UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences

Transfer Learning on Electrophysiological Data

Petro Fagurel
Promotors: Prof. Gianluca Bontempi, Martin Colot and Cédric Simar
Masters Thesis in Computer Science

2023

Disclaimer

I hereby declare that this thesis is my original work. It has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own.

Acknowledgment

This project would not have been possible without the support, encouragement, and guidance of many individuals who have contributed to its success.

First and foremost, I would like to extend heartfelt gratitude to my parents, for their endless support.

I am deeply thankful to my esteemed supervisors, Martin Colot, Cédric Simar, and Prof. Gianluca Bontempi, for their mentorship and critical insights.

Special thanks also go to my brothers, for guidance in various aspects of life beyond this academic endeavor.

Lastly, I would like to acknowledge the Université Libre de Bruxelles for providing valuable knowledge.

Contents

I Introduction **5**
1 Focus of the thesis . . . 6
2 Structure of the Thesis . . . 7
3 Abbreviations . . . 8
4 Contributions . . . 9
5 Electrophysiological signals . . . 10
5.1 EEG . . . 10
5.2 EMG . . . 10
6 Transfer Learning . . . 12
6.1 Examples of transfer learning . . . 13
6.2 Notions . . . 14
6.3 Negative Transfer Learning . . . 19
6.4 Transfer Learning on electrophysiological data . . . 19

II State-of-the-art **21**
7 Supervised Transfer Learning techniques . . . 22
7.1 Fine Tuning . . . 22
7.2 Transfer Forest Classifier . . . 23
7.3 TrAdaBoost . . . 24
7.4 FA . . . 25
8 Unsupervised Transfer Learning techniques . . . 26
8.1 CORAL . . . 26
8.2 KLIEP . . . 27
8.3 SA . . . 27
9 Divergence measure between domains . . . 29
9.1 PAD . . . 29

III Methods **31**
10 Background 32
10.1 Data splitting for classification 32
10.2 Classification 33
10.3 Enhancement 34
11 Dataset 35
12 Preprocessing 37
12.1 L2-normalization 37
13 Data visualization 37
13.1 PCA 38
13.2 UMAP 38
14 Transfer Learning models 39
14.1 The choice of the algorithms 39
14.2 The baseline models 40
14.3 ML libraries 40
15 The accuracy method 41
16 Experiments 43
16.1 Increment test 44
17 Significance tests 45
17.1 Friedman test 45
17.2 Nemenyi test 46

IV Results and discussions **47**
18 Data visualization results 48
19 Transfer Learning performance 50
19.1 Supervised both hands estimation 50
19.2 Supervised single hand estimation 53
19.3 Supervised estimation between hands 56
19.4 Unsupervised single hand estimation 64
19.5 Divergence measurements 68
20 Discussion 70

V Conclusion **74**
21 Summary of Transfer Learning results 75
22 Future research 76

Part I

Introduction

1 Focus of the thesis

Today's technologies allow us to observe the electrophysiological signals of the human body. Machine learning models try to use them to tackle a wide range of problems. One of those problems is hand gesture estimation. While there are a multitude of possible use cases for gesture estimation, inlcuding sign language translation and human-computer interaction, an important use case is robotic arm prostheses. It can allow humans with disabled hands to interact with the world again by using prosthetic hands.

The process of gesture recognition requires capturing and interpreting the signals of the hands, which can be achieved through electromyography. In this context, electropmyography serves as a tool, recording the activities of the hand muscles. Those signals can be used to address a problem, known as intra-subject gesture classification, which involves recognising a set of possible gestures within the same subject. However, we aim to solve a more challenging problem which is inter-subject classification. Its importance is described as follows:

1. Realistic application scenarios: Inter-subject classification is important in real-world applications where a model trained on a group of subjects must be applied to a new, unseen subject.

2. Avoiding data collection phase: The creation of personalised models requires extensive data collection from the specific subject. In some cases, like when a subject is missing a hand, this data collection might not be possible or practical. A model that can generalize across subjects would bypass this limitation.

However this problem is hard to solve because the electrophysiological signals are subject to a lot of variability between different individuals. This can be due to differences in muscle structure, movement patterns, skin conductivity, and other physiological factors. This high degree of variability makes it challenging for a model to generalize its predictions to new, unseen subjects.

To tackle this problem we would want to take advantage of the previously acquired training "knowledge". The field of machine learning tackling this problem is called Transfer Learning. The thesis looks at different methods used in Transfer Learning, and those that work with electrophysiological data. The main part will apply some Transfer Learning techniques in the inter-subject setting. We will try to answer whether performance improvement can be achieved in this setting using the chosen Transfer Learning techniques.

2 Structure of the Thesis

The thesis is divided into several parts:

Introduction We begin with an overview of the thesis topic and mention the reasons for conducting this research. We also define key terms and concepts used in the study. We start with an introduction to Electrophysiological signals, explaining what they are and why they're important for our research. Next, we introduce the idea of Transfer Learning, discussing what it is and how it's used. We explain the different problems and categories in Transfer Learning. Finally, we present the main challenge of this study: the application of Transfer Learning in the context of Electrophysiological data.

State-of-the-art This section talks about some state-of-the-art Transfer Learning methods. We provide brief descriptions of some interesting Transfer Learning algorithms.

Methods Initially, we describe the Transfer Learning models used, as well as the methodology for assessing performance in different settings. We introduce the electrophysiological dataset made by Colot[10], and showcase its classification performance across various settings. We've used supervised and unsupervised Transfer Learning algorithms for different applications, such as: both hands, single hand and inter-hands settings. Finally, we present baseline models for performance comparison.

Results and discussions We apply transfer learning techniques on the dataset. We then proceed by evaluating the performance of the models in different settings.

Conclusion We summarise the results and make a discussion about some possible future reasearch.

3 Abbreviations

EEG Electroencephalography

EMG Electromyography

sEMG Surface electromyography

iEMG/imEMG Intra-electromyography

LDAC/LDA Linear discriminant analysis classifier

SVM Support vector machines

MLPC/MLP Multi-layer perception classifier

NN Neural network

GP Gaussian process

SVR Support vector regression

PNN Progressive neural network

ML Machine learning

TL Transfer learning

NTL Negative Transfer Learning

NTG Negative Transfer Gap

RAW Raw data

RMS Root mean square

CORAL CORrelation ALignment

ULB Université libre de Bruxelles

CNN Convolutional Neural Network

WL Waveform length

MAV Mean Absolute Value

ZC Zero Crossing

WA Wilson Amplitude

MAA Maximum Absolute Amplitude

KNN K-nearest neighbors

RF Random Forest

SER Structure expansion/reduction

STRUT Structure transfer

DG Divergence gain

IG Information gain

SA Subspace alignment

FA Feature augmentation

KLIEP Kullback-Leibler Importance Estimation Procedure

PCA Principal component analysis

DA Domain adaptation

L2-norm Least squares normalisation

UMAP Uniform Manifold Approximation and Projection

PAD Proxy A-distance

CD Critical Distance

4 Contributions

- We implemented a methodology for assessing the performance of Transfer Learning algorithms on an EMG dataset. Inspired by the well-known leave-one-out method, our approach involves a leave-one-subject-out strategy with one session omitted for transfer learning. Its flexibility has been demonstrated by applying it in various settings, including inter-subject, session-subject (intra-subject), using data from both hands and a single hand and transfer between hands(sections 19.1, 19.2, 19.3).

- We applied this methodology using various Transfer Learning models, including both supervised and unsupervised algorithms, as well as parameter-based, feature-based, and instance-based algorithms, making use of pretrained models or not making use of pretrained models.

- We confirmed that applying Transfer Learning to a target subject in the inter-subject setting improves the performance of the models, particularly when using a small amount of data. Moreover, data from a single hand is sufficient for training either the same hand or the other one, thus greatly reducing the amount of data required at the collection stage.

- We also visually and numerically demonstrated one of the challenges in achieving good Transfer Learning performance - the disparity between the domains of different subjects. Visually and numerically, the difference between the domains of different subjects is greater than the difference between the domains of various sessions from the same subject.

5 Electrophysiological signals

5.1 EEG

Electroencephalography is a technique used to measure the activity of the brain(neuron firing). It is non-invasive as it applies electrodes on the surface of the scalp. All electrodes are linked to an amplifier, which enhances the voltage difference between the reference and measurement electrodes, taking the ground electrode as a reference point. It requires the use of a gel to facilitate measurement of the electrical activity[27]. EEG can be used in a number of applications, such as those related to healthcare (e.g., supporting clinical diagnosis) and gesture prediction. Making a diagnosis is not an easy task, as it requires multiple measurements, such as motor movements. Machine Learning (ML) algorithms can assist in making decisions by providing accurate predictions, even in the early stages of a disease[30]. Although we did not incorporate EEG data in our Transfer Learning experiments for gesture prediction, the inclusion of such data could potentially enhance the process.

5.2 EMG

Electromyography is the recording and analysis of the electrical activity of the skeletal muscles(muscles that are under voluntary control). First, the electromyogram signal is extracted. It is then amplified hundreds of times and transmitted to a display.[40]
Mainly, there are two electromyography recording technologies:

1. **sEMG** - surface EMG, uses electrodes placed on the surface of the skin.

2. **iEMG/imEMG** - intramuscular EMG, uses electrodes placed inside the muscles.

sEMG are also subject to displacement of the skin on the muscles. Thus, the recorded muscle can change during the gesture. However iEMGs are more difficult to use since they require surgery.[44]

Robotic hand prosthesis

sEMGs are mainly used for robotic hand prosthesis. There are some commercially available prosthesis[8] that use sEMG signals as input: i-limb, Bebionic, ultra revolution, LUKE arm, Michelangelo hand, hero arm, taska hand and Vincent evolution 3. The non-invasive nature of sEMG makes it suitable for a wide range of commercial use cases, including robotic hand prostheses.

Hand gesture estimation

The estimation of the hand gesture can be done using Machine Learning techniques. Classification and regression can be both implemented. Classification recognizes a set of gestures while regression should recognize any movement of the hand. Popular models used for classification include: Linear discriminant analysis classifier (LDAC)[16], Support vector machine (SVM)[36], Multilayer perception classifier (MLPC)[17]. Popular models used for regression include: Neural network (NN)[34], Gaussian process (GP)[34], Support Vector regression (SVR)[2]. Performance tests were implemented by Colot [10], which showed the performances of the algorithms in different data splitting settings (Figure 8).

Time series analysis for classification

In the task of estimating gestures from EMG signals, a specific segmentation approach that divides the signals into discrete samples and windows is utilized[10]. This division into samples and windows involves creating intervals of fixed duration. The concept is illustrated in Figure 1, which presents a diagram of an electromyogram with the applied segmentation.

In order to train a classification model, the extraction of specific features from the sEMG signal is essential. These features can be derived either from the time domain or the frequency domain. In this thesis, our focus is on the time domain features. Each channel of the sEMG signal is collected in

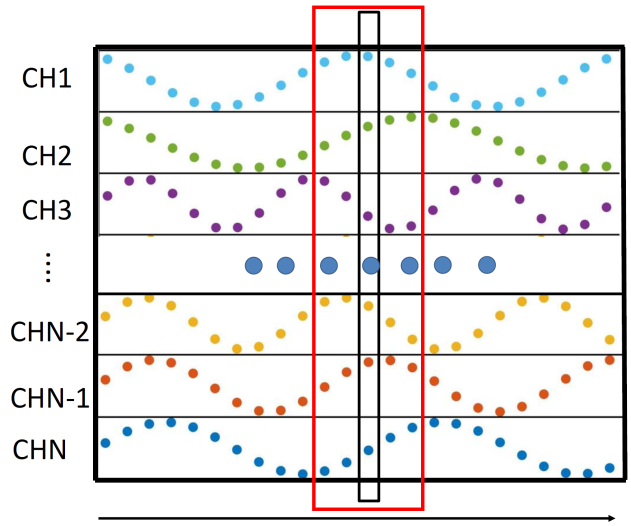


Figure 1: Example of an EMG input signal with a selected sample(black rectangle) and a selected window(red rectangle). The electrical activity is represented as a time series with values(usually mV) for each channel. The arrow indicates the direction of time(x-axis).[51]

the time domain. When examining a specific window of this signal, features can be extracted through a variety of methods and techniques. However, we will refrain from delving into it here. The selected features we concentrate on include the Mean Absolute Value(MAV), Root Mean Square(RMS), Waveform Length(WL), Zero Crossing(ZC), Wilson Amplitude(WA), Max Absolute Amplitude(MAA), Integral. Experiments were conducted that revealed that the classification of hand gestures using these features yields good results [17].

6 Transfer Learning

Transfer Learning is a subfield of Machine Learning. Its aim is to enhance a learning model in one domain by leveraging information from a related domain. In the world of machine learning, traditional techniques vary widely, encompassing supervised learning (which requires labeled data for training), semi-supervised learning (where the training data are only partially labeled), and unsupervised learning (where no labels are available). These techniques often demand significant quantities of data to function effectively. In scenarios where sufficient data in the target domain are lacking, Transfer Learning emerges as a potential solution. It allows a model to improve its learning process by taking advantage of existing data from another domain, provided that there is a pertinent relationship between the two domains. This connection enhances the model's performance in the target domain even with limited data[24].

6.1 Examples of transfer learning

In order to familiarize ourselves with Transfer Learning, we showcase some examples.

Image Recognition: A straightforward example is found in the domain of image recognition[20]. A model initially trained to identify dogs can be further trained with Transfer Learning to identify cats. This transition is possible because both tasks involve recognizing animal features, shapes, and textures.

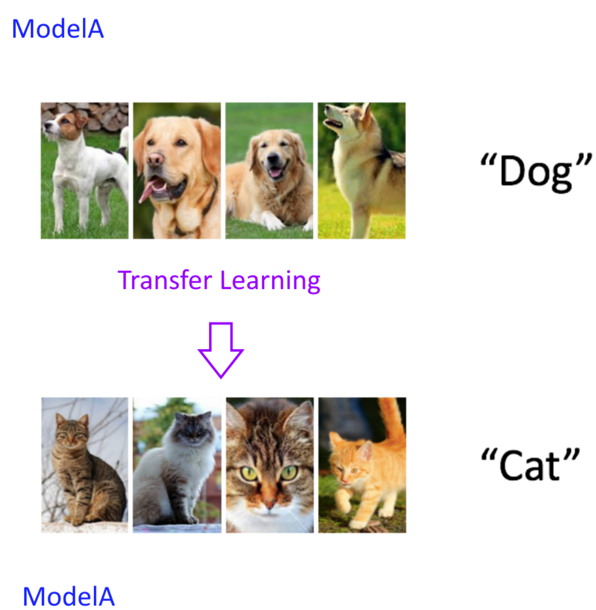


Figure 2: Example illustrating Transfer Learning transforming a "Dog" recognizing model into a "Cat" recognizing model[22].

Speech Recognition: Another instance can be seen in speech recognition[25][47]. A model developed to recognize German speech might be adapted with Transfer Learning to recognize Dutch speech. Since German and Dutch share certain phonetic and linguistic characteristics, the knowledge from the German model can be leveraged to recognize Dutch.

Real-World Simulations: Last but not least, Transfer Learning plays an important role in real-world simulations. Often, it is easier to gather data from a digital model than from the real world. An example can be found in autonomous car driving[38]. First, a virtual world mimicking the real world is created, complete with a virtual car. A model is trained in this digital environment, and then Transfer Learning is used to adapt that model to real-world applications.

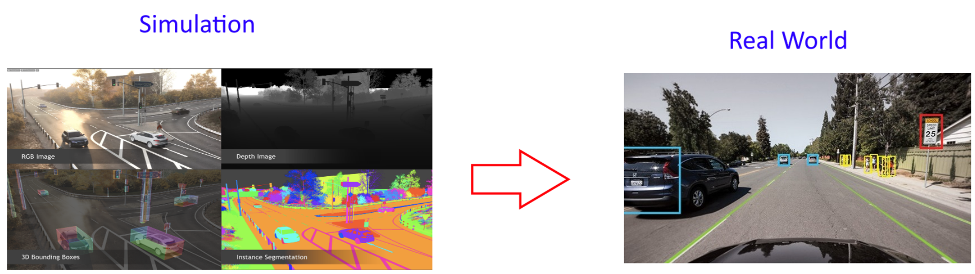


Figure 3: Example illustrating Transfer Learning from a simulated environment to the real world. The images[12][41] are used for illustration purposes and are not necessarily related.

6.2 Notions

Dataset

A dataset is defined as a collection of instances X:

$$X = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$$

where $x^{(j)}$ is the j-th instance in the dataset and m is the total number of instances in the dataset.

An instance x is a specific point or feature vector, described as:

$$x = \{x_1, x_2, \ldots, x_n\}$$

where n denotes the dimensionality, indicating the number of features in the instance.

In supervised learning, a dataset may also include a set of labels Y, where each label $y^{(j)}$ corresponds to an instance $x^{(j)}$. The labels provide the target outputs or classes for each instance, essential for training and evaluating predictive models. The set of labels is defined as:

$$Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}$$

where each $y^{(j)}$ is the label associated with the j-th instance.

Domains

A domain $\mathcal{D}$ is characterized by two main components:

- **Feature Space** ($\mathcal{X}$): This represents the set of all possible feature vectors. Each vector, denoted as x, is an instance in this space.

- **Marginal Probability Distribution** $P(x)$: Describes the likelihood or distribution of observing a particular instance x within the feature space $\mathcal{X}$.

The domain is formally represented as:

$$\mathcal{D} = \{\mathcal{X}, P(x)\}$$

Tasks

A task $\mathcal{T}$ represents the objective we aim to address using the instances from the domain. It encompasses:

- **Label Space** ($\mathcal{Y}$): Constitutes the set of all potential labels or outcomes. Each label, represented as y, is an instance in this space.

- **Predictive Function** $f(\cdot) = P(y|x)$: Denotes the probability of a label y given a feature vector x.

