. Optimizing the feature set and electrode configuration of high-density electromyogram via interpretable deep forest. *Biomedical Signal Processing and Control* 87 (2024), 105445.

[51] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. 2015. BackHand: Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 557–564. https://doi.org/10.1145/2807442.2807462

[52] Linjie Liu, Na Zhang, and Ke Li. 2022. Research on gesture recognition of surface EMG based on machine learning. In *Chinese Conference on Biometric Recognition*. Springer, 276–283.

[53] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1471–1482. https://doi.org/10.1145/3442381.3449890

[54] Hao Lü and Yang Li. 2012. Gesture coder: a tool for programming multi-touch gestures by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2875–2884. https://doi.org/10.1145/2207676.2208693

[55] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[56] Enzo Mastinu, Bo Håkansson, and Max Ortiz-Catalan. 2017. Low-cost, open source bioelectric signal acquisition system. In *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 19–22. https://doi.org/10.1109/BSN.2017.7935997

[57] Moritz Alexander Messerschmidt, Sachith Muthukumarana, Nur Al-Huda Hamdan, Adrian Wagner, Haimo Zhang, Jan Borchers, and Suranga Chandima Nanayakkara. 2022. ANISMA: A Prototyping Toolkit to Explore Haptic Skin Deformation Applications Using Shape-Memory Alloys. *ACM Trans. Comput.-Hum. Interact.* 29, 3, Article 19 (Jan. 2022), 34 pages. https://doi.org/10.1145/3490497

[58] Dinmukhammed Mukashev, Nimesha Ranasinghe, and Aditya Shekhar Nittala. 2023. TactTongue: Prototyping ElectroTactile Stimulations on the Tongue. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 105, 14 pages. https://doi.org/10.1145/3586183.3606829

[59] Aditya Shekhar Nittala, Andreas Karrenbauer, Arshad Khan, Tobias Kraus, and Jürgen Steimle. 2021. Computational design and optimization of electrophysiological sensors. *Nature communications* 12, 1 (2021), 6351.

[60] Aditya Shekhar Nittala, Arshad Khan, Klaus Kruttwig, Tobias Kraus, and Jürgen Steimle. 2020. PhysioSkin: Rapid Fabrication of Skin-Conformal Physiological Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376366

[61] Aditya Shekhar Nittala and Jürgen Steimle. 2022. Next Steps in Epidermal Computing: Opportunities and Challenges for Soft On-Skin Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 389, 22 pages. https://doi.org/10.1145/3491102.3517668

[62] Yinxi Niu, Wensheng Chen, Hui Zeng, Zhenhua Gan, and Baoping Xiong. 2024. Optimizing semg gesture recognition: Leveraging channel selection and feature compression for improved accuracy and computational efficiency. *Applied Sciences* 14, 8 (2024), 3389.

[63] Kostas Nizamis, Noortje HM Rijken, Robbert Van Middelaar, João Neto, Bart FJM Koopman, and Massimo Sartori. 2020. Characterization of forearm muscle activation in duchenne muscular dystrophy via high-density electromyography: A case study on the implications for myoelectric control. *Frontiers in neurology* 11 (2020), 231.

[64] Max Ortiz-Catalan, Rickard Brånemark, and Bo Håkansson. 2013. BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. *Source code for biology and medicine* 8 (2013), 1–18.

[65] Max Ortiz-Catalan, Rickard Brånemark, and Bo Håkansson. 2013. Evaluation of classifier topologies for the real-time classification of simultaneous limb motions. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 6651–6654. https://doi.org/10.1109/EMBC.2013.6611081

[66] Sai Nandan Panigrahy, Chang Hyeon Lee, Vrahant Nagoria, Mohammad Janghorban, Richa Pandey, and Aditya Shekhar Nittala. 2024. ecSkin: Low-Cost Fabrication of Epidermal Electrochemical Sensors for Detecting Biomarkers in Sweat. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 336, 20 pages. https://doi.org/10.1145/3613904.3642232

[67] Aman Parnami, Apurva Gupta, Gabriel Reyes, Ramik Sadana, Yang Li, and Gregory D. Abowd. 2016. Mogeste: A Mobile Tool for In-Situ Motion Gesture Design. In *Proceedings of the 8th Indian Conference on Human-Computer Interaction* (Mumbai, India) *(IndiaHCI '16)*. Association for Computing Machinery, New York, NY, USA, 35–43. https://doi.org/10.1145/3014362.3014365

[68] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Amy J. Ko, and James Landay. 2010. Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/1866029.1866038

[69] Luis Pelaez Murciego, Mauricio C Henrich, Erika G Spaich, and Strahinja Dosen. 2022. Reducing the number of EMG electrodes during online hand gesture classification with changing wrist positions. *Journal of NeuroEngineering and Rehabilitation* 19, 1 (2022), 78.

