Journal of Neural Engineering

TOPICAL REVIEW • OPEN ACCESS

Deep learning for biosignal control: insights from basic to real-time methods with recommendations

To cite this article: Arnau Dillen *et al* 2022 *J. Neural Eng.* **19** 011003

Journal of Neural Engineering

CrossMark

OPEN ACCESS

RECEIVED
31 October 2021

REVISED
6 January 2022

ACCEPTED FOR PUBLICATION
27 January 2022

PUBLISHED
28 February 2022

TOPICAL REVIEW

Deep learning for biosignal control: insights from basic to real-time methods with recommendations

Arnau Dillen[1,2,5], Denis Steckelmacher[2], Kyriakos Efthymiadis[2], Kevin Langlois[3], Albert De Beir[3], Uros Marusic[4], Bram Vanderborght[3], Ann Nowé[2], Romain Meeusen[1], Fakhreddine Ghaffari[5], Olivier Romain[5] and Kevin De Pauw[1,*]

[1] Human Physiology and Sports Physiotherapy Research Group, Vrije Universiteit Brussel, Brussels, Belgium
[2] Artificial Intelligence Research Group, Vrije Universiteit Brussel, Brussels, Belgium
[3] Robotics and MultiBody Mechanics Research Group, Vrije Universiteit Brussel and IMEC, Brussels, Belgium
[4] Department of Health Sciences, Alma Mater Europaea—ECM, Maribor, Slovenia
[5] Equipes Traitement de l'Information et Systèmes, CY Cergy Paris University, Paris, France
[*] Author to whom any correspondence should be addressed.

E-mail: Kevin.De.Pauw@vub.be

Keywords: deep learning, biosignal control, scoping review, applied, human-computer interaction

Abstract

Objective. Biosignal control is an interaction modality that allows users to interact with electronic devices by decoding the biological signals emanating from the movements or thoughts of the user. This manner of interaction with devices can enhance the sense of agency for users and enable persons suffering from a paralyzing condition to interact with everyday devices that would otherwise be challenging for them to use. It can also improve control of prosthetic devices and exoskeletons by making the interaction feel more natural and intuitive. However, with the current state of the art, several issues still need to be addressed to reliably decode user intent from biosignals and provide an improved user experience over other interaction modalities. One solution is to leverage advances in deep learning (DL) methods to provide more reliable decoding at the expense of added computational complexity. This scoping review introduces the basic concepts of DL and assists readers in deploying DL methods to a real-time control system that should operate under real-world conditions. *Approach.* The scope of this review covers any electronic device, but with an emphasis on robotic devices, as this is the most active area of research in biosignal control. We review the literature pertaining to the implementation and evaluation of control systems that incorporate DL to identify the main gaps and issues in the field, and formulate suggestions on how to mitigate them. *Main results.* The results highlight the main challenges in biosignal control with DL methods. Additionally, we were able to formulate guidelines on the best approach to designing, implementing and evaluating research prototypes that use DL in their biosignal control systems. *Significance.* This review should assist researchers that are new to the fields of biosignal control and DL in successfully deploying a full biosignal control system. Experts in their respective fields can use this article to identify possible avenues of research that would further advance the development of biosignal control with DL methods.

1. Introduction

Biosignal control systems decode user intent from biological signals (also referred to as biosignals), enabling users to interact with electronic devices through their thoughts and other biological phenomena resulting from voluntary action intent. One of the main benefits of this new interaction paradigm is allowing people that suffer from a paralyzing condition to partially regain the use of their motor functionality through wearable robots and other assistive robotic devices (Ren *et al* 2019). Another population that might benefit from this technology are individuals with an amputation, allowing users to control a prosthetic device in a manner similar to their actual limb (George *et al* 2018). Ultimately, this technology could reach a point where it will also enable users to control everyday devices such as personal computers.

An example of this approach are brain–computer interfaces (BCI), which decode signals emanating from brain activity and increase the sense of agency for users (Caspar *et al* 2021). Therefore, creating a responsive control system is important to make users feel that they are in control of the device, enabling an optimal user experience. Alternatively, muscle activity measured through electromyogram (EMG) is another commonly used control signal (Ameri *et al* 2019) that has found success in robotics applications among others (Bi *et al* 2019).