Formally, a task is characterized as:

$$\mathcal{T} = \{\mathcal{Y}, P(y|x)\}$$

Definition

Consider:

- $\mathcal{D}_S$ as the source domain from which knowledge is drawn.
- $\mathcal{T}_S$ as the source learning task aligned with the source domain.
- $\mathcal{D}_T$ as the target domain to which knowledge is applied.
- $\mathcal{T}_T$ as the target learning task associated with the target domain.
- $f_T(\cdot)$ as the target predictive function we aim to enhance.

> **Transfer learning** aims to enhance the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge acquired from $\mathcal{D}_S$ and $\mathcal{T}_S$. This is applicable when $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

[29]

Transfer Learning problems

Given $\mathcal{D}_S$, $\mathcal{T}_S$, $\mathcal{D}_T$ and $\mathcal{T}_T$, there are five main Transfer Learning problems[48]:

1. $$\mathcal{X}_S \neq \mathcal{X}_T$$
   The source and target feature spaces are different. e.g., the two models do syntax correction in different languages.

2. $$P(x_s) \neq P(x_t)$$
   This problem is called *Domain Adaptation*. Source and target marginal distributions are different, e.g., in the spam filtering problem the target user receives significantly different mails than the source user.

3. $$P(y_s|x_s) \neq P(y_t|x_t)$$
   The conditional probabilities of labels given features are different in the source and target domains. This can be expressed in terms of the joint and marginal distributions as $P(y|x) = \frac{P(y \cap x)}{P(x)}$. Any difference in $P(y \cap x)$ between the domains will directly affect $P(y|x)$.

4. $$\mathcal{Y}_S \neq \mathcal{Y}_T$$
   E.g., $\mathcal{Y}_S$ are race labels and $\mathcal{Y}_T$ are colour labels. The two models have different label spaces. One model predicts the race of the cats and another one predicts the eye colour of the cats.

5. $\mathcal{T}_S$ and $\mathcal{T}_T$ have no labels.
   This problem is called *Unsupervised Transfer Learning*. The problem when only $\mathcal{T}_T$ has no labels is called *Transductive Transfer Learning*.

More broadly, if $\mathcal{X}_S = \mathcal{X}_T$ then the problem is part of a more generalized *Homogenous Transfer Learning* problem. By contrast, if $\mathcal{X}_S \neq \mathcal{X}_T$, then the problem is part of a *Heterogeneous Transfer Learning* problem.

Transfer Categories

In the context of Transfer Learning, various approaches are designed to address those problems. An essential aspect of Transfer Learning is identifying the type of knowledge being transferred between domains. There are four main categories of Transfer Learning(related to what is being transferred)[48]:

- **Transfer learning through instances**. Transfer learning through instances involves reweighting instances from the source domain to minimize the differences between the source and target domains. This approach often assumes that the discrepancies in the distributions are primarily attributed to sample selection bias[53][52] where proper randomization of data is not achieved. By aligning the distributions through reweighting, the model can leverage the adjusted instances from the source domain to learn more effectively when applied to the target domain. These reweighting algorithms work best when the conditional distribution is the same in both domains.

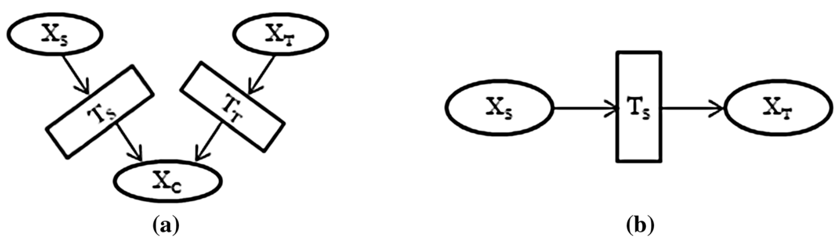
- **Transfer learning through features**. This category has two subcategories (Figure 4):

  1. **Asymmetric feature transformation**. It is about globally changing the weights of the features in the source domain (e.g., by computing second-order statistics such as covariance matrices[46]) to match the features of the target domain.

2. **Symmetric feature transformation**. It is about trying to discover meaningful structures between the domains to find a common feature space that has predictive qualities and reduces the marginal distribution between the domains. It assumes that there exists a common space in which distribution discrepancies of different domains can be minimized[48].

- **Transfer learning through shared parameters**. Parameter-based approaches in transfer learning facilitate the transfer of knowledge by leveraging the parameters of the model. These approaches operate under the assumption that the distributions of model parameters across different domains are either identical or exhibit significant similarity[1][26].

- **Transfer learning through relationship**. It is about trying to transfer knowledge knowing the existence of some relationship between the source and target domains. Relation-based approaches operate on the assumption that certain internal logical rules present in the source domain persist in the target domain.[48]



[48]

Figure 4: (a) The symmetric transformation (b) the asymmetric transformation

Homogenous Transfer Learning

In Homogenous Transfer Learning there are three approaches[48]:

1. Correction of the marginal distribution difference in the source
2. Correction of the conditional distribution difference in the source
3. Correction of both the marginal and conditional distribution differences in the source.

Heterogeneous Transfer Learning

The majority of approaches in Heterogeneous Transfer Learning can be described in two general steps[48]:

1. If domain distributions are not the same, then domain adaptation
2. If domain distributions are the same, alignment of the input spaces of the source and target domains

6.3 Negative Transfer Learning

Negative Transfer Learning (NTL) refers to the situation where the transfer of knowledge from the source domain to the target domain leads to a decrease in learning performance. This phenomenon can be formally defined as[54]:

$$\epsilon_T \left(\theta(x_s, y_s; x_t, y_t)\right) > \epsilon_T \left(\theta(\emptyset; x_t, y_t)\right),$$

Here, ϵ_T denotes the error rate on the target domain, θ represents the learning algorithm being used, x_s and y_s are the features and labels in the source domain, respectively, x_t and y_t are the features and labels in the target domain, respectively. $\theta(\emptyset; x_t, y_t)$, with the empty set, indicates that no information from the source domain is present.

In other words, the algorithm's performance on the target domain is worse when utilizing information from the source domain compared to when the source information is ignored altogether.

The Negative Transfer Gap (NTG) provides a metric to quantify this phenomenon[54]:

$$NTG = \epsilon_T \left(\theta(x_s, y_s; x_t, y_t)\right) - \epsilon_T \left(\theta(\emptyset; x_t, y_t)\right).$$

Where a positive value indicates that negative transfer has occurred. When using unsupervised algorithms, it is not possible to compare to a target only algorithm, however, it is still possible to compare to source only if source labels are available. It's worth noting that in the unsupervised problem, it is impossible to compute the NTG if the labels lack completely.

6.4 Transfer Learning on electrophysiological data

Machine Learning models can be trained on electrophysiological data from either single or multiple subjects. The purpose of training on EMG data is

to decode and interpret the underlying muscular activities, enabling applications such as gesture recognition. Gesture recognition can further lay the foundation for the development of hand prostheses. However, the high inter-subject variability (e.g., Figure 5) poses significant challenges. Furthermore, training a new model for a new subject takes resources and time, thus making it less comfortable for the end user. Additionally, it might be impossible to collect the necessary data if the subject has no hand.
Transfer Learning comes handy to the situation as it tries to solve related problems. For example, in Figure 5, we would want to use the data from the first five subjects seen as a single or multiple source domains, in order to improve the performance of a classifier on the sixth subject seen as the target domain.

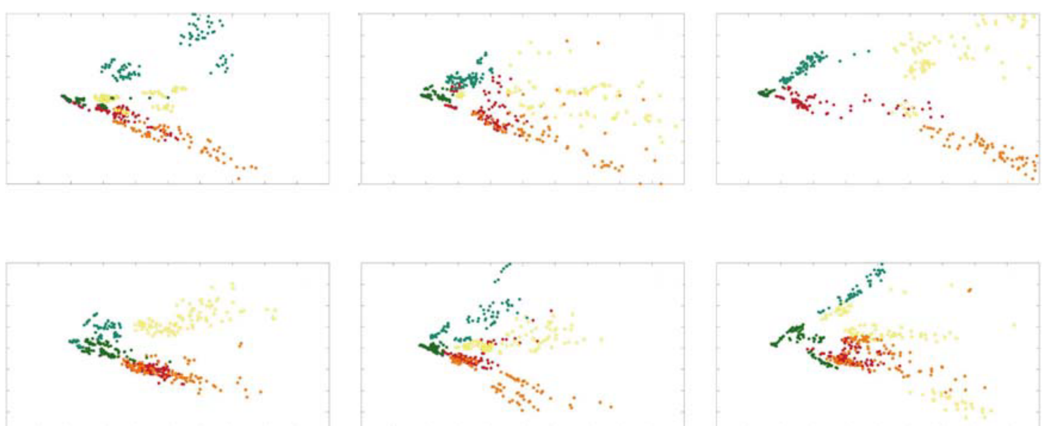


Figure 5: sEMG signals distribution, represented in 2D PCA projections. The signals are collected from six different users, each performing five distinct hand gestures.[55]

Part II

State-of-the-art

7 Supervised Transfer Learning techniques

We introduce Fine Tuning, the most popular technique in Transfer Learning. Although we did not use it, it is widely used[48] and today is even more prevalent with the advent of Large Language Models (LLMs) [35][21]. We find it to be an important and simple technique in transfer learning and a must to showcase as an introduction to the reader. However, there exist a multitude of Transfer Learning algorithms[48], our focus is primarily on selected Homogeneous Transfer Learning algorithms from the ADAPT python library[32]. This library provides state-of-the-art, easy to use Transfer Learning algorithms.

7.1 Fine Tuning

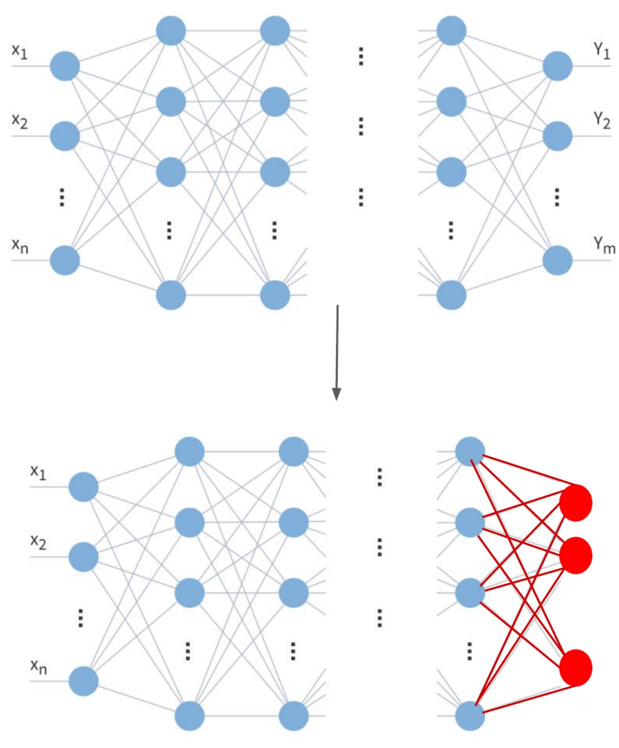


Figure 6: Fine Tuning by replacing the last layer and having a different output layer[20]

Fine Tuning is about training a model using a pre trained one[20]. We already talked about different Transfer Learning problems and transfer categories(what information is being transferred). It is important to know about the techniques used to do Transfer Learning in practice. It is not a secret that Neural Networks are popular nowadays, thus the reason why Fine Tuning is one of the most used Transfer Learning techniques.
Fine Tuning is a technique that can be applied to both homogenous and Heterogeneous Transfer Learning problems. It can also be applied on models with different label spaces. The main idea intuitively related to knowledge transferring is to initialize the weights of the target network with the weights of the pre trained one. The next step is to reset the last layer, intuitively we want to predict new/different labels. A simple example is illustrated in the Figure 6, where the label spaces are different. The idea is to replace the old layer by a new one and reset its weights. We could choose to retrain more than just the last layer, but [50] suggests that is will be less efficient.

However it is usually not enough as there still are too many weights for the amount of data available and it over fits. Thus, the final step is to freeze some initial layers. This will allow retaining feature selection knowledge already acquired in lower layers and train with fewer weights, thus avoid over fitting.

7.2 Transfer Forest Classifier

The TransferForestClassifier is a set of supervised parameter-based transfer learning algorithms for Random Forests. It consists of three distinct algorithms: RELAB[32], the Structure Expansion/Reduction (SER)[43] and the Structure Transfer (STRUT)[43]. We've only used RELAB and STRUT because SER was too slow to use on the available hardware.

RELAB

RELAB is a method designed for conditional distribution correction in decision trees, allowing for adjustment of node values according to target data without altering the tree structure.

Algorithm Steps:

1. For each node in the decision tree, compute the class distribution of the target data that falls into the node.

2. Update the value of each node (including internal nodes and leaves) to reflect the new class distribution.

Thus, this technique utilizes the existing tree structure to partition the target data, without modifying it.

STRUT

STRUT modifies the thresholds within the decision tree to cater to target samples, ensuring that the modifications to the tree structure are kept to a minimum. The underlying idea is that similar domains should yield similar tree structures.

Algorithm Steps:

1. Prune any unattainable nodes in the target domain. An unattainable node in the target domain is a node that can't be reached by any data from the target domain.

2. Adjust the Decision Tree by replacing numeric threshold values with new thresholds $\tau(v)$ at node v. The goal is to optimize the decision threshold with respect to the target training data.

   In order to achieve that, it formulates an optimization problem of maximizing the divergence gain DG, constrained by the information gain IG[15], to choose the threshold for node $\tau(v)$ with respect to target data that can reach v. DG is a measure that quantifies the similarity between the original and the new label distributions at node v. It is based on the Jensen-Shannon divergence[28].

3. Determine the final decision value at each leaf based on target training data similarly to RELAB.

7.3 TrAdaBoost

TrAdaBoost[13] is a boosting algorithm that extends upon the AdaBoost[19] algorithm to tackle transfer learning problems. It is a supervised instance based and model agnostic transfer learning method. TrAdaBoost is using an iterative process to train weak classifiers. It adjusts the weights of training data to focus on misclassified samples. It assumes that while the source and target domain data utilize the same set of features and labels, the distribution of these data is different across the two domains. This divergence in data distribution suggests that some source domain data could be beneficial for learning in the target domain, while other data might not be useful or could even be detrimental. To account for this, TrAdaBoost works by reweighting source domain data to minimize the influence of harmful source data and magnify the contribution of beneficial source data to the target domain. For each iteration round, the base classifier is trained on the weighted source and target data, and the error is only calculated on the target data. Moreover, for each iterative round, if a training instance from a different distribution is erroneously predicted, its training weight is reduced. Over time, the instances that align better with the same-distribution ones will attain larger training weights, while those that are dissimilar will receive lower weights. The final classifier is constructed through weighted majority voting from all the trained weak classifiers.

Algorithm 1 TrAdaBoost

Initialize weights w_i^1 for $i \in X_s, X_t$ with values specified by the user.
for $t = 1, 2, ..., N$ **do**
  Train a weak learner h_t using w_i^t, apply h_t to all x_i in X_s and X_t.
  Compute error rate $e_t = \frac{\sum_{i \in X_t} w_i^t |h_t(x_i) - c(x_i)|}{\sum_{i \in X_t} w_i^t}$, set $\beta_t = \frac{e_t}{1-e_t}$ and $\beta = \frac{1}{1+\sqrt{2 \ln n/N}}$.
  Update weights $w_i^{t+1} = w_i^t \beta^{|h_t(x_i) - c(x_i)|}$ for $i \in X_s$ and $w_i^{t+1} = w_i^t \beta_t^{-|h_t(x_i) - c(x_i)|}$ for $i \in X_t$.
end for
Final model: $h_f(x) = \begin{cases} 1, & \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-h_t(x)} \geq \prod_{t=\lceil N/2 \rceil}^{N} \beta_t^{-\frac{1}{2}} \\ 0, & \text{otherwise} \end{cases}$.

7.4 FA

Feature Augmentation (FA), introduced by Hal Daumé III in "Frustratingly Easy Domain Adaptation"[4], represents a simple transfer learning method. This supervised method enhances the feature space for both source (X_s) and target (X_t) data, facilitating model transferability across various domains.

- Source input features (X_s): Each source feature vector is transformed into a new format $(X_s, 0, X_s)$, with a middle segment of zeros matching the dimension of X_s and X_t.
- Target input features (X_t): Each target feature vector is transformed into $(0, X_t, X_t)$, with the initial segment being a null vector of the same size as X_s and X_t.

The meaning of this transformation is described as following:

1. Specific source features: The first part of the augmented vector represents features unique to the source domain, capturing specific behaviors in the source data.
2. Specific target features: The second segment captures target domain-specific features, reflecting unique behaviors in the target data.
3. General features: The third segment consists of features common to both domains, representing generalizable behaviors.

This model is easily generalizable for multisource transfer learning. Thus, contrary to other algorithms, for this algorithm we will always consider each subject (data from each subject) to be a distinct domain.

8 Unsupervised Transfer Learning techniques

This section presents the algorithms used for unsupervised transfer learning. As previously mentioned, the algorithms are implemented in the ADAPT Python library [32].

8.1 CORAL

CORAL[46] stands for *Correlation Alignment.* In contrast to the previous state-of-the-art techniques mentioned, it is an unsupervised, model agnostic, state-of-the-art transfer learning technique, which means that the target data has no labels. It works by aligning the distributions of the source and target features. It uses the covariance in order to achieve it.
First of all the feature spaces are normalized. Next, the distance of the covariance matrices of the source and target feature spaces is minimized. The algorithm is divided in two parts:

1. Whitening of the source data
2. Recolouring it with the target covariance.

The entire algorithm can be described as following :

Algorithm 2 CORAL for Unsupervised Domain Adaptation

1: $C_S = \text{Covariance}(X_s) + \text{IdentityMatrix}(\text{NumberOfFeatures}(X_s))$
2: $C_T = \text{Covariance}(X_t) + \text{IdentityMatrix}(\text{NumberOfFeatures}(X_t))$
3: $new\_X_s = X_s \times \text{InverseSquareRoot}(C_S)$
4: $X_{\text{StarS}} = X_s \times \text{SquareRoot}(C_T)$

8.2 KLIEP

The Kullback-Leibler Importance Estimation Procedure (KLIEP)[45] is an unsupervised instance-based Transfer Learning method.