[70] Ashirbad Pradhan, Jiayuan He, and Ning Jiang. 2022. Multi-day dataset of forearm and wrist electromyogram for hand gesture recognition and biometrics. *Scientific Data* 9, 1 (Nov. 2022). https://doi.org/10.1038/s41597-022-01836-y

[71] Cosima Prahm, Michael Bressler, Korbinian Eckstein, Hideaki Kuzuoka, Adrien Daigeler, and Jonas Kolbenschlag. 2022. Developing a wearable Augmented Reality for treating phantom limb pain using the Microsoft Hololens 2. In *Proceedings of the Augmented Humans International Conference 2022* (Kashiwa, Chiba, Japan) *(AHs '22)*. Association for Computing Machinery, New York, NY, USA, 309–312. https://doi.org/10.1145/3519391.3524031

[72] Cosima Prahm, Korbinian Eckstein, Max Ortiz-Catalan, Georg Dorffner, Eugenijus Kaniusas, and Oskar C Aszmann. 2016. Combining two open source tools for neural computation (BioPatRec and Netlab) improves movement classification for prosthetic control. *BMC Research Notes* 9 (2016), 1–7.

[73] Kartik S Prakash and Nissan Kunju. 2025. An optimized electrode configuration for wrist wearable EMG-based hand gesture recognition using machine learning. *Expert Systems with Applications* (2025), 127040.

[74] Alberto Rainoldi, G Melchiorri, and I Caruso. 2004. A method for positioning electrodes during surface EMG recordings in lower limb muscles. *Journal of neuroscience methods* 134, 1 (2004), 37–43.

[75] Seema Rawat, Somya Vats, and Praveen Kumar. 2016. Evaluating and exploring the MYO ARMBAND. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*. 115–120. https://doi.org/10.1109/SYSMART.2016.7894501

[76] Artin Saberpour Abadian, Ata Otaran, Martin Schmitz, Marie Muehlhaus, Rishabh Dabral, Diogo Luvizon, Azumi Maekawa, Masahiko Inami, Christian Theobalt, and Jürgen Steimle. 2023. Computational Design of Personalized Wearable Robotic Limbs. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 68, 13 pages. https://doi.org/10.1145/3586183.3606748

[77] Sasha Salter, Richard Warren, Collin Schlager, Adrian Spurr, Shangchen Han, Rohin Bhasin, Yujun Cai, Peter Walkington, Anuoluwapo Bolarinwa, Robert J Wang, et al. 2025. emg2pose: A large and diverse benchmark for surface electromyographic hand pose estimation. *Advances in Neural Information Processing Systems* 37 (2025), 55703–55728.

[78] T Scott Saponas, Desney S. Tan, Dan Morris, and Ravin Balakrishnan. 2008. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 515–524. https://doi.org/10.1145/1357054.1357138

[79] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (Victoria, BC, Canada) *(UIST '09)*. Association for Computing Machinery, New York, NY, USA, 167–176. https://doi.org/10.1145/1622176.1622208

[80] T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making muscle-computer interfaces more practical. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 851–854. https://doi.org/10.1145/1753326.1753451

[81] Christos Sapsanis, A Tzes, and G Georgoulas. 2014. sEMG for basic hand movements data set. *UCI Machine Learning Repository* (2014).

[82] Matthias Schröder, Jonathan Maycock, and Mario Botsch. 2015. Reduced marker layouts for optical motion capture of hands. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games* (Paris, France) *(MIG '15)*. Association for Computing Machinery, New York, NY, USA, 7–16. https://doi.org/10.1145/2822013.2822026

[83] Adwait Sharma, Michael A. Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. 2021. SoloFinger: Robust Microgestures while Grasping Everyday Objects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 744, 15 pages. https://doi.org/10.1145/3411764.3445197

[84] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping Microgestures: Eliciting Single-hand Microgestures for Handheld Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300632

[85] Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. 2023. SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-grained Finger Microgestures. *ACM Trans. Comput.-Hum. Interact.* 30, 3, Article 39 (June 2023), 40 pages. https://doi.org/10.1145/3569894

[86] Kirill A. Shatilov, Dimitris Chatzopoulos, Alex Wong Tat Hang, and Pan Hui. 2019. Using Deep Learning and Mobile Offloading to Control a 3D-printed Prosthetic Hand. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 102 (Sept. 2019), 19 pages. https://doi.org/10.1145/3351260

[87] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 59 (June 2020), 25 pages. https://doi.org/10.1145/3397309

[88] Sidi Mohamed Sid'El Moctar, Imad Rida, and Sofiane Boudaoud. 2024. Comprehensive Review of Feature Extraction Techniques for sEMG Signal Classification: From Handcrafted Features to Deep Learning Approaches. *IRBM* (2024), 100866.