However, at the current level of this technology, many applications still do not provide the user experience that is expected from commercial applications. While invasive approaches, which require a surgery to implant sensors, can provide higher quality signals, non-invasive methods are often more suited for general use. Current non-invasive methods are typically not accurate enough or introduce undesired latency, making existing interaction modalities more suited for these applications at the moment. To achieve the sense of agency that biosignal control could provide, better recognition methods are necessary and several issues need to be addressed.

Recognition of user intent from biosignals is typically achieved with artificial intelligence (AI) methods. These methods are trained on data originating from a subject performing an action related to the action that the algorithm should identify. Some of the most successful AI methods use deep learning (DL) to achieve state-of-the art results in their respective domains. Therefore, DL could be a good choice as a decoding method when implementing a biosignal based control system.

In this scoping review biosignal control methods that use DL are investigated. The overall aim of this review is to provide an overview of publications that build a proof-of-concept biosignal control system that uses DL methods for decoding user intent and identifying the most important gaps in the field of biosignal control.

1.1. Rationale

Multiple reviews and surveys discuss the different aspects of biosignal decoding. Some focus on DL for electroencephalogram (EEG) analysis (Thomas *et al* 2017, Craik *et al* 2019, Roy *et al* 2019, Merlin Praveena *et al* 2020), while others provide general overviews of machine learning (ML) methods used in the context of BCI (Ramadan and Vasilakos 2017, Rashid *et al* 2020). Others, like (Buongiorno *et al* 2019, Simão *et al* 2019) review EMG decoding methods and Rim *et al* (2020) reviews general biosignal decoding. However, most reviews lack an extensive discussion of the background knowledge that is required to develop new biosignal control systems. There is also a distinct lack of clear guidelines and best practices on how to gather and share data for biosignal control research. Several publications also do not report their results properly to allow for the reproduction of performed experiments and to provide support that the developed model is suited for real-time decoding in a real-world setting. Evaluation of research prototypes is also insufficient to take the system's technology readiness level[6] beyond a proof-of-concept (POC). These issues also impede the comparison of biosignal control systems.

This scoping review distinguishes itself from similar review articles such as Zhang *et al* (2020b) and Rechy-Ramirez and Hu (2021) by focusing on publications that implement and evaluate a real control application. The specific scope of this review covers biosignal decoding with DL methods, in the specific context of control applications, without limiting itself to just one type of application like Rechy-Ramirez and Hu (2021) or focusing solely on brain signals like Zhang *et al* (2020b). We also highlight some of the major differences between offline signal analysis (open-loop) and real-time decoding in a control setting (closed-loop) in section 5.2. To the best of our knowledge, no other review discusses this specific topic.

1.2. Objectives

The main objective of this review is to provide the necessary insights to understand DL and biosignal decoding concepts. The necessary knowledge to implement a biosignal decoding pipeline that incorporates DL methods will be presented for this purpose. Specifically, this scoping review will tackle the following research questions.

What can be achieved with DL for biosignal decoding? DL is currently a successful method in multiple research fields that outperforms most other ML methods. However, certain prerequisites should be met to effectively use DL methods. The problem at hand should also warrant the increased complexity that is introduced by DL methods. Ensuring that these conditions are met requires a good grasp on the basic principles of DL methods.

What are the challenges that the field is facing in terms of data gathering and processing? Gathering high-quality biological data is an important aspect of constructing any ML model that will be used in a biosignal control application. Typically, task-specific data should be gathered, as publicly available data is often unsuited for the training of a model for a specific control system. Following the right procedures greatly facilitates the later stages of data processing and ensures better reproducibility of the research if the data used for training a model cannot be shared publicly.

[6] For more information on the technology readiness level: https://ec.europa.eu/research/participants/data/ref/h2020/other/wp/2016_2017/annexes/h2020-wp1617-annex-g-trl_en.pdf.

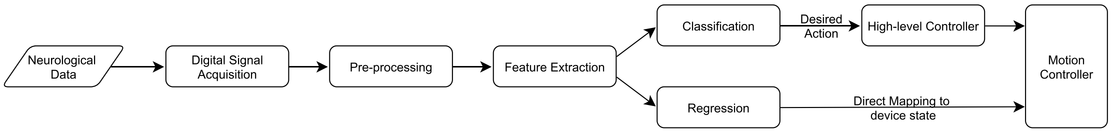


Figure 1. Pipeline for decoding user commands from a biosignal.