The idea centers around estimating the ratio between the source and target distribution probabilties. This ratio is called the importance function. The algorithm takes the following steps:

1. Modeling the importance function: it uses a linear or kernel-based model to represent the importance function, resulting in a convex optimization problem.
2. Minimize domain divergence: it optimizes the model parameters to minimize the divergence between the estimated and true target input densities.
3. Cross-Validation for model selection: it implement a cross-validation method using target input samples to select the model, enhancing accuracy due to the abundance of target data.

8.3 SA

Subspace Alignment (SA)[18] is a method designed for unsupervised Transfer Learning. Similarly to CORAL, it is a feature based transfer learning method. It uses PCA for its initial step.

Subspace generation

To handle more robust representations of the source and target domains, SA uses PCA to transform every source and target data to a D-dimensional z-normalized vector (i.e., of zero mean and unit standard deviation). It selects the d eigenvectors corresponding to the d largest eigenvalues for each domain. These eigenvectors are used as bases of the source and target subspaces, respectively denoted by X_S and X_T ($X_S, X_T \in R^{D\times d}$). X'_S and X'_T are orthonormal. In the following, X_S and X_T are used to learn the shift between the two domains.

Subspace alignment

Each source (x_S) and target (x_T) data (where $x_S, x_T \in R^{1\times D}$) is projected to its respective subspace X_S and X_T by the operations $x_S X_S$ and $x_T X_T$, respectively. Then, a linear transformation that maps the source subspace to the target one is learned. This step allows the direct comparison of source and target samples in their respective subspaces. A transformation matrix M from X_S to X_T ($M \in R^{d\times d}$) is obtained as $M = X'_S X_T$. This implies that the new coordinate system is equivalent to $X_a = X_S X'_S X_T$. The new coordinate system, referred to as the target aligned source coordinate system, allows the transformation of the source subspace coordinate system into the target subspace coordinate system by aligning the source basis vectors with the target ones.

Algorithm 3 Subspace alignment algorithm in the lower d-dimensional space

1: $X_S = \text{PCA}(X_s; d)$
2: $X_T = \text{PCA}(X_t; d)$
3: $X_a = X_S \cdot X'_S \cdot X_T$
4: $S_a = X_s \cdot X_a$
5: $T_T = X_t \cdot X_T$
6: $Y_{T\_}pred = \text{NN Classifier}(S_a; T_T; Y_s)$

9 Divergence measure between domains

9.1 PAD

Ben-David et al. (2010)[6] introduced the concept of target error for models with 2 different domains. This error depends on three key components: the source error, the divergence $d_{H\Delta H}$ between the data distributions of the two domains, and a term λ.

$$e_T(h) = e_S(h) + d_{H\Delta H} + \lambda$$

Where:

- h is the learned hypothesis.
- $e_T(h)$ represents the target generalization error.
- $e_S(h)$ denotes the source generalization error.
- λ is the error of the ideal joint hypothesis on D_S and D_T, and it's anticipated to be negligible when successful adaptation is achievable.

This formulation suggests that for effective adaptation, it is essential to select a hypothesis that not only performs well on the source domain D_S but also minimizes the divergence between the two domains[18].

The term $d_{H\Delta H}$ provides a measure of the divergence between the data distributions of domains D_S and D_T. At its core it has to identify a specific hypothesis h that minimizes the classification error between samples from D_S and D_T. However, achieving this with exact precision is impractical.

To provide a feasible solution, an approximation technique was introduced. It proposes training a model that differentiates between samples from the source and target domains. The divergence measure obtained through this method is referred to as the Proxy A-distance (PAD).

To compute the PAD, the following steps are taken[39]:

1. Merge the datasets and label each instance(feature vector) based on its domain of origin.
2. Train a classifier on the combined dataset.
3. Measure the classifier's error e using an independent test set.

4. Calculate PAD as $2(1 - 2e)$.

In our case for classificaiton algorithms, we've chosen Random Forests.

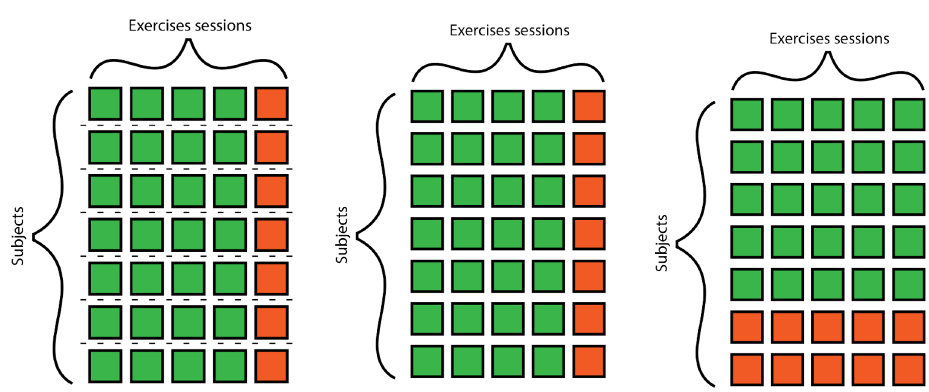
Part III

Methods

10 Background

In this study, we aim to build on the work done at ULB, where a detailed process was set up to create a new dataset[10]. This dataset combines in-sync hand movements with EMG signals from the forearm. This data was tested by creating both classification and regression models, as shown in the figures 8 and 9. In the original study, the classification models showed good quality of the data compared to other datasets used for hand posture classification, and the regression models opened up possibilities for estimating hand gestures with many degrees of freedom from the dataset[10].

10.1 Data splitting for classification



(a) (b) (c)

Figure 7: (a) Intra-subject (b) Inter-subject with known subjects (c) Inter-subject with new subjects [10]

To evaluate the models in a classification problem, three distinct data splitting methodologies are employed, comparing their respective performances:

Intra-subject estimation. This approach utilizes a subject-specific model evaluation by training and testing each model on individual subject data. Here, certain exercise sessions are allocated for model training, and the rest are utilized for testing. The mean estimation score derived from all subjects is then used as a performance metric for the model.

This data splitting strategy focuses on the subject-specific features of the sEMG, which may result in model overfitting to these unique features.

Inter-subject estimation with known subjects. This data splitting strategy trains the model using data derived from all subjects. Again, certain exercise sessions from each subject are set aside for model training, while the remaining sessions are utilized for testing.

The outcome of this data splitting method provides an understanding of whether a model, trained using extensive data drawn from multiple subjects, can yield a superior estimation score than one trained on a single-subject data set and gives an upperbound for inter-subject estimation without adaptation to a specific target subject.

Inter-subject estimation with new subjects. This approach involves training the model using some subjects and then testing it using the remaining subjects. This simulates a real-world scenario in which a new user utilizes an already-existing model.

10.2 Classification

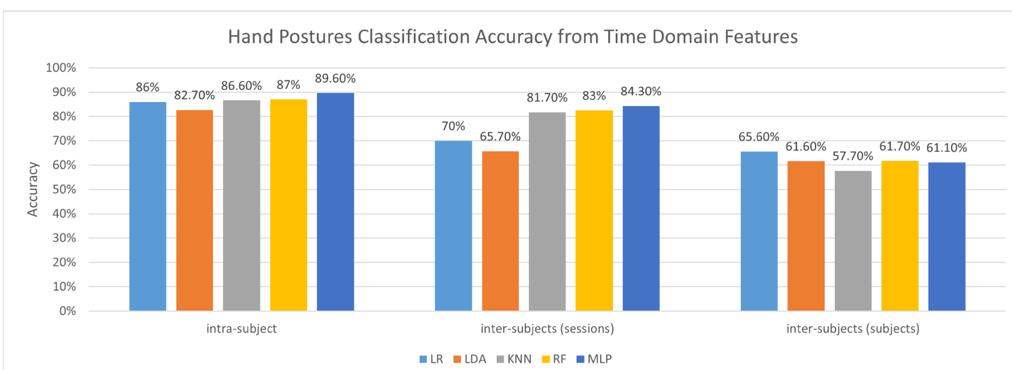


Figure 8: Classification accuracy of the 4 gestures. On the left, the average accuracy for each subject using a 5-fold cross-validation. In the center, the estimation accuracy using all subjects for training and testing with a 5-fold cross-validation on the repetition of the exercise. On the righ,the estimation accuracy using 80% of the subjects for training and the rest for testing with a 5-fold cross-validation.[10]

Experiments were conducted[10] to classify hand postures using recordings of predefined gestures, and five different classification models were trained: LR (Logistic Regression), LDA (Linear Discriminant Analysis), KNN (K-nearest neighbors), RF (Random Forest), and MLP (Multi-layer perceptrons). The tests were performed with different data splitting settings, and the classification accuracy varied across these settings, regardless of the model.

Intra-subject classification achieved an average accuracy of 85.6%, while inter-subject classification resulted in an average accuracy of 73.11% when sessions were isolated, and 60.9% when subjects were isolated. This outcome was expected since the EMG signals exhibit significant variability between subjects.

10.3 Enhancement

Our current research aims to add to this work by using transfer learning techniques. The goal is to improve generalization and speed up learning in the models. We will look at and test various latest techniques in transfer learning, checking how effective they are and if they are suitable for this task.

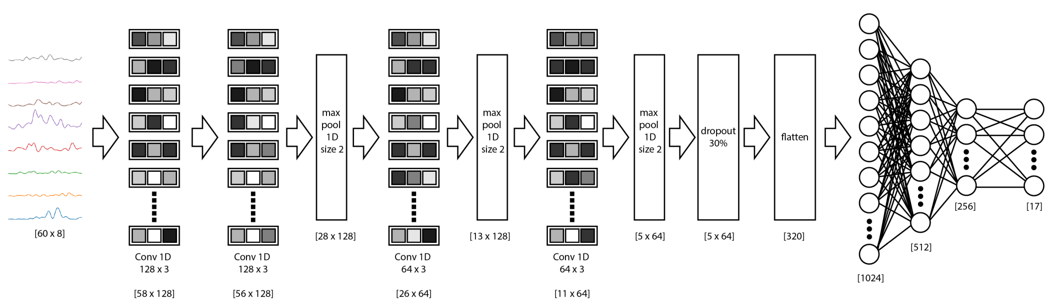


Figure 9: ULB developed a CNN model for regression with 17 neurons in the output layer, corresponding to 17 articulations.[10]

11 Dataset

The dataset has 48 columns. It contains 7 features (Mean Abs. Value, Root Mean Square, Waveform Length, Zero Crossing, Wilson Amplitude, Max Abs. Amplitude, Integral) for each of eight channels. The dataset includes 14 subjects. Each subject performed five sessions with each hand, alternating between right and left, starting with the right. The sessions lasted about five minutes and involved four gestures from the Taiwanese sign language: IDs 2, 7, 19, and 23 (Figure 10). The subjects also did a resting pose (open hand, ID 12) between each gesture. They repeated each gesture six times with both hands in each session. The raw data was collected using a Pico sEMG device(from Cometa[11]).

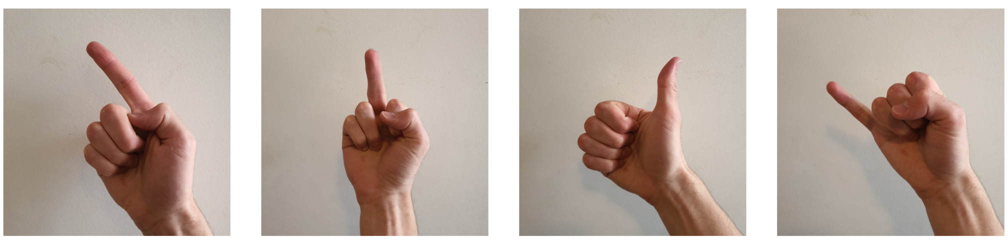


Figure 10: The four gestures of the dataset.[10]

Channels	8
Features per channel	MAV, RMS, WL, ZC, WA, MAA, Integral
Subjects	14
Sessions per subject	Five with each hand
Gestures involved	(IDs 2, 7, 19, 23)
Gesture duration	2 seconds

Table 1: Dataset characteristics.

Label imbalance is a common problem in machine learning where one class has significantly more samples than another class. This can lead to poor performance of the model on the minority class. Fortunately, the dataset is glabally(Figure 11) and on the subject level(Figure 12) balanced.

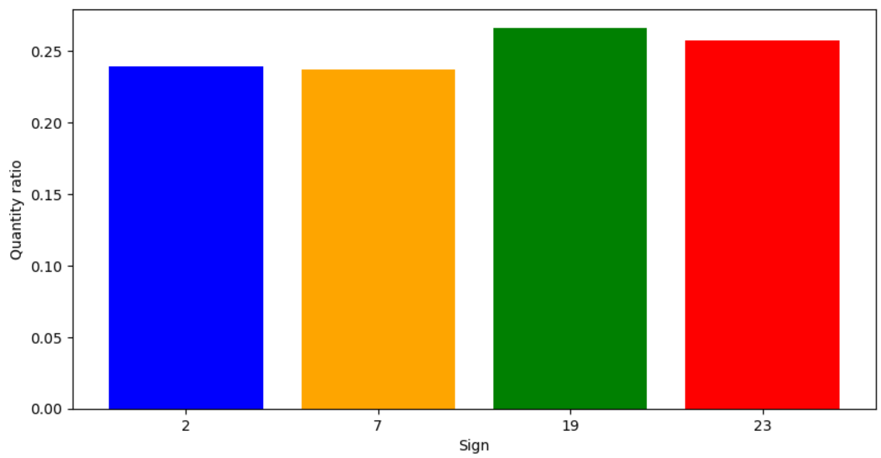


Figure 11: Dataset sign total balance. The pause has been removed from the dataset. Each sample is computed from a window.

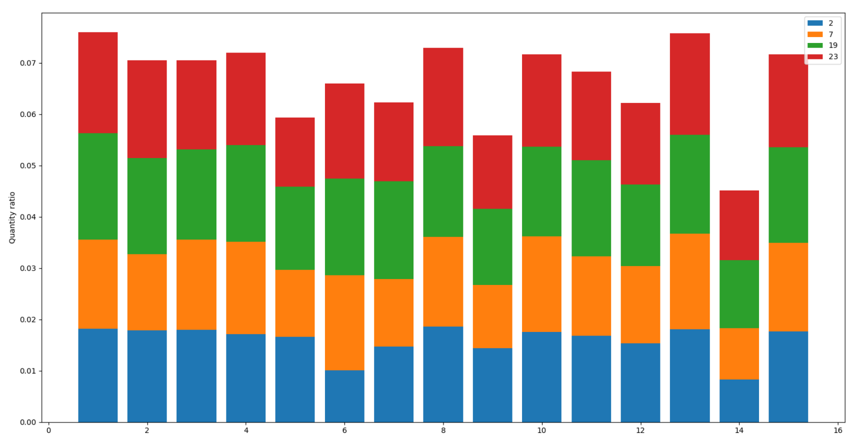


Figure 12: Dataset sign balance per subject. The pause has been removed from the dataset. Each sample is computed from a window.

12 Preprocessing

12.1 L2-normalization

Normalization plays an instrumental role in machine learning data preparation. This technique aims to scale numeric columns in the dataset to a common range, enhancing the performance and stability of machine learning models. We use L-2 normalization to decrease the differences between subjects, as each person has different mulscular capacities, which lead to different amplitudes of the EMG signal. We could use different kinds of normalization (i.e. based on maximum voluntary contraction), but L2 gave the bast results. However, we need to have all the unlabelled target samples available.

L2-normalization[49], also known as least squares normalization, is a specific type of normalization technique. Its primary purpose is to adjust the values in a dataset such that the sum of the squares of each row amounts to one.

For a given dataset, let $\mathbf{x}$ be a feature vector with elements $x_1, x_2, ..., x_n$. The L2-norm (or Euclidean length) of this vector, denoted by $l2\_norm(\mathbf{x})$, is computed as:

$$l2\_norm(\mathbf{x}) = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Then, the L2-normalization of the feature vector $\mathbf{x}$, denoted by $\mathbf{x}_{\text{norm}}$, is obtained by dividing each element of $\mathbf{x}$ by its L2-norm:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x}}{l2\_norm(\mathbf{x})}$$

Most of the classification & transfer learning algorithms in this document will be applied on l2-regularized data.

13 Data visualization

Data visualization plays an important role in assessing data quality. It enables the identification of potential anomalies within the data. Specifically, in the context of EMG data involving various subjects, visualization allows the observation of distribution differences among these subjects. Such insights can be beneficial in the context of Transfer Learning.

Furthermore, we decided to use both Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP) as data visualization algorithms. Our decision is based on the advantages each technique provides. PCA, a linear method, is widely used in various fields. On the other hand, UMAP, a nonlinear technique, has the capability to capture nonlinear structures in the data. Moreover, one of its primary objectives since its conception was data visualization.

13.1 PCA

Principal Component Analysis(PCA) [7] is one of the most popular methods for performing dimensionality reduction of the dataset. It operates by identifying principal components, which are newly formulated variables derived from the original ones. These principal components are linear combinations constructed to encapsulate the maximum possible variance within the dataset. They can be viewed as compressed versions of the original features. The hierarchy of principal components is such that the first principal component accounts for the highest variation in the data. Subsequent components, orthogonal to their predecessors, account for the maximum possible remaining variation. The operational mechanism of PCA involves projecting the data from its native space into a new space, defined by these principal components.

13.2 UMAP

Uniform Manifold Approximation and Projection (UMAP)[33] is a dimensionality reduction technique, widely employed in data visualization. UMAP provides a way to visualize high-dimensional data in a lower-dimensional space. While the method shares some principles with techniques such as Principal Component Analysis (PCA), it offers enhanced scalability, and handles non-linear data structures more effectively.

The core operation of UMAP involves creating a simplified "map" of the higher-dimensional data structure in a lower-dimensional space. The UMAP process consists of two main steps:

1. UMAP first builds a high-dimensional graph. Each data point is a node on this graph. Nodes are linked if their data points are 'close' to each other.