[89] Paul Strohmeier, Seref Güngör, Luis Herres, Dennis Gudea, Bruno Fruchard, and Jürgen Steimle. 2020. bARefoot: Generating Virtual Materials using Motion Coupled Vibration in Shoes. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 579–593. https://doi.org/10.1145/3379337.3415828

[90] Aaron Tabor, Scott Bateman, Erik Scheme, David R. Flatla, and Kathrin Gerling. 2017. Designing Game-Based Myoelectric Prosthesis Training. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1352–1363. https://doi.org/10.1145/3025453.3025676

[91] Aaron Tabor, Scott Bateman, Erik Scheme, David R. Flatla, and Kathrin Gerling. 2017. Designing Game-Based Myoelectric Prosthesis Training. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1352–1363. https://doi.org/10.1145/3025453.3025676

[92] Peter Khoa Duc Tran, Purna Valli Anusha Gadepalli, Jaeyeon Lee, and Aditya Shekhar Nittala. 2023. Augmenting On-Body Touch Input with Tactile Feedback Through Fingernail Haptics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 79, 13 pages. https://doi.org/10.1145/3544548.3581473

[93] Jorge R Vergara and Pablo A Estévez. 2014. A review of feature selection methods based on mutual information. *Neural computing and applications* 24 (2014), 175–186.

[94] Scott H. Walbran, Emilio P. Calius, G. Reg Dunlop, and Iain A. Anderson. 2009. A technique for optimizing electrode placement for electromyographic control of prostheses. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Minneapolis, USA). IEEE, New York, NY, USA, 1331–1334. https://doi.org/10.1109/IEMBS.2009.5332404

[95] Clifford R. Wheeless. 2025. Flexor Carpi Radialis. https://www.wheelessonline.com/muscles-tendons/flexor-carpi-radialis/. Last Accessed: 2025-02-28.

[96] Mulun Wu, Yanbin Xu, Chenguang Yang, and Ying Feng. 2018. Omnidirectional Mobile Robot Control based on Mixed Reality and sEMG Signals. In *2018 Chinese Automation Congress (CAC)*. 1867–1872. https://doi.org/10.1109/CAC.2018.8623114

[97] Willa Yunqi Yang, Yifan Zou, Jingle Huang, Raouf Abujaber, and Ken Nakagaki. 2024. TorqueCapsules: Fully-Encapsulated Flywheel Actuation Modules for Designing and Prototyping Movement-Based and Kinesthetic Interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 98, 15 pages. https://doi.org/10.1145/3654777.3676364

[98] Jianzhuang Zhao, Ye Tian, Yuxuan Wang, Weiye Ji, and Mingchi Zhu. 2024. Adaptive optimization of electromyographic channels for intelligent prosthetic hands based on individual differences. *Advances in Mechanical Engineering* 16, 12 (2024), 16878132241306261.