How to evaluate DL methods and control systems, and report for optimal reproducibility? DL models have many components and hyperparameters that need to be chosen by the implementer. Performance of models can sometimes greatly vary between software and hardware that are used in training and evaluation. Evaluating the performance of a model requires great care in selecting the appropriate metrics to report. Additionally, evaluating the full control system requires more investigating than only assessing the decoding performance of the DL model. Recommendations will be formulated regarding reporting standards in relevant scientific fields.

This scoping review describes the basic concepts of ML and DL, which should provide a deeper understanding of the techniques used in biosignal decoding research and DL methods. This knowledge should assist in implementing, evaluating and deploying DL models that will be integrated in biosignal control pipelines. Best practice procedures for gathering and pre-processing data and an overview of which DL models have been used in which biosignal control contexts are presented. A discussion of the evaluation methods for biosignal control applications highlights the key gaps in current methods. Finally, a literature-based guide to developing and evaluating a biosignal control application is presented.

2. Background

In this section, we give a brief overview of the methods that are employed in biosignal decoding applications. A more detailed background is presented in appendix.

2.1. Biosignal decoding

In biosignal control, the biosignals related to voluntary activity from a subject are measured to enable a user to control a device with their thoughts and movements. To identify the desired action from the user's biosignal activity, the signals are processed by a pipeline that consists of the steps depicted in figure 1.

After acquiring the signal, it typically needs to be cleaned because biosignals have a low signal-to-noise ratio (SNR). This means that there will be more noise present in the acquired data, making it harder to extract useful information. Therefore, a preprocessing step is necessary to improve SNR and yield a 'clean' signal that eliminates artifacts, such as those resulting from power supplies or unrelated muscle activity. Next, features are extracted that describe the relevant information contained in the signal.

To decode the user's intent from the extracted signal, an algorithm, usually implemented as a ML model, processes the signal. This can either be classification into a discrete set of classes or regression into a value that is relevant to the control application. Finally, the output is passed to the motion controller software that ensures the correct action is executed by the device.

The EEG is the most commonly used signal when using brain signals for control. Multiple control paradigms exist for EEG signals that influence the design of a control system (Ramadan and Vasilakos 2017, Rashid *et al* 2020). EEG can be measured invasively through implanted sensors or non-invasively, which is the approach we focus on in this review. Alternatively, the EMG is the most commonly used non-brain signal. It measures electrical activity resulting from muscle contractions. It is relatively simple to incorporate in existing robotic systems and it is easier to identify which muscle is active and subsequently map this activity to an action of the controlled device (Simão *et al* 2019). The choice between using EEG and EMG generally depends on the application.

Other biosignals are also leveraged, but to a lesser extent, as there are drawbacks that currently make them impractical compared to EEG and EMG. Alternative signals include magnetoencephalogram (Gross 2019), functional near infrared spectroscopy (Naseer and Hong 2015), and functional Magnetic Resonance Imaging (Sitaram *et al* 2007). Lesser known signal types are also becoming usable, such as acoustic resonance (Norman *et al* 2021), which uses the Doppler effect to detect changes in brain activity, and Photoplethysmogram (Han *et al* 2020), which uses optical reflections to monitor brain activity, in a similar way to near infrared spectroscopy. Another possibility is to monitor spinal cord activity using magnetospinography (Sakaki *et al* 2020) if one is only interested in lower-limb activity.

Some systems combine multiple signals to take advantage of the redundant information that might be lost when using only one signal type, but can reliably be extracted from another signal type. Such systems are referred to as hybrid BCI. One example of such a hybrid system is to combine EEG and EMG for movement detection (Leeb *et al* 2011, Loopez-Larraz *et al* 2018, Tortora *et al* 2020b), as this allows detection of motor planning from the EEG signal and determining the exact movement more reliably from EMG.

2.2. Machine learning

ML is a subfield of AI which consists of algorithms that extract knowledge from data and involves two steps. First, a *learning* phase consists of summarizing the data in some machine form. Once knowledge is extracted, a *prediction* phase uses it to produce values to solve a specific task in a computer system.

The most common form of ML is supervised learning, in which we assume that the data is presented as a set of input-output pairs, a *dataset*, which we call *labeled data*, as each