2. UMAP then creates a simpler, lower-dimensional version of this graph. It tries to keep 'close' points close together and 'far' points far apart, just like in the high-dimensional graph.

Key parameters in UMAP are *n_neighbors* and *min_dist*, which significantly influence the balance between local and global structure in the final projection.

n_neighbors, representing the number of approximate nearest neighbors used to construct the initial high-dimensional graph, plays an important role in how UMAP weighs local versus global structure. Lower values lead UMAP to prioritize local structure by limiting the number of neighboring points considered during the high-dimensional analysis. Conversely, higher values guide UMAP to capture broader structural patterns.

min_dist, or the minimum distance between points in the low-dimensional space, determines how closely UMAP clusters points together. Lower *min_dist* values result in denser groupings, while larger values lead to more spread-out clusters, emphasizing the preservation of broader topological structures over dense clustering.

14 Transfer Learning models

14.1 The choice of the algorithms

We selected a variety of Transfer Learning algorithms to explore. Initially, we chose RELAB and STRUT, making use of a pretrained model. As alternatives, TrAdaBoost and FA were selected, which are model agnostic.

Subsequently, we explored unsupervised Transfer Learning algorithms: CORAL, KLIEP and SA. These models don't require any labels from either the source or target domains, and they lay in the category of feature-based algorithms.

For the underlying classification models that paired with the model agnostic Transfer Learning algorithms, our choices included linear models, Multilayer Perceptrons and Random Forests. This progression from simpler to more complex models is well adapted in machine learning. It allows to explore a diverse range of estimations, beginning with fundamental techniques and gradually incorporating more complex methods.

The Table 2 shows a comparison of the Transfer Learning algorithms:

Name	Transfer Category	Source Data	Target Data	Model Agnostic
RELAB	Parameter	Labelled	Labelled	No
STRUT	Parameter	Labelled	Labelled	No
TrAdaBoost	Instance	Labelled	Labelled	Yes
FA	Symmetric feature	Labelled	Labelled	Yes
CORAL	Asymmetric feature	Labelled	Unlabelled	Yes
KLIEP	Instance	Labelled	Unlabelled	Yes
SA	Symmetric feature	Labelled	Unlabelled	Yes

Table 2: Choice of the Transfer Learning algorithms

14.2 The baseline models

Furthermore, we define several baseline models that are useful(mostly in the supervised setting) to compare against the transfer learning algorithms:

- SrcOnly: A source only model is defined which represents a classifier trained on X_S, Y_S (the dataset) without the data from the target subject and tested on the sessions 1 through 4 of the target subject.
- All: When the accuracy method is paired with transfer learning models, the target session data is utilized. This does not present a fair comparison when stacked against the source only model, as the latter doesn't incorporate target session data. Consequently, a more fitting baseline model is required to provide an equitable comparison. To address this need, we initiated classification experiments on the EMG dataset using an enhanced model. This model amalgamates data from the target transfer session with the source data.
- TgtOnly: A classification model trained only on the target subject data.

14.3 ML libraries

The experiments were carried out in Python using libraries such as SciKit Learn[37] and Adapt[32]. SciKit Learn is an open source Python library for machine learning applications. In our case it allowed us to use models such as Random Forests and then compute some metrics. For the TL procedure we used Adapt. It is a package providing some well known domain adaptation methods.

15 The accuracy method

In the context of transfer learning, we've formulated an inter-subject data splitting strategy, as illustrated in Figure 13. We initially pre-trained the model using a selected set of subjects, and then applied transfer learning to the first session of a new subject. The final test set consisted of the remaining sessions of the same new subject. This refined strategy builds upon the inter-subject estimation with new subjects approach.

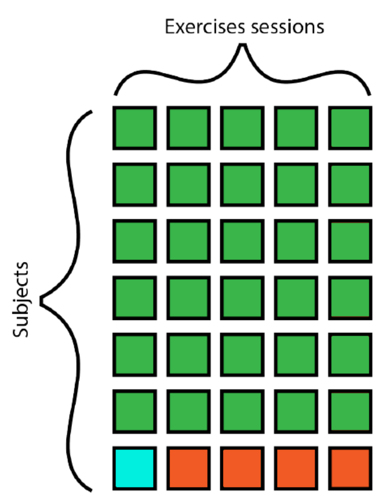


Figure 13: Inter-subject TL splitting[10]

This data splitting strategy is implemented in the following method(Algorithm 4):

1. Loop through each data of the subject
2. Data splitting
3. Pretrain the model

Algorithm 4 Assessing the quality of a TL model on the EMG dataset

```
for different models do
    for each subject do
        X_S,Y_S ← dataset
        X_TL,Y_TL ← subject session 0 data
        X_T,Y_T ← subject sessions 1...4 data
        remove the subject from X_S,Y_S
        pretrain the model on X_S,Y_S
        apply Transfer Learning on X_TL,Y_TL
        compute their respective accuracies on X_T
    end for
    compute the mean accuracy per model
end for
return the mean accuracies of all the models
```

4. Apply Transfer Learning

5. Compute accuracy

However, to test the accuracies of the models (e.g Random Forests) with different parameters(e.g number of trees) an external loop can be added.

The outer loop iterates over a set of models. The inner loop processes each subject in the dataset one at a time. For each subject, the algorithm splits the data into three parts:

- X_S, Y_S: Source data. These are the features (X) and the corresponding labels (Y) for the entire dataset.

- X_{TL}, Y_{TL}: Transfer Learning data. These are the features and labels for the "session 0" data for the subject, which are used as the transfer learning data.

- X_T, Y_T: Test data. These are the features and labels for the "session 1 to 4" data for the subject, which are used as the target or test data.

The algorithm removes the data for the current subject from the X_S, Y_S dataset. This is to ensure that the model does not have prior knowledge of the subject when pretraining and applying transfer learning, which might bias the results. The algorithm then trains each model on the modified dataset (i.e., the dataset minus the current subject's data). The pretrained model is then adapted through transfer learning on the session 0 data for the current subject (X_{TL}, Y_{TL}). After applying transfer learning, the algorithm computes the accuracy of the model on the subject's sessions 1 to 4 data (X_T, Y_T). After the inner loop is finished (i.e., after all subjects have been processed), the algorithm computes the mean accuracy of each model. Finally, the algorithm returns the mean accuracy for each model across all subjects in the dataset. The method is leave-one-subject-out cross-validation.

16 Experiments

Furthermore, the method is applied to assess model performance in different settings, varying based on the composition of source data, transfer learning data, and test data.

First, the method can be applied to data from both hands together. Intuitively, using more data should allow for better performance.

The second application involves using data from only one hand. Testing this will determine if using data from a single hand can perform as well as using data from both hands.

An additional application involves training a model with data from one hand and then applying the learned knowledge to data from the other hand. The dataset is divided into two components: right hand data and left hand data. The left hand data is used for model training, while the right hand data is the target for applying the acquired knowledge. This approach assumes the availability of a sufficient amount of data from one hand, which should enable effective transfer learning and potentially reduce the need to collect data for the other hand.

Another, slightly different application is the intra-subject version of transfer learning between hands, which aims to determine if there is a difference in performance when only using target subject's data, contrary to the inter-subject version where no target subject's data is used in the source.

Lastly, there is unsupervised transfer learning. In this application, labelled data is available from the source task and relatively a bigger amount of unlabelled data from the target task, when compared to other settings. This approach is beneficial as new data can be generated post-training by the end-user. In cases where labeling may not be available, unsupervised methods become useful.

The table 3 summarizes the distinct data settings used in our analysis.

Experiment	**Source data**	**Transfer Learning data**	**Test data**
Single Hand	Right hand data from all subjects, excluding target	Session 0 right hand data of target	Sessions 1-4 right hand data of target
Both Hands	Right and left hand data from all subjects, excluding target subject's right hand	Session 0 right hand data of target	Sessions 1-4 right hand data of target
Inter-Hands Intra-Subject	Left hand data of target subject only	Session 0 right hand data of target	Sessions 1-4 right hand data of target
Inter-Hands Inter-Subject	Left hand data from all subjects except target	Session 0 right hand data of target	Sessions 1-4 right hand data of target
Unsupervised TL(single hand)	Labelled right hand data from all subjects, excluding target	Unlabelled sessions 0-4 right hand data of target	Labelled sessions 0-4 right hand data of target

Table 3: Comparison of Transfer Learning data settings

16.1 Increment test

For all the experiments, we've also made tests with a small amount of Transfer Learning data. The tests represent a gradual increase in the Transfer Learning data, starting at 10% and finishing at 100%, with a 10% increment at each step. This way, we can discern which Transfer Learning algorithms work best with a small amount of data. The 'target only' and the 'All' baselines are also included. For the 'Both hands' setting, the test was carried out for only 10% of the data because producing results for each step would have taken too long on the available hardware.

17 Significance tests

Statistical tests for multiple comparisons were conducted in experiments involving data from multiple algorithms. They allow confirming the significance of the obtained results.

The tests were carried out as follows: for each experiment (Table 3), the first batch of results was computed from the first 10% of Transfer Learning data for each algorithm. Then, another batch of results was computed from the next 10% of Transfer Learning data, excluding data that had already been used. This process continued until 100% of the Transfer Learning data was utilized. In total, 40 batches were produced from 4 different sessions.

The limited amount of Transfer Learning data made it possible to conduct such tests. However, for the "both hands" settings, it was not possible on the available hardware.

Autorank[23] was used to automatically produce the tests. It is a library that simplifies statistical analysis. A detailed explanation of the statistical tests is provided in the following sections.

17.1 Friedman test

The Friedman test[14] is a non-parametric statistical test for repeated measures. This test ranks algorithms based on their performance, from the best to the worst. For this purpose, it assigns average ranks $R_j = \frac{1}{N}\sum_i r_i^j$, which are derived from the original ranks for each dataset out of N. To test the null hypothesis, i.e., to evaluate whether all algorithms perform equally, first the Friedman statistic χ_F^2 and then an improved version of it, the Iman-Davenport statistic F_F are computed:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right) \tag{1}$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{2}$$

Here, k represents the number of algorithms being compared.

17.2 Nemenyi test

If the performance differences between the algorithms are found to be significant (the null hypothesis is rejected, i.e., the equation (2) indicates significance), we can proceed with a Nemenyi test[14] to identify specific algorithms with significant performance differences. In the literature, this type of test is referred to as a post-hoc test.

The Nemenyi test computes a Critical Distance (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

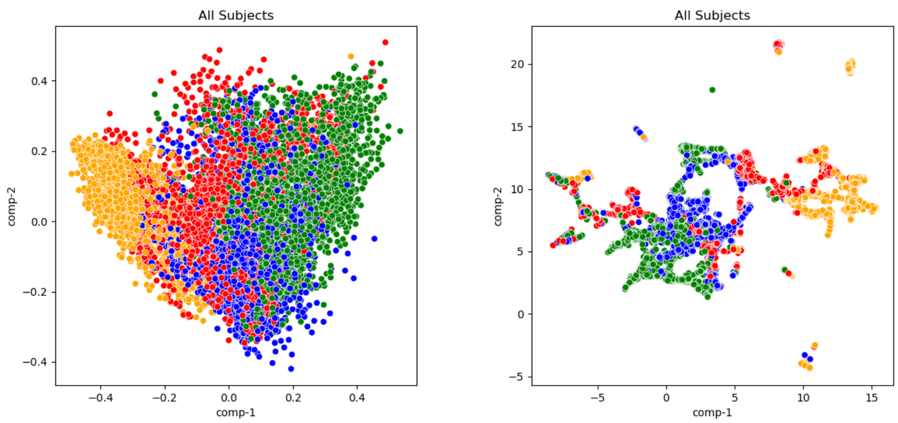
The critical values q_α are from the Studentized range statistic[42] divided by $\sqrt{2}$, k is the number of algorithms, and N is the number of datasets.

The performance of two algorithms is considered significantly different if their average ranks differ by at least the CD.

Part IV

Results and discussions

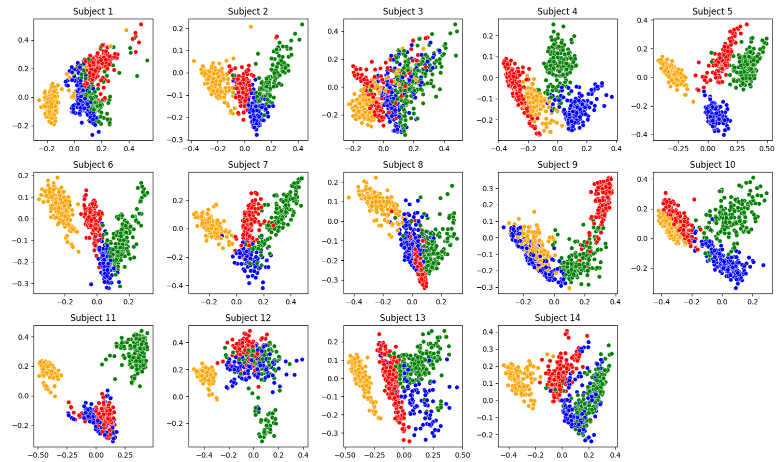
18 Data visualization results



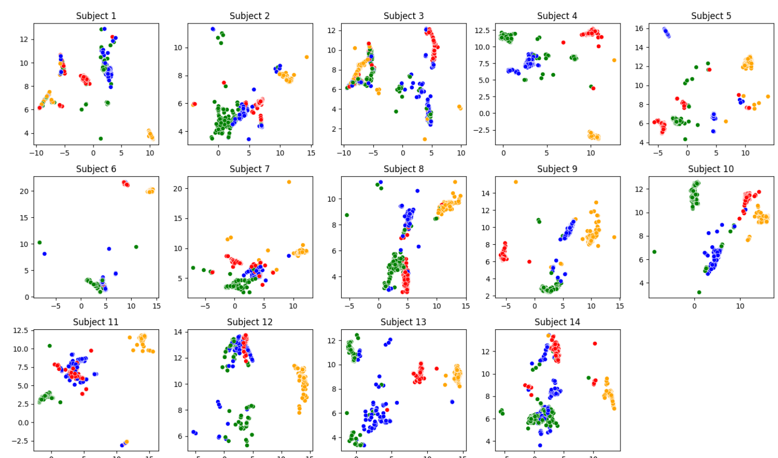
(a) (b)

Figure 14: (a) PCA projections of entire dataset, showcasing only the first two components. (b) UMAP visualization of the entire dataset. Showcasing only the first two components. *n_neighbors*=15 and *min_dist*=0.1 . The sign colors are the same as in the figures 11 and 12. The data are l2-normalised.

The visualization is done by identifying a shared subspace through the analysis of the entire dataset, as illustrated in Figure 14. Subsequently, the focus shifts to examining the individual data projections for each subject, which are derived from this common subspace. Figure 15 shows that, even tho the cluster are separable when looking at a single subject, the clusters locations are different with each subject. Overlapping across different subjects makes the problem more difficult. Both PCA and UMAP present a similar picture. Thus, UMAP confirms that the separation is challenging.



(a)



(b)

Figure 15: (a) PCA projections of the data from each subject of the dataset, showcasing only the first two components. The first component is on the x-axis, and the second component is on the y-axis (b) UMAP visualization of the data from each subject of the dataset. Showcasing only the first two components. $n\_neighbors$=15 and min_dist=0.1 . The sign colors are the same as in the figures 11 and 12. The data are l2-normalised.

19 Transfer Learning performance

19.1 Supervised both hands estimation

We've proceeded with applying the method to both hands together. This setting allows us to observe the model's performance using most of the available data. The source data comprises left and right hand data from all subjects, including the target's left hand data but excluding the target's right hand data.

Algorithm	Accuracy
FA_mlp	0.85
FA_rf	0.83
TrAdaBoost_linear	0.83
FA_linear	0.82
MLP_TgtOnly	0.78
TrAdaBoost_mlp	0.77
TrAdaBoost_rf	0.75
Linear_All	0.75
STRUT	0.74
RF_All	0.74
RELAB	0.72
Linear_TgtOnly	0.71
MLP_All	0.71
RF_TgtOnly	0.71

Table 4: Both hands mean for 10 % of target task data utilized, inter-subject: 1) TgtOnly: Uses only target task data. 2) All: Combines data from both source and target tasks. 3) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 4) Linear: A linear classifier model. 5) RF: Random Forest model with 50 trees. 6) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 7) Data normalization: All data are L2-normalized. 8) Gesture Data: Models use data from right hand performing 4 types of gestures.

The Transfer Learning data consists of the right hand data from the first session of the target subject, while the test data includes the remaining sessions.

FA models excel when using the entire session for Transfer Learning (Figure 16). Target only models and TrAdaBoost with a linear model follow, then "All" baselines, and finally, source-only models. Figure 17 further confirms this trend on a more detailed level. The results are more pronounced for 10 % of Transfer Learning data only(Table 4).