A APPENDIX

No. of Electrodes	RMS + SVC	RMS + Logistic Regression	RMS + KNN	RMS + Naive Bayes	RMS + XGB	RMS + Random Forest	SHAP + SVC	SHAP + Logistic Regression	SHAP + KNN	SHAP + Naive Bayes	SHAP + XGB	SHAP + Random Forest	Mutual Information + SVC	Mutual Information + Logistic Regression	Mutual Information + KNN	Mutual Information + Naive Bayes	Mutual Information + XGB	Mutual Information + Random Forest	Permutation Importance + SVC	Permutation Importance + Logistic Regression	Permutation Importance + KNN	Permutation Importance + Naive Bayes	Permutation Importance + XGB	Permutation Importance + Random Forest
2	35.93	31.48	39.81	33.70	35.93	37.59	9.81	10.00	17.04	11.30	20.93	34.63	12.04	9.44	31.11	11.30	31.11	35.00	13.52	10.00	19.44	16.11	25.37	33.52
3	38.15	40.00	44.44	39.26	43.52	47.22	12.96	24.63	22.41	15.37	48.15	50.19	16.30	17.41	50.56	14.81	53.33	57.96	20.00	15.93	41.48	19.07	48.15	61.48
4	41.85	47.41	50.56	44.26	53.33	57.96	20.74	30.37	28.33	22.78	64.81	73.52	23.33	28.70	60.00	28.15	62.22	70.93	24.07	23.33	49.26	22.78	70.56	71.30
5	53.70	54.63	57.22	45.93	56.48	62.22	27.04	36.48	46.85	35.37	69.81	73.52	25.19	40.00	63.33	30.19	65.74	73.33	26.85	36.67	56.85	22.78	72.04	71.85
6	63.52	66.67	65.37	60.56	66.48	74.81	26.48	49.07	49.81	39.26	76.48	78.33	23.70	45.56	61.85	28.89	65.37	73.70	29.07	41.48	57.04	25.74	75.37	74.63
7	65.74	68.15	65.37	62.04	66.30	72.41	27.22	50.74	58.15	39.81	77.59	79.81	23.89	50.19	62.59	31.11	68.15	74.44	27.96	45.93	59.81	25.37	77.22	80.19
8	66.48	71.30	69.63	67.22	69.44	77.96	29.07	56.11	59.63	38.15	78.70	78.33	25.19	53.70	63.70	32.04	68.52	74.63	27.96	54.26	60.74	25.56	78.15	82.04
9	68.89	72.04	72.22	68.15	72.78	78.52	28.33	60.93	66.48	38.15	78.33	78.52	23.89	55.74	61.85	32.41	69.07	76.85	26.48	57.04	63.70	29.26	80.19	83.52
10	70.74	75.00	74.26	70.74	73.33	78.15	29.26	60.93	70.19	37.96	80.19	78.15	29.63	65.93	70.74	34.63	70.56	80.93	26.11	58.70	65.74	29.44	80.56	85.74
11	75.56	79.81	76.48	75.74	77.04	82.78	29.44	62.22	68.33	37.59	82.22	79.44	28.33	67.04	71.67	33.33	69.26	79.63	26.30	62.04	66.48	28.33	80.93	86.30
12	77.96	83.33	79.44	79.26	78.52	85.74	27.96	66.11	68.33	37.78	81.85	77.78	27.96	68.33	71.67	32.59	72.04	80.00	25.37	65.00	68.33	27.59	83.15	87.41
13	79.26	84.26	79.44	79.63	78.52	84.26	27.22	72.04	67.22	39.26	81.85	77.96	28.15	70.00	70.74	35.19	73.15	80.93	24.07	68.52	67.41	29.81	82.41	87.22
14	79.07	84.07	79.44	78.15	79.63	84.63	25.93	75.19	67.41	40.00	82.96	80.19	27.59	70.19	70.19	35.19	72.04	80.74	24.44	70.37	70.00	30.37	80.93	86.48
15	78.15	84.07	79.07	80.19	80.00	86.48	25.56	76.67	69.63	38.89	83.52	79.63	28.15	70.93	71.30	36.48	72.96	80.00	23.89	71.67	68.89	31.30	80.37	88.70
16	78.33	83.89	78.15	80.00	76.85	86.30	26.30	77.78	70.00	38.52	82.78	81.11	29.07	74.07	71.11	37.04	72.41	80.74	23.52	73.15	69.63	31.48	81.48	87.41
17	78.52	85.37	79.26	81.11	80.74	87.59	26.11	79.44	70.74	40.93	83.52	81.48	27.41	76.11	70.37	37.04	72.96	82.22	22.78	73.52	69.81	32.22	82.41	86.48
18	79.63	84.44	79.63	81.30	81.85	86.85	25.19	79.44	70.93	42.04	81.48	80.74	28.52	75.56	70.00	37.96	72.78	82.04	23.33	73.70	72.78	31.85	82.78	88.15
19	79.63	85.19	80.37	80.74	80.37	86.85	23.89	80.37	71.30	41.11	81.11	81.11	28.33	75.93	70.37	38.52	72.59	82.04	23.89	74.81	72.59	34.26	84.26	87.59
20	80.93	87.22	79.81	81.30	79.63	88.33	24.81	81.48	71.30	43.15	83.33	81.48	29.63	74.81	68.89	38.89	74.44	82.22	24.44	74.81	72.22	33.33	84.63	86.85

Figure 13: Raw data after setting a constraint of 20 electrodes for all the combinations