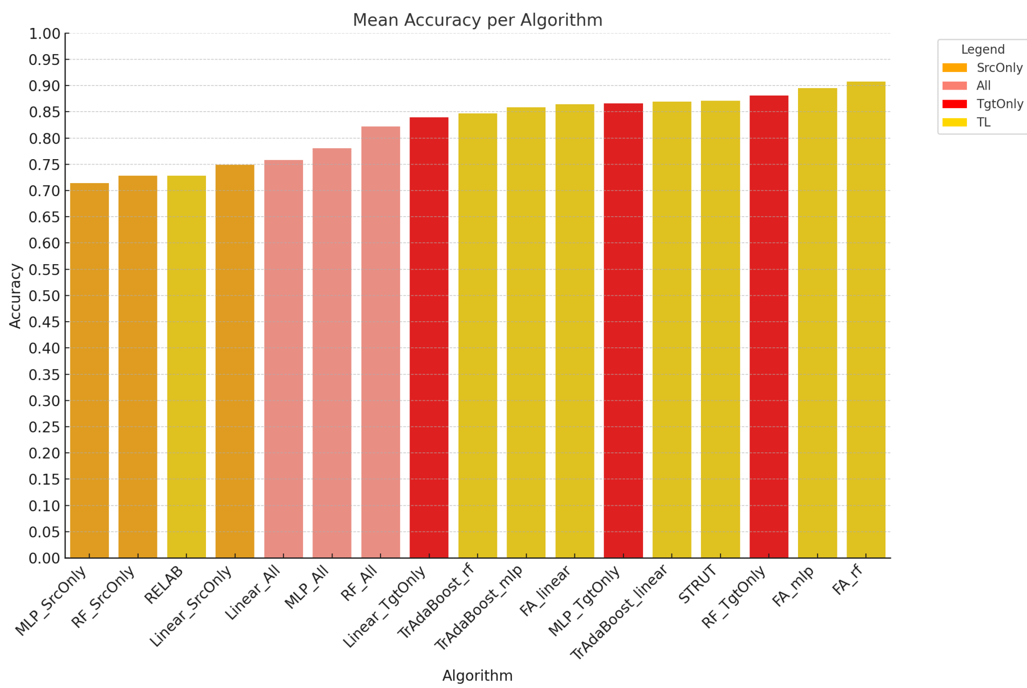


Figure 16: Mean accuracy on both hands data, inter-subject: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) Linear: A linear classifier model. 6) RF: Random Forest model with 50 trees. 7) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 8) Data normalization: All data are L2-normalized. 9) Gesture Data: Models use data from both hands performing 4 types of gestures.

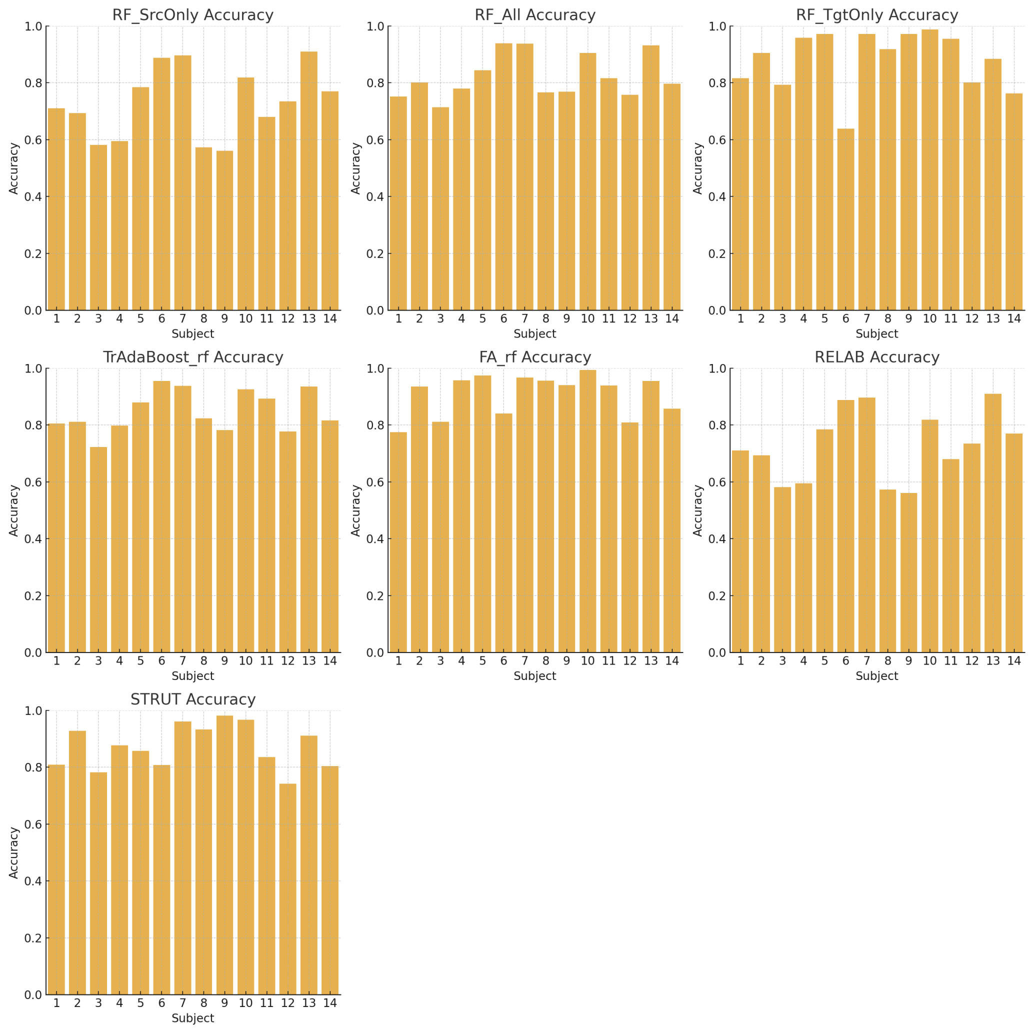


Figure 17: Per subject accuracy on both hands data, inter-subject with RF models: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) RF: Random Forest model with 50 trees. 6) Data normalization: All data are L2-normalized. 7) Gesture Data: Models use data from both hands performing 4 types of gestures.

19.2 Supervised single hand estimation

Contrary to the both-hands setting, in this setting, we use data from the right hand only. It serves as a baseline against both-hands settings, which should allow us to see if using more data, but from a different hand, improves the performance of the algorithms.

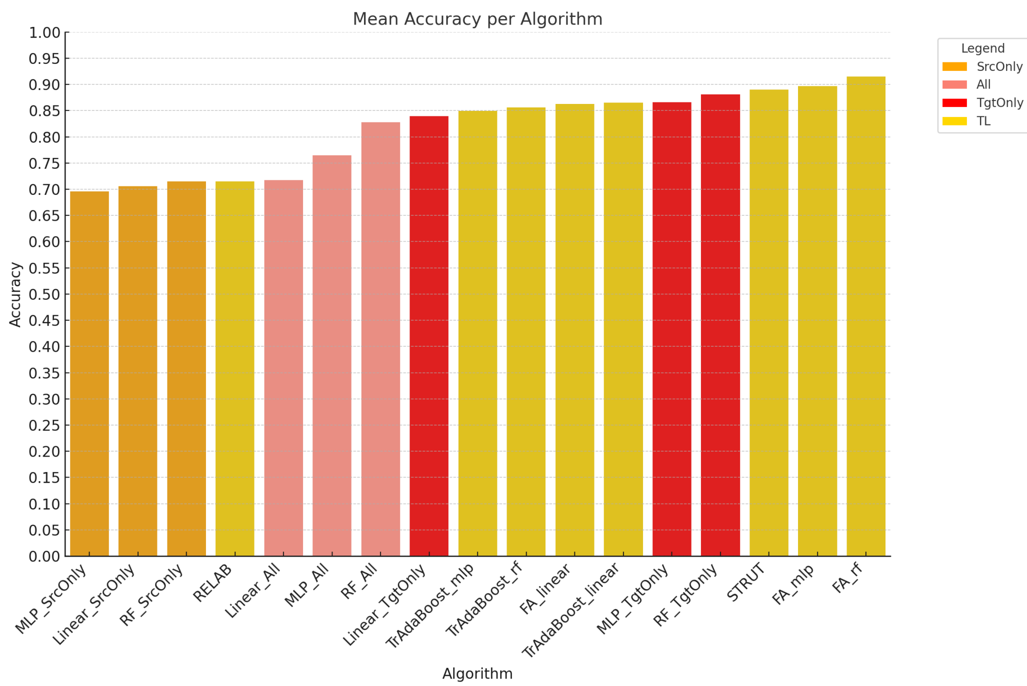


Figure 18: Mean accuracy on right hand data, inter-subject: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) Linear: A linear classifier model. 6) RF: Random Forest model with 50 trees. 7) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 8) Data normalization: All data are L2-normalized. 9) Gesture Data: Models use data from right hand performing 4 types of gestures.

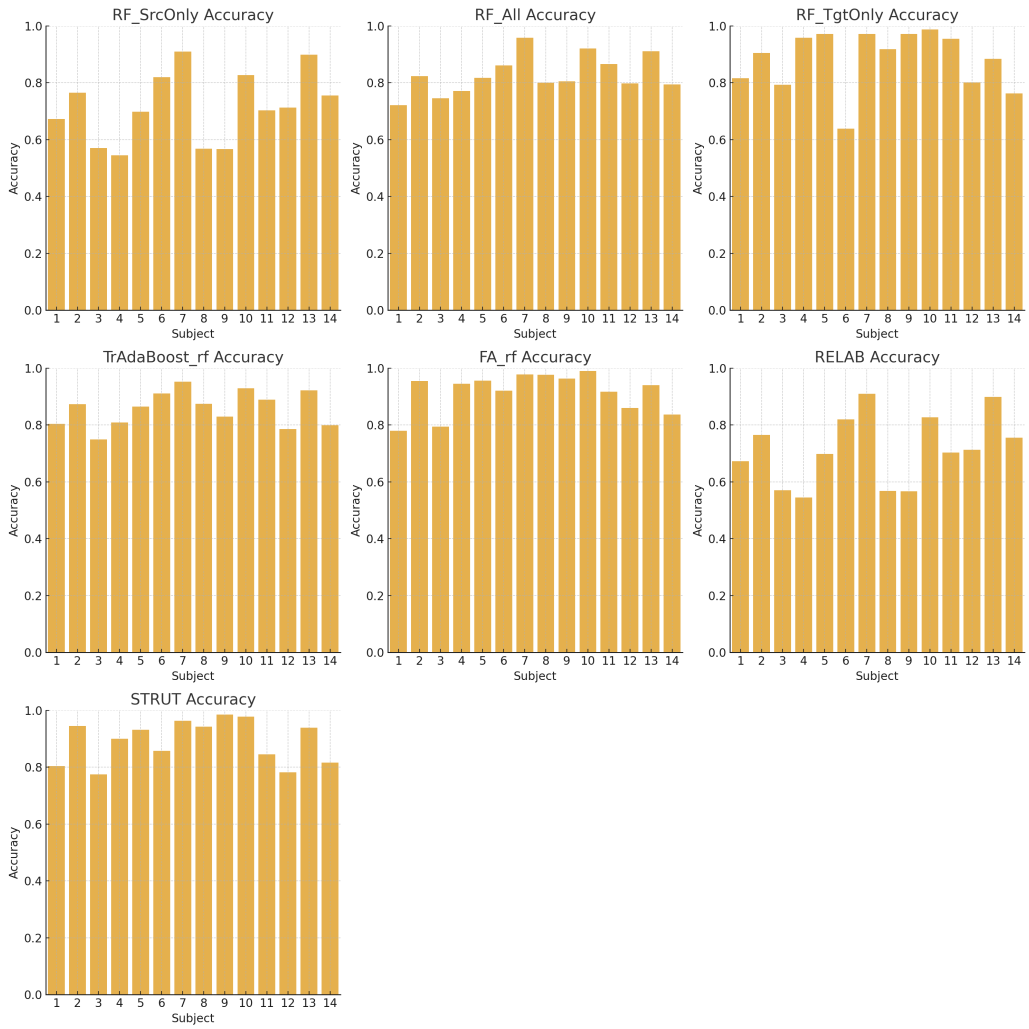


Figure 19: Per subject accuracy on right hand data, inter-subject with RF models: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) RF: Random Forest model with 50 trees. 6) Data normalization: All data are L2-normalized. 7) Gesture Data: Models use data from right hand performing 4 types of gestures.

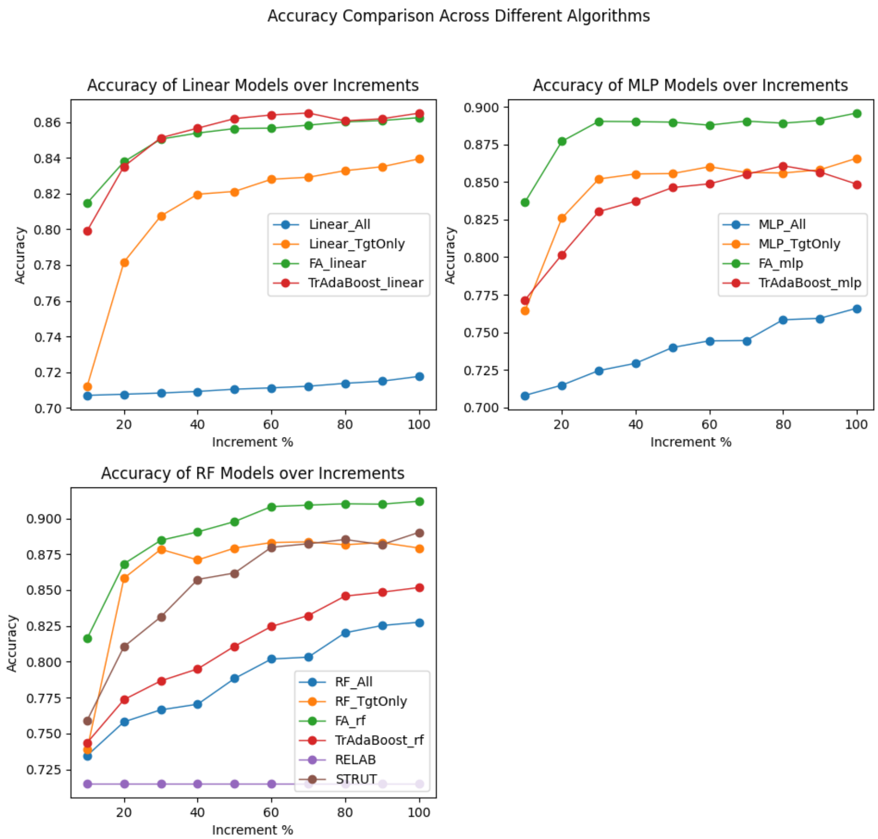


Figure 20: Right hand mean as a function of the percentage of target task data utilized, inter-subject: 1) TgtOnly: Uses only target task data. 2) All: Combines data from both source and target tasks. 3) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 4) Linear: A linear classifier model. 5) RF: Random Forest model with 50 trees. 6) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 7) Data normalization: All data are L2-normalized. 8) Gesture Data: Models use data from right hand performing 4 types of gestures.

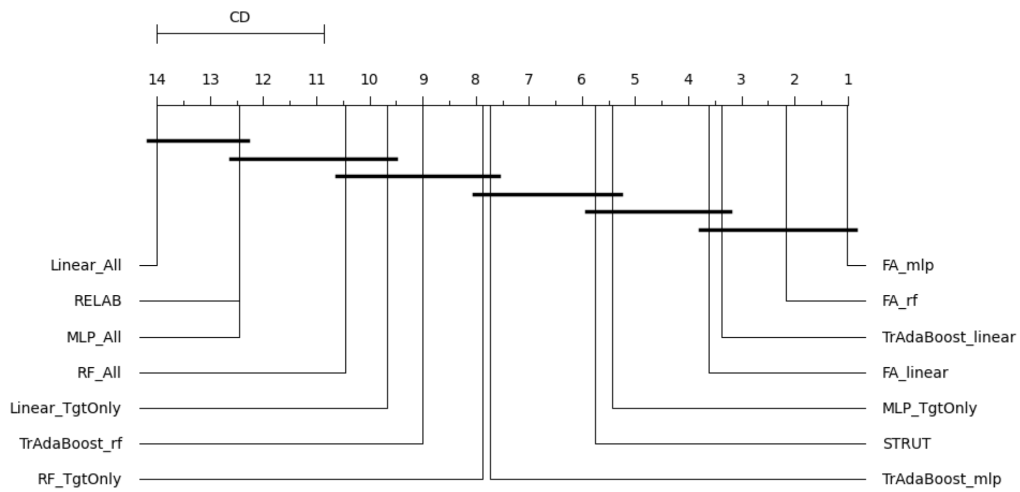


Figure 21: Average rank over 40 runs at 10 % of the Transfer Learning session data of the different models in the supervised single hand setting(the lower, the better). Models that are not significantly different are connected (at p = 0.05 found by a Nemenyi test, Section 17)

The source data is composed of right-hand data from all the subjects, excluding the target subject. The Transfer Learning data consists of right-hand data from the first session of the target subject, and the remaining sessions are used for testing.

As in the both-hands setting, FA models are ahead (Figure 18). The gap is even bigger when less data is used for the Transfer Learning data (Figure 20). Furthermore, TrAdaBoost with a linear model performs well compared to the target only Linear model when using less data (Figure 20). Source only and ”All” models underperform (Figure 18).

19.3 Supervised estimation between hands

From left hand to right hand within the same subject

The inter-hands intra-subject setting is interesting to investigate because, in a hypothetical scenario where a subject has already collected a significant amount of data for one hand, Transfer Learning could be applied to the second hand, potentially reducing the amount of data needed for training the second hand.

It uses the first session's right-hand data from the subject for Transfer Learning and testing (the remaining sessions), and left-hand data (all sessions) from the same subject as source data.

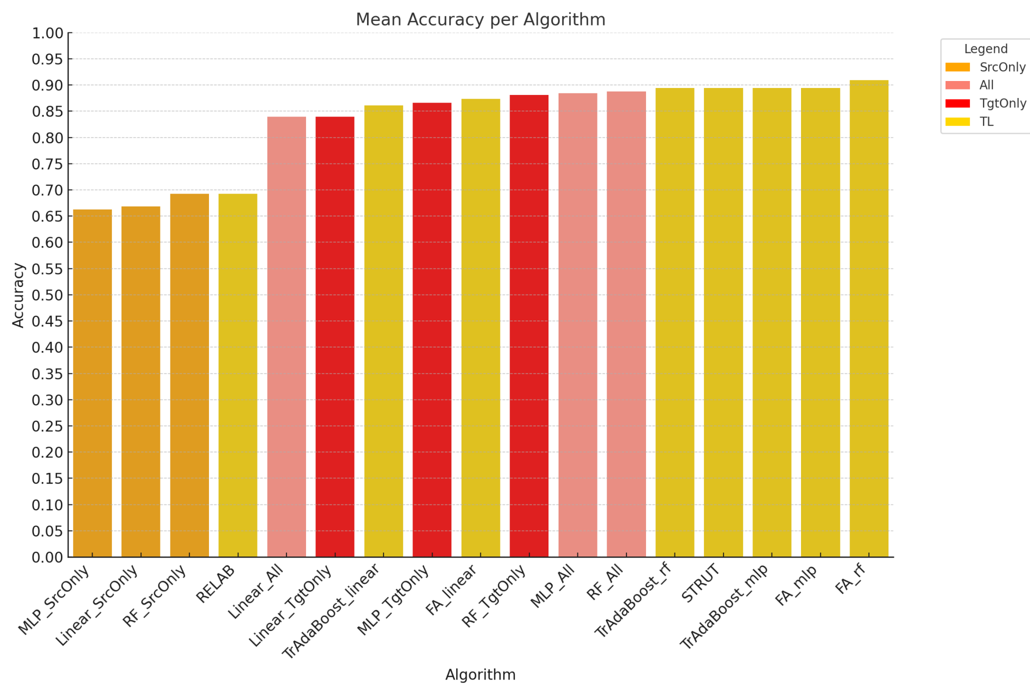


Figure 22: Mean accuracy between hands, intra-subject TL: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) Linear: A linear classifier model. 6) RF: Random Forest model with 50 trees. 7) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 8) Data normalization: All data are L2-normalized. 9) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

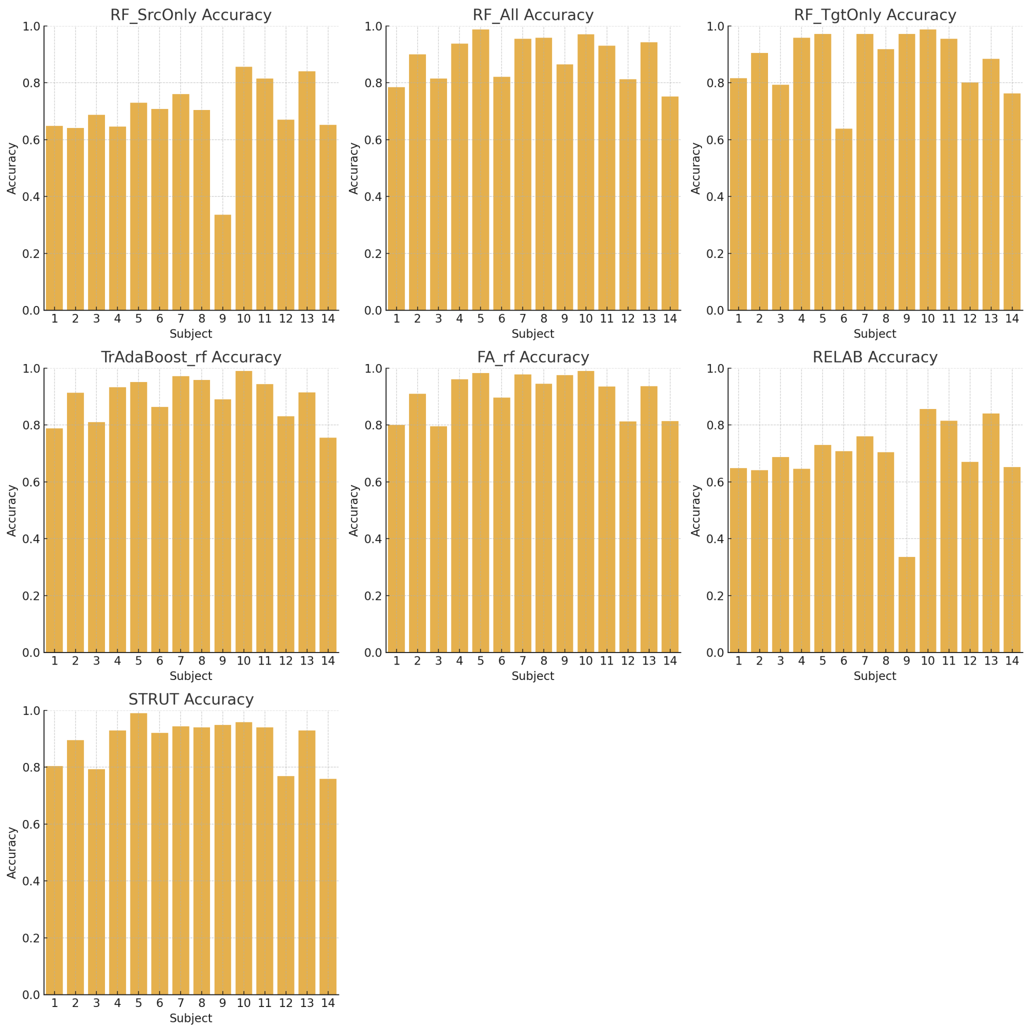


Figure 23: Per subject accuracy between hands, intra-subject TL with RF models: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) RF: Random Forest model with 50 trees. 6) Data normalization: All data are L2-normalized. 7) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

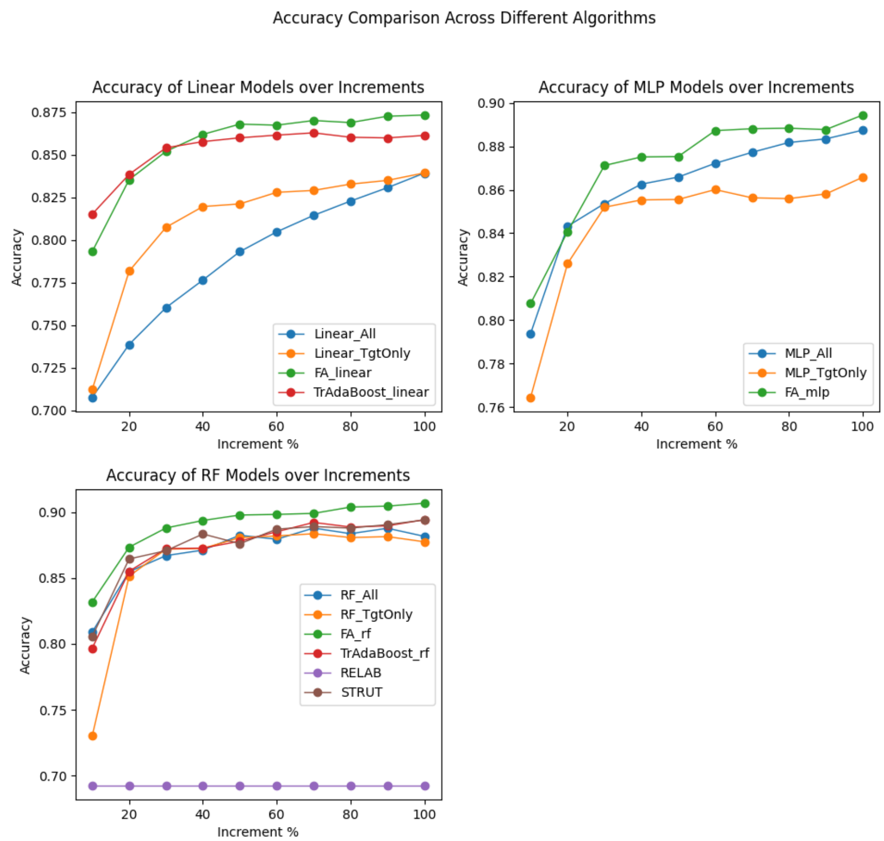


Figure 24: Mean accuracy between hands as a function of the percentage of target task data utilized, intra-subject TL: 1) TgtOnly: Uses only target task data. 2) All: Combines data from both source and target tasks. 3) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 4) Linear: A linear classifier model. 5) RF: Random Forest model with 50 trees. 6) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 7) Data normalization: All data are L2-normalized. 8) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

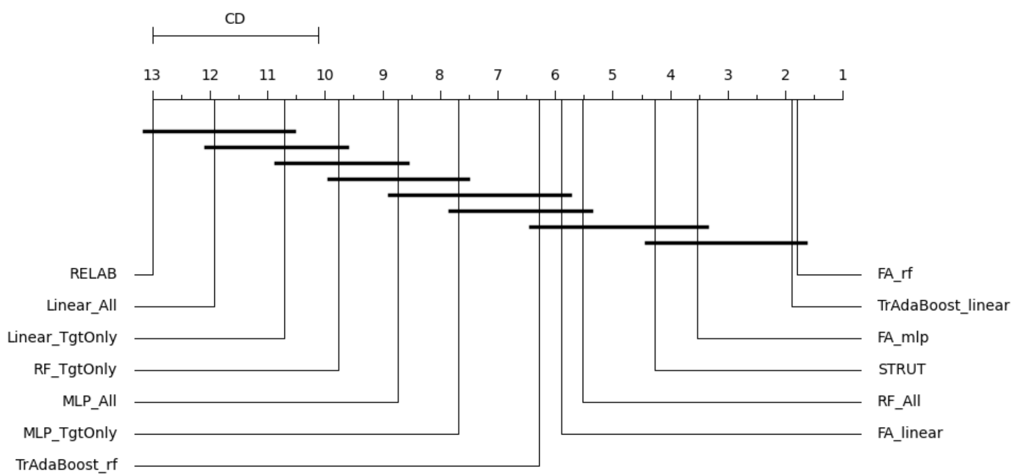


Figure 25: Average rank over 40 runs at 10 % of the Transfer Learning session data of the different models in the inter-hands intra-subject setting(the lower, the better). Models that are not significantly different are connected (at p = 0.05 found by a Nemenyi test, Section 17)

FA leads in this setting as well (Figures 22 and 25). As in the supervised single-hand setting, TrAdaBoost with a linear model performs well when less data is available (Figure 25). Contrary to the other settings, they are accompanied by STRUT and the Random Forest ”All” baseline.

It should be noted that TrAdaBoost with an MLP model could not be trained in this setting when using limited data, because errors were encountered.

From left hand to right hand inter-subject

In this setting, we investigate the performance achievable when only left hand data from other subjects is used with transfer learning to a target subject's right hand.

The source data comprises the left hand data from all subjects, excluding the target subject. The Transfer Learning data consists of the first session right hand data of the target subject, and the test data comprises the remaining sessions.

The results show that transfer learning can be achieved from the left hand data, as FA models and TrAdaBoost with a linear model achieve performance improvements over the target only models (Figures 26, 28).

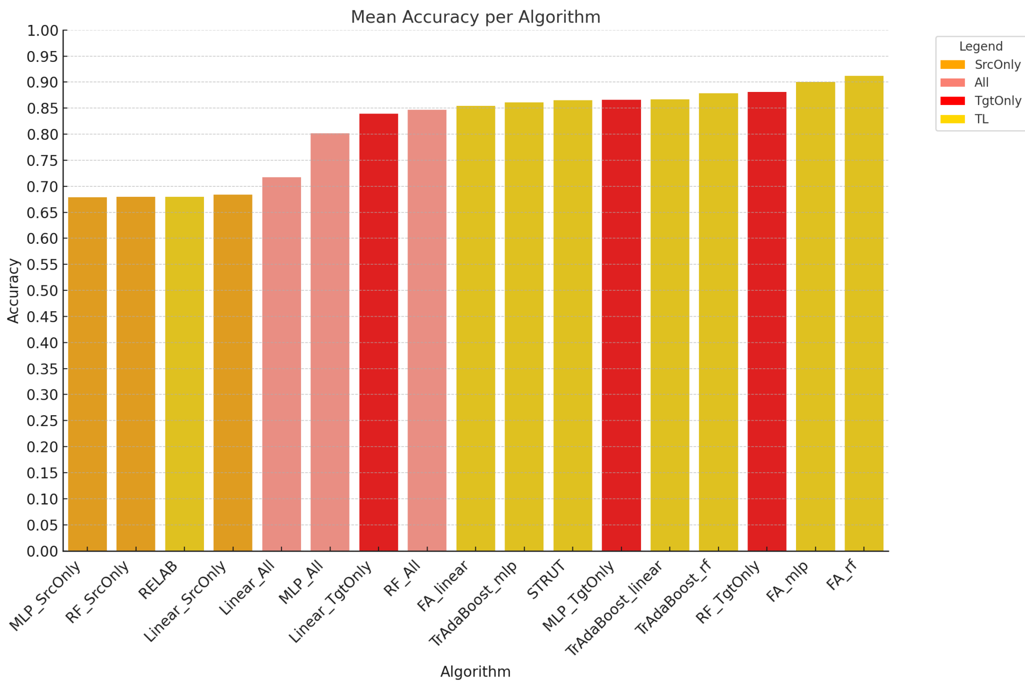


Figure 26: Mean accuracy between hands, inter-subject TL: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) Linear: A linear classifier model. 6) RF: Random Forest model with 50 trees. 7) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 8) Data normalization: All data are L2-normalized. 9) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

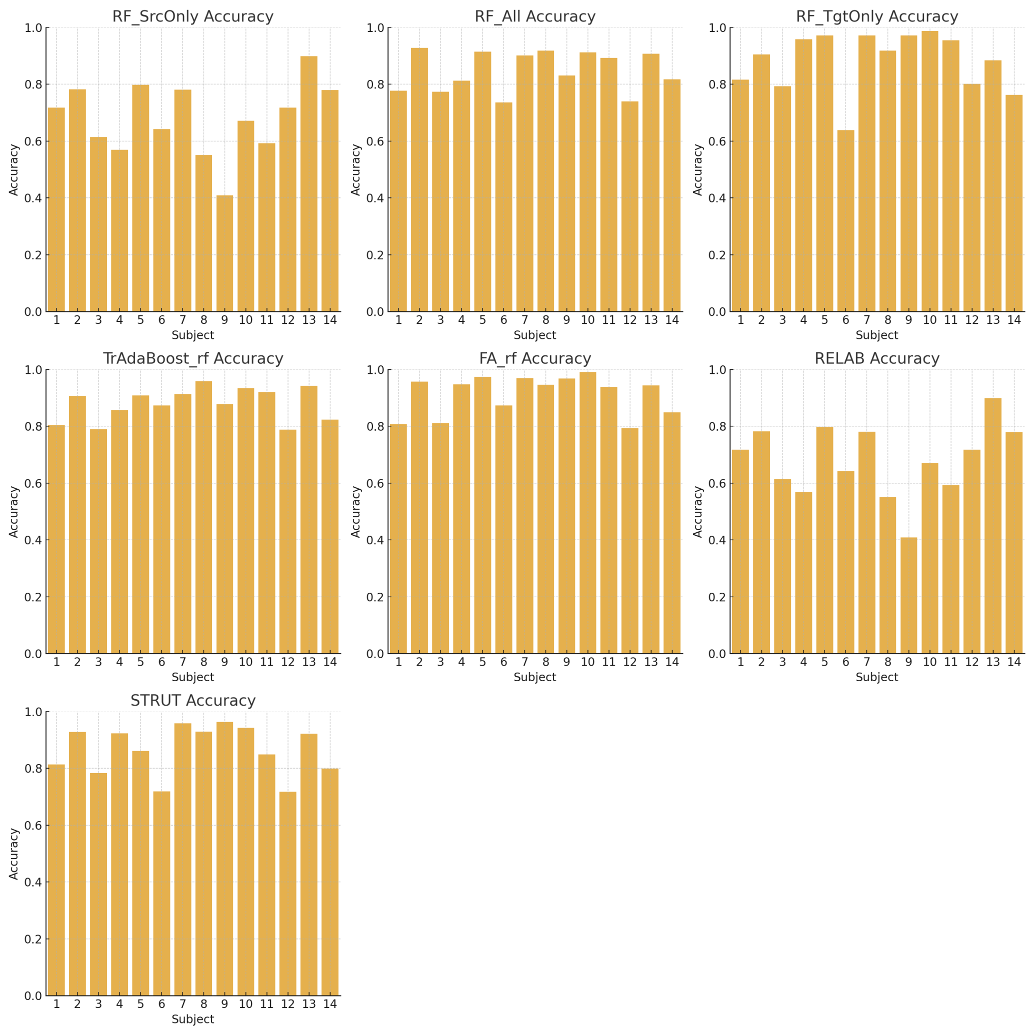


Figure 27: Per subject accuracy between hands, inter-subject TL with RF models: 1) SrcOnly: Uses only source task data. 2) TgtOnly: Uses only target task data. 3) All: Combines data from both source and target tasks. 4) TL: Utilizes data from both tasks with Transfer Learning methods, Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 5) RF: Random Forest model with 50 trees. 6) Data normalization: All data are L2-normalized. 7) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

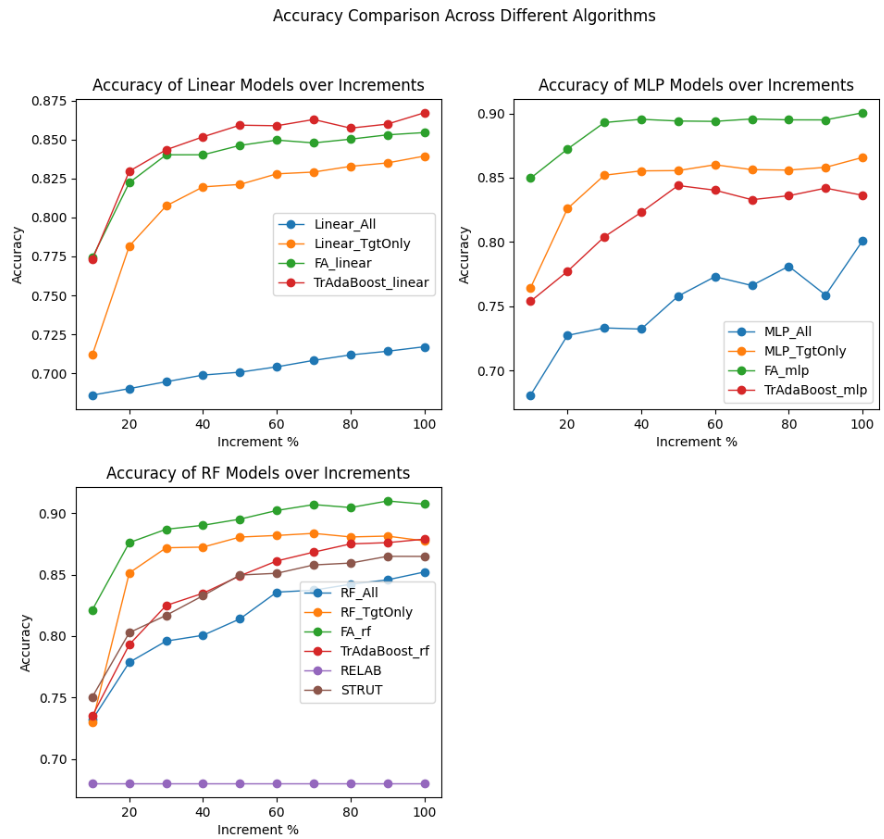


Figure 28: Mean accuracy between hands as a function of the percentage of target task data utilized, inter-subject TL: 1) TgtOnly: Uses only target task data. 2) All: Combines data from both source and target tasks. 3) TL: Utilizes data from both tasks with Transfer Learning methods, including Random Forest-based (RELAB, STRUT) and model-agnostic algorithms (TrAdaBoost, FA) applied to target task data. 4) Linear: A linear classifier model. 5) RF: Random Forest model with 50 trees. 6) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 7) Data normalization: All data are L2-normalized. 8) Gesture Data: Models use data from the left hand for the source task and data from the right hand for the target task, performing 4 types of gestures.

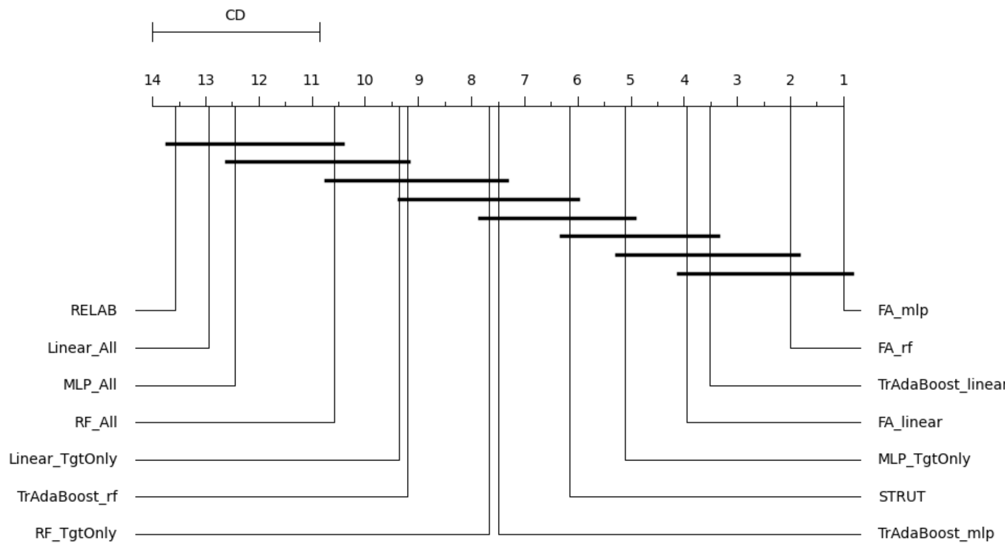


Figure 29: Average rank over 40 runs at 10 % of the Transfer Learning session data of the different models in the inter-hands inter-subject setting(the lower, the better). Models that are not significantly different are connected (at p = 0.05 found by a Nemenyi test, Section 17)

19.4 Unsupervised single hand estimation

In this last setting, we can attest the performance of Transfer Learning algorithms in the case where no labels can be collected from the target subject. It may be the case when it is difficult to collect labelled(e.g the target subject has no hand). Not using any labels for the source and target domains is called unsupervised transfer learning. Effectively, the Transfer Learning algorithms used in this setting do not use any labels. However, source labels are still used for the final classification models.

The source data is composed of the right hand data from all the subjects, except the target. The Transfer Learning data is composed of the unlabelled sessions 0-4 data of the target subject and the test data is composed of the labelled sessions 0-4 data of the target subject. Thus, more data is available for Transfer Learning than in the supervised single hand setting.

SA with a linear model performs the best, outperforming all other algorithms, while SA with an MLP model comes in second, showing good performance relative to the source-only MLP (Figures 30, 32). KLIEP does not show any improvement over the baselines, and CORAL performs worse than the baselines (Figures 30, 32). An interesting observation for all the unsupervised algorithms from the Figure 32 is that adding more data does not improve the performance, except for some minor fluctuations.

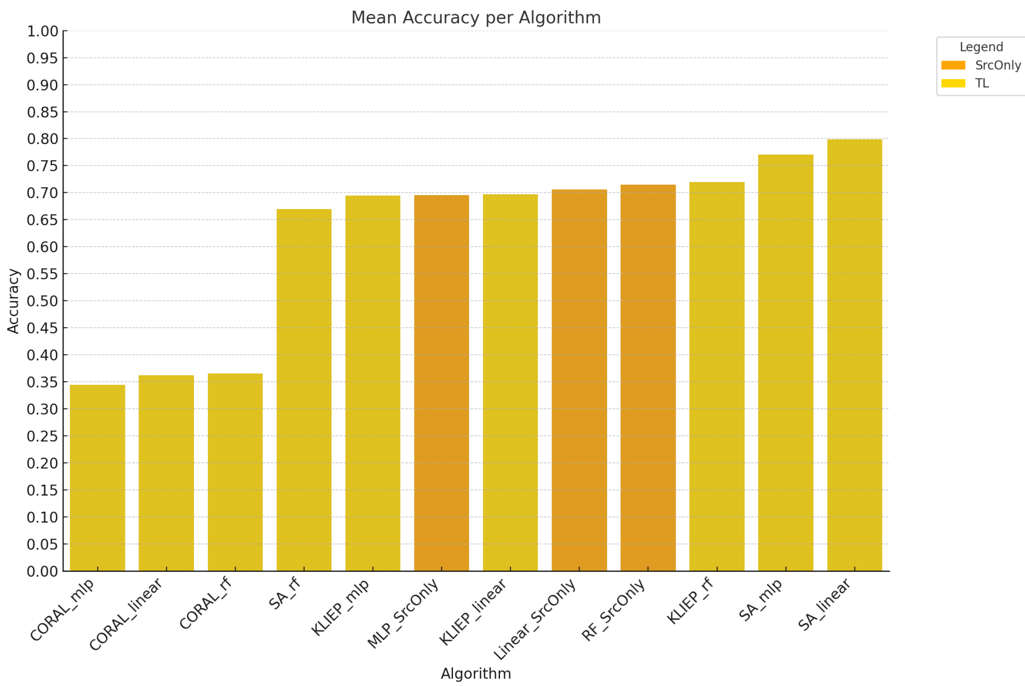


Figure 30: Mean accuracy on unlabelled right hand data, inter-subject TL: 1) SrcOnly: Uses only source task data. 2) Model-agnostic TL algorithms (CORAL, KLIEP and SA) with classification models are applied to target task data by laveraging source task data. 3) Linear: A linear classifier model. 4) RF: Random Forest model with 50 trees. 5) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 6) Data normalization: All data are L2-normalized. 7) Gesture Data: Models use data from the right hand, performing 4 types of gestures.

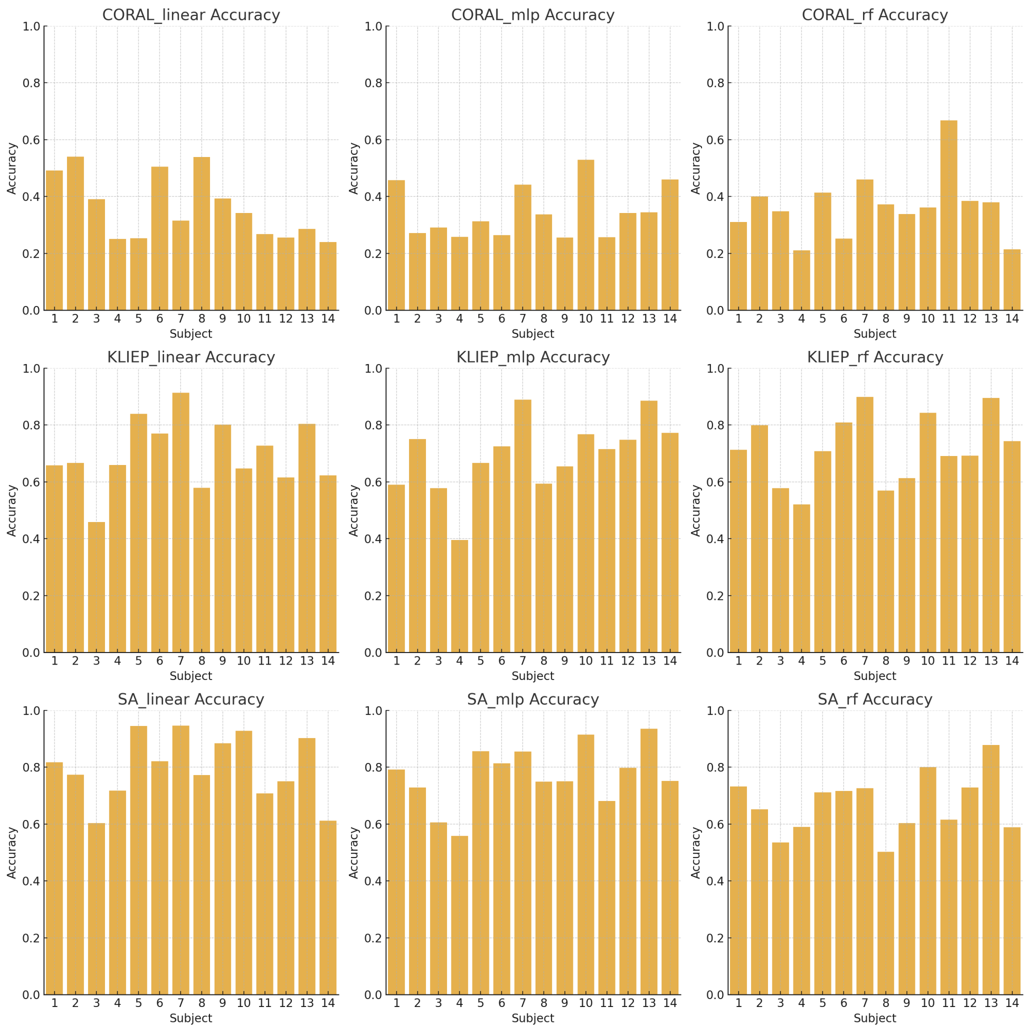


Figure 31: Per subject accuracy on unlabelled right hand data, inter-subject TL with RF models: 1) Model-agnostic TL algorithms (CORAL, KLIEP and SA) with classification models are applied to target task data by laveraging source task data. 2) Linear: A linear classifier model. 3) RF: Random Forest model with 50 trees. 4) Data normalization: All data are L2-normalized. 5) Gesture Data: Models use data from the right hand, performing 4 types of gestures.

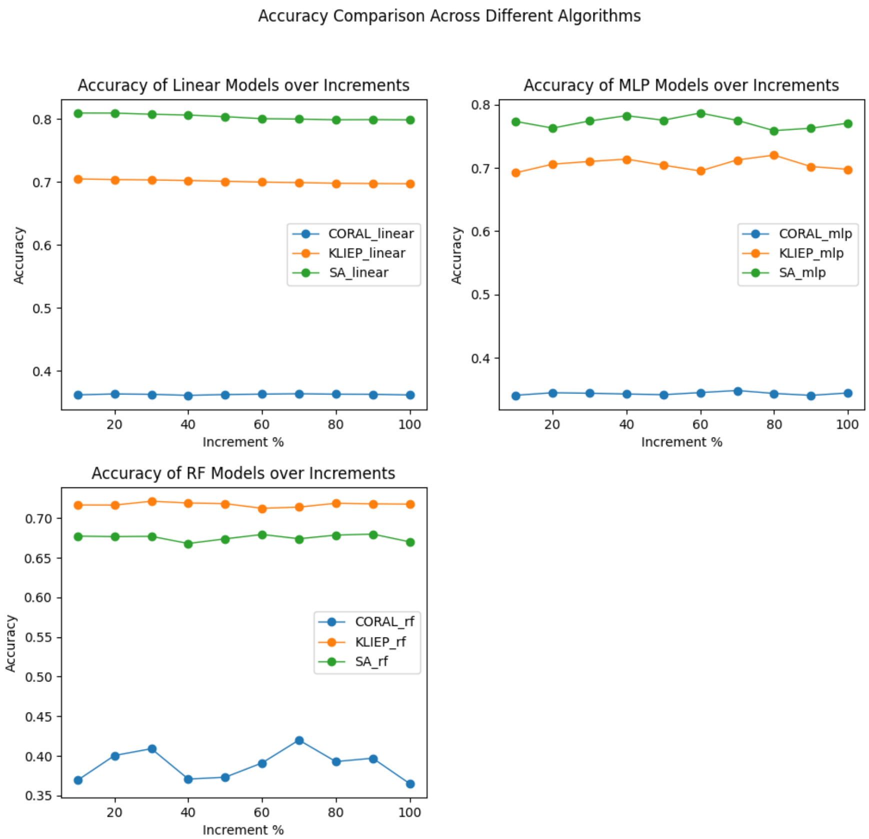


Figure 32: Mean accuracy on unlabelled right hand as a function of the percentage of target task data utilized, inter-subject TL: 1) Model-agnostic TL algorithms (CORAL, KLIEP and SA) with classification models are applied to target task data by laveraging source task data. 2) Linear: A linear classifier model. 3) RF: Random Forest model with 50 trees. 4) MLP: Multi-Layer Perceptron with a 10-neuron hidden layer, trained over 2000 epochs. 5) Data normalization: All data are L2-normalized. 6) Gesture Data: Models use data from the right hand, performing 4 types of gestures.

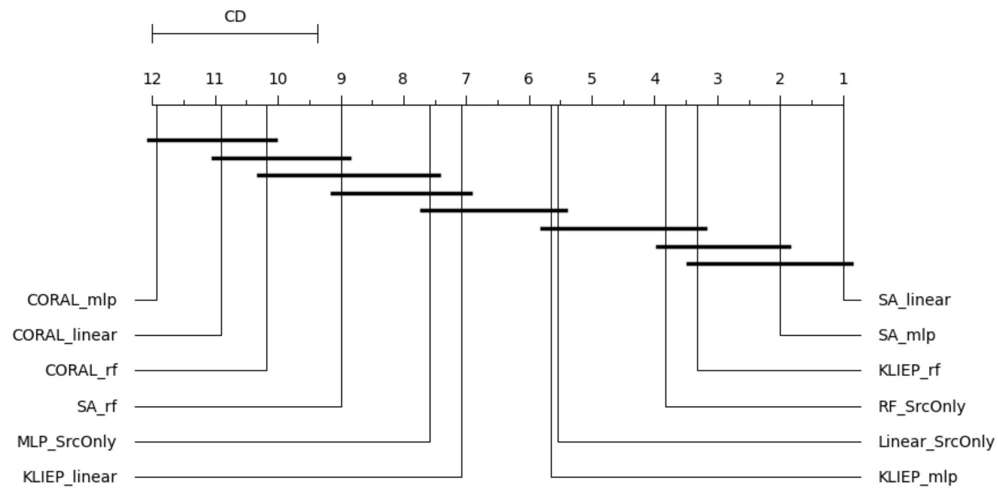


Figure 33: Average rank over 40 runs at 10 % of the Transfer Learning session data of the different models in the unsupervised setting(the lower, the better). Models that are not significantly different are connected (at p = 0.05 found by a Nemenyi test, Section 17)

19.5 Divergence measurements

In the intra-subject setting, we determined the PAD distance between sessions of the same subject. Specifically, we computed the PAD values between session 1 and the subsequent four sessions. These values are presented in Table 2. It should be noted that the value for the 5th session of the 9th subject is missing, as there is no 5th session for this subject in the dataset.
The inter-subject setting involves calculating the PAD values between the domains of different subjects, with each domain comprised of combined sessions. The PAD values for this setting are displayed in Table 19.5.
The results highlight a significant divergence between the domains of individual subjects. While this divergence is less pronounced between sessions.

Table 5: Divergence percentages (represented as %) between session 0 and the rest for each subject. The data are l2-normalised. Right hand.

Subject	Session 1	Session 2	Session 3	Session 4
1	66	77	76	70
2	90	93	89	–
3	76	57	63	52
4	69	69	66	79
5	69	96	81	96
6	90	90	95	90
7	22	81	61	53
8	87	48	73	86
9	42	57	57	68
10	24	32	59	52
11	60	54	75	89
12	76	55	52	55
13	62	57	78	75
14	81	43	68	64

Subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	97	100	100	99	100	98	98	99	100	100	97	98	96
2	100	0	98	98	99	100	96	97	97	97	99	97	91	98
3	99	98	0	99	99	99	98	98	99	100	99	95	100	99
4	98	96	98	0	99	98	96	99	99	97	99	97	94	98
5	99	98	99	97	0	99	96	100	97	99	99	99	99	98
6	99	100	99	99	100	0	97	96	99	99	99	98	99	99
7	98	93	99	94	100	99	0	96	98	97	98	93	98	97
8	100	96	97	99	99	97	94	0	98	98	98	98	97	99
9	98	95	97	99	97	98	96	95	0	99	99	94	98	100
10	99	95	100	98	99	99	100	99	97	0	99	98	96	99
11	99	96	99	99	100	96	99	98	98	99	0	99	100	99
12	98	97	98	100	100	100	100	98	98	99	100	0	96	98
13	98	90	99	95	98	98	97	97	99	99	98	98	0	96
14	94	96	99	96	100	100	99	99	100	99	99	95	94	0

Table 6: Divergence percentages between the subjects. The data are l2-normalised. Right hand.

20 Discussion

Both hands

In Figures 16 and 17, we can observe that, adding data from the target subject enhances the model's performance. This improvement is initially evident in the 'All' baseline when compared to the source only. Moreover, the target only baseline outperforms both the source only and 'All' baselines. The Transfer Learning algorithms either match or surpass the performance of the target only approach. Notably, Feature Augmentation (FA), achieving an average accuracy of 90 % across subjects.
Certain transfer learning models show better performance with less labeled data. In order to examine the threshold at which transfer learning outper-

forms a simple classification with the transfer session included, we apply an accuracy test on varying fractions of a subject's session 0 data, starting from 10% and incrementing by 10% up to the full session data. However, we've only used this approach for 10 % of Transfer Learning data in this setting because we were limited by the hardware as using both hands data takes much more time to train. At 10 % of data the results are more pronounced with bigger gaps between the baselines and Transfer Learning algorithms(Table 4).

Single hand

The Figures 18 and 19, show a similar result as in the both hands. FA is again the most performant algorithm, achieving a good average accuracy across subjects.
The incremental approach allows us to discern that FA consistently outperforms(Figure 20) the "All" and target only baselines. The performance difference is the biggest for the initial 10-20% of the data but becomes smaller thereafter.

Between hands

The inter-hands intra-subject application shows similar but slightly different results. As observed in Figures 22 and 23, the 'All' baseline performs higher, even surpassing the target only for some models. Transfer Learning algorithms still lead, with FA using Random Forest and MLP as frontrunners. The incremental approach in Figure 24 confirms these findings, although the difference between the Transfer Learning algorithms becomes smaller.
The inter-hands inter-subject performance is more closely aligned with the single-hand and both-hands experiments. In this scenario, target only models outperform the 'All' baseline (as seen in Figures 26 and 27) when using the entire session, with FA leading in most cases (Figure 28).
Overall, the results from this application are encouraging, as they suggest that having data from only one hand for the source task is sufficient to train models for both hands on the target subject.

Unsupervised TL

Colot et al. concluded[9] that using SA improved inter-subject classification accuracy in an Unsupervised Transfer Learning setting. This work confirms those results. In this experiment, no labelled data is available for the target task (excluding test data), which causes the algorithms to produce inferior results compared to supervised algorithms. The unavailability of target data makes it impossible to compare with baselines such as 'All' and target only. However, a comparison with source only is still feasible.

Method	SrcOnly	KLIEP	SA
Linear	0.70	0.69	**0.79**
MLP	0.69	0.69	**0.77**
RF	0.71	**0.71**	0.66

Table 7: Comparison of average accuracies between Source Only and Unsupervised TL algorithms, with bold measurements representing the most performant algorithms.

In Figures 30, 31, and 32, it is observed that CORAL underperforms compared to KLIEP and SA. When compared to source only (as shown in Table 7), SA significantly outperforms with Linear and MLP models with performance similar to FA in the single hand supervised setting with few data, but not with Random Forest. Meanwhile, KLIEP maintains the same performance level as source only.
Additionally, there is little to no difference observed when using more data in the incremental approach (Figure 32). This suggests that these algorithms are unsuitable in hypothetical scenarios where new end user data, past some threshold, could be collected and added to the models post training.

RELAB anomaly

Across all related results, RELAB performed on par with a source only Random Forest (Figures 16, 17, 18, 19, 22, 23, 26, 27). This suggests a significant occurrence of negative transfer learning(Table 8). Seemingly, using less data(Figures 20, 24, 28) had no impact on the algorithm's performance, reinforcing the source only behavior.

STRUT related results

In related work from 2021, Marano et al. [31] applied a combination of Random Forests and transfer learning to EMG data using a public dataset

Table 8: Negative Transfer Gap (NTG) Values for RELAB (error rates of RELAB and target only Random Forest)

Both Hands	Single Hand	Inter-Hands Intra-Subject	Inter-Hands Inter-Subject
0.15	0.17	0.19	0.20

and TransferForestClassifier.
The method employed by Marano et al. shares significant similarities with our approach and can be summarized as follows:

1. A leave-one-out strategy is adopted for data division, where a single subject's data is set aside for transfer learning and testing, while the rest of the subjects form the source data.

2. The source data is used for the initial pretraining of the model.

3. The pretrained model is then refined using the target subject's data.

4. Finally, the accuracy of the refined model is evaluated on the test data.

They utilized data from the NinaPro database[3].
In terms of transfer learning settings, they conducted experiments using both inter-subject and intra-subject settings. However, they adopted a different configuration of sessions for these experiments.
Regarding the results, they align with our findings. The inter-subject single hand experiments indicate that utilizing prior experience from other subjects does not lead to significant improvement compared to the scenario without transfer learning on the target only (Figure 20).
In the inter-hands intra-subject setting, a substantial improvement over target only is observed(Figure 24). However, the "All" baseline performs at the same level.

Part V

Conclusion

21 Summary of Transfer Learning results

This study has explored some aspects of Transfer Learning on EMG data, revealing insights into data visualization, performance evaluation, and divergence measurements. The following sections summarize the key findings:

1. **Data visualization**: PCA and UMAP were utilized for data visualization. While PCA captured linear structures, UMAP revealed more complex nonlinear structures. visualization showed that data from individual subjects could be separated, but the separation became more challenging when data from all subjects were combined.

2. **Transfer Learning Performance**: We explored various transfer learning models to address inter-subject variability, including RELAB, STRUT, TrAdaBoost, FA, Coral, KLIEP, and SA. The comparison revealed different advantages of Transfer Learning.

    - Generally, the Transfer Learning algorithms surpass the performance of the most effective baseline (target only) by a small margin. This gap significantly widens in favor of Transfer Learning when the amount of target data is very limited. From those algorithms, FA is the most performing one.
    - The results showed good performance when using either the right or left hand separately, compared to using data from both hands. Thus, using half the amount of data, as opposed to the both-hands setting, is a substantial reduction in data needed at the collection stage.
    - Furthermore, the inter-hands inter-subject setting further eliminates the need to collect data for both hands, as the performance is on par with the supervised single hand setting when applying Transfer Learning from one hand to another.
    - Finally, when no target labels are available, we confirmed that SA greatly improves performance over a source only model by using only a small amount of data. Adding more data does not result in increased performance.

3. **Divergence measurements**: The divergence in PAD values between different domains and sessions of subjects was found to be significant. As

expected, there is much less divergence between sessions of the same subject, while there is considerable divergence between data from different subjects.

22 Future research

These are some possible pathways for future research:

- UMAP and SA can compute new spaces for the source and/or target data. Divergence metrics could be employed to determine whether the new spaces are closer to each other than the original spaces.

- The visualization techniques revealed that the data are difficult to separate when all the subjects are grouped together (inter-subject setting). Techniques such as Metric Learning could be employed to transform the feature space into a more separable one[5]. UMAP can utilize the labels for better class separation. Different versions of SA, such as ITML-PCA and LMSA[18], also attempt to use labels and metric learning for more effective subspace generation.

- Future research could focus on popular Transfer Learning algorithms that were not touched in this thesis employing deep learning and fine tuning.

- A simplified version of the problem could be explored, such as reducing it to only two gestures or applying Transfer Learning between two sessions, whether involving the same or different subjects.

- We have already employed a multi-source TL method (FA), but more algorithms of this kind can be utilized. Multi-source Transfer Learning involves using multiple different but related sources, instead of pooling all the source subjects into a single source.

- An experiment can be carried out to test whether adding or removing subjects from the source domain affects Transfer Learning performance. This should allow for testing the scalability of Transfer Learning methods.

Bibliography

[1] Pulkit Agrawal, Ross B. Girshick, and Jitendra Malik. "Analyzing the Performance of Multilayer Neural Networks for Object Recognition". In: *European Conference on Computer Vision*. 2014. URL: https://api.semanticscholar.org/CorpusID:1811598.

[2] Ali Ameri et al. "Support Vector Regression for Improved Real-Time, Simultaneous Myoelectric Control". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22 (May 2014). DOI: 10.1109/TNSRE.2014.2323576.

[3] Manfredo Atzori et al. "Electromyography data for non-invasive naturally-controlled robotic hand prostheses". In: *Scientific Data* 1 (2014). URL: https://api.semanticscholar.org/CorpusID:3488676.

[4] Hal Daumé III au2. *Frustratingly Easy Domain Adaptation*. 2009. arXiv: 0907.1815 [cs.LG].

[5] Aurélien Bellet, Amaury Habrard, and Marc Sebban. "A Survey on Metric Learning for Feature Vectors and Structured Data". In: *ArXiv* abs/1306.6709 (2013). URL: https://api.semanticscholar.org/CorpusID:168956.

[6] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine Learning* 79 (2010), pp. 151–175. URL: https://api.semanticscholar.org/CorpusID:8577357.

[7] Gianluca Bontempi. *"Statistical foundations of machine learning: the handbook"*. Feb. 2021.

[8] Alexandre Calado, Filomena Soares, and Demétrio Matos. "A Review on Commercially Available Anthropomorphic Myoelectric Prosthetic Hands, Pattern-Recognition-Based Microcontrollers and sEMG Sensors used for Prosthetic Control". In: Apr. 2019. DOI: 10.1109/ICARSC.2019.8733629.

[9] Martin Colot et al. "EMG Subspace Alignment and Visualization for Cross-Subject Hand Gesture Classification". In: *Proceedings of the ECML-PKDD 2023 Workshop on Adapting to Change: Reliable Learning Across Domains*. 2023.

[10] Martin Colot et al. "Hand gestures estimation from emg and vr: A machine learning approach". MA thesis. ULB, 2022. URL: https://mlg.ulb.ac.be/wordpress/members-2/martincolot.

[11] Cometa. *PicoEMG*. https://www.cometasystems.com/products/picoemg. Accessed: 2023. 2023.

[12] Matt Cragun. *NVIDIA DRIVE Sim Powered by Omniverse Available for Early Access This Summer*. 2021. URL: https://blogs.nvidia.com/blog/2021/04/12/nvidia-drive-sim-omniverse-early-access/ (visited on 05/06/2022).

[13] Wenyuan Dai et al. "Boosting for transfer learning". In: *ICML '07*. 2007.

[14] Janez Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". In: *J. Mach. Learn. Res.* 7 (2006), pp. 1–30. URL: https://api.semanticscholar.org/CorpusID:7553535.

[15] James Doran. "Book Review. Experiments in Induction Earl B. Hunt, Janet Marin, and Phillip J. Stone , 1966 ; 247. (New York and London : Academic Press , 76s.)" In: *The Computer Journal* 10 (Mar. 1967), pp. 299–299. DOI: 10.1093/comjnl/10.3.299.

[16] Kevin Englehart and Bernard Hudgins. "A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control". In: *IEEE transactions on bio-medical engineering* 50 (Aug. 2003), pp. 848–54. DOI: 10.1109/TBME.2003.813539.

[17] Jose Fajardo, Orlando Gomez, and Flavio Prieto. "EMG hand gesture classification using handcrafted and deep features". In: *Biomedical Signal Processing and Control* 63 (Sept. 2020). DOI: 10.1016/j.bspc.2020.102210.

[18] Basura Fernando et al. “Subspace Alignment For Domain Adaptation”. In: *ArXiv* abs/1409.5241 (2014).

[19] Yoav Freund and Robert E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *European Conference on Computational Learning Theory*. 1997.

[20] Tatiana Gaintseva. *Transfer Learning*. Lecture at Deep Learning School, Moscow Institute of Physics and Technology. Available at: `https://youtu.be/oLpREso27Zw`. 2020.

[21] Gemini Team Google. *Gemini: A Family of Highly Capable Multimodal Models*. Technical Report. Google, 2023.

[22] Branav Kumar Gnanamoorthy Branav Kumar Gnanamoorthy. *Machine Learning*. 2018. URL: `https://medium.com/@gnabr/machine-learning-c28daf3cf60a` (visited on 05/06/2022).

[23] Steffen Herbold. “Autorank: A Python package for automated ranking of classifiers”. In: *Journal of Open Source Software* 5.48 (2020), p. 2173. DOI: `10.21105/joss.02173`. URL: `https://doi.org/10.21105/joss.02173`.

[24] Asmaul Hosna et al. “Transfer learning: a friendly introduction”. In: *Journal of Big Data* 9 (Oct. 2022). DOI: `10.1186/s40537-022-00652-w`.

[25] Julius Kunze et al. “Transfer Learning for Speech Recognition on a Budget”. In: *Rep4NLP@ACL*. 2017. URL: `https://api.semanticscholar.org/CorpusID:2410413`.

[26] Jian Liang, D. Hu, and Jiashi Feng. “Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation”. In: *International Conference on Machine Learning*. 2020. URL: `https://api.semanticscholar.org/CorpusID:211205159`.

[27] Gregory A. Light et al. “Electroencephalography (EEG) and Event-Related Potentials (ERPs) with Human Participants”. In: *Current Protocols in Neuroscience* 52.1 (2010), pp. 6.25.1–6.25.24. DOI: `https://doi.org/10.1002/0471142301.ns0625s52`. eprint: `https://currentprotocols.onlinelibrary.wiley.com/doi/pdf/10.1002/0471142301.ns0625s52`. URL: `https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/0471142301.ns0625s52`.

[28] J. Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151. DOI: 10.1109/18.61115.

[29] Y. P. Lin and T. P. Jung. "Improving EEG-Based Emotion Classification Using Conditional Transfer Learning". In: *Front Hum Neurosci* 11 (2017). [PubMed Central:PMC5486154] [DOI:10.3389/fnhum.2017.00334] [PubMed:21994498], p. 334.

[30] Ana María Maitín, Juan Pablo Romero Muñoz, and Álvaro José García-Tejedor. "Survey of Machine Learning Techniques in the Analysis of EEG Signals for Parkinson's Disease: A Systematic Review". In: *Applied Sciences* (2022). URL: https://api.semanticscholar.org/CorpusID:250466277.

[31] Giulio Marano et al. "Questioning Domain Adaptation in Myoelectric Hand Prostheses Control: An Inter- and Intra-Subject Study". In: *Sensors (Basel, Switzerland)* 21 (2021).

[32] Antoine de Mathelin et al. "ADAPT: Awesome Domain Adaptation Python Toolbox". In: *arXiv preprint arXiv:2107.03049* (2021).

[33] Leland McInnes and John Healy. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *ArXiv* abs/1802.03426 (2018). URL: https://api.semanticscholar.org/CorpusID:3641284.

[34] Jimson Ngeo, Tomoya Tamei, and Tomohiro Shibata. "Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model". In: *Journal of neuroengineering and rehabilitation* 11 (Aug. 2014), p. 122. DOI: 10.1186/1743-0003-11-122.

[35] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[36] Aleksander Pałkowski and Grzegorz Redlarski. "Basic Hand Gestures Classification Based on Surface Electromyography". In: *Computational and Mathematical Methods in Medicine* 2016 (Jan. 2016), pp. 1–7. DOI: 10.1155/2016/6481282.

[37] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[38] Aayush Prakash et al. "Self-Supervised Real-to-Sim Scene Generation". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2020), pp. 16024–16034. URL: https://api.semanticscholar.org/CorpusID:237213759.

[39] Reid Pryzant. *proxy-a-distance*. GitHub repository. 2023. URL: https://github.com/rpryzant/proxy-a-distance.

[40] Dandu Sitarama Raju and D LakshmayyaNaidu. "Electromyography-principles and applications: A review-I". In: *annals and essences of dentistry* 4 (2012), pp. 49–54.

[41] *Real Self-Driving Cars are on the way thanks to Nvidia - gallery*. URL: https://www.topspeed.com/cars/car-news/real-self-driving-cars-are-on-the-way-thanks-to-nvidia-ar177903/pictures.html (visited on 05/06/2023).

[42] Real Statistics Using Excel. *Studentized Range q Table*. Accessed: 2023. 2023. URL: https://real-statistics.com/statistics-tables/studentized-range-q-table/.

[43] Noam Segev et al. "Learn on Source, Refine on Target: A Model Transfer Learning Framework with Random Forests". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.9 (2017), pp. 1811–1824. DOI: 10.1109/TPAMI.2016.2618118.

[44] Lauren H. Smith and Levi J. Hargrove. "Comparison of surface and intramuscular EMG pattern recognition for simultaneous wrist/hand motion classification". In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2013), pp. 4223–4226.

[45] Masashi Sugiyama et al. "Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation". In: *Neural Information Processing Systems*. 2007. URL: https://api.semanticscholar.org/CorpusID:9133542.

[46] Baochen Sun, Jiashi Feng, and Kate Saenko. *Correlation Alignment for Unsupervised Domain Adaptation*. 2016. arXiv: 1612.01939 [cs.CV].

[47] Dong Wang and Thomas Fang Zheng. "Transfer learning for speech and language processing". In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (2015), pp. 1225–1237. URL: `https://api.semanticscholar.org/CorpusID:15301376`.

[48] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3 (May 2016). DOI: `10.1186/s40537-016-0043-6`.

[49] Eric W. Weisstein. *L^2-Norm.* `https://mathworld.wolfram.com/L2-Norm.html`.

[50] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *NIPS*. 2014. URL: `https://api.semanticscholar.org/CorpusID:362467`.

[51] Zhipeng Yu et al. "Surface EMG-Based Instantaneous Hand Gesture Recognition Using Convolutional Neural Network with the Transfer Learning Method". In: *Sensors* 21 (Apr. 2021), p. 2540. DOI: `10.3390/s21072540`.

[52] Jing Zhang, W. Li, and Philip Ogunbona. "Joint Geometrical and Statistical Alignment for Visual Domain Adaptation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5150–5158. URL: `https://api.semanticscholar.org/CorpusID:23629495`.

[53] Lei Zhang. "Transfer Adaptation Learning: A Decade Survey". In: *IEEE transactions on neural networks and learning systems* PP (2019). URL: `https://api.semanticscholar.org/CorpusID:75137541`.

[54] Wen Zhang et al. "A Survey on Negative Transfer". In: *IEEE/CAA Journal of Automatica Sinica* 10 (2020), pp. 305–329. URL: `https://api.semanticscholar.org/CorpusID:235790783`.

[55] Yingwei Zhang et al. "Dual layer transfer learning for sEMG-based user-independent gesture recognition". In: *Personal and Ubiquitous Computing* (Apr. 2020). DOI: `10.1007/s00779-020-01397-0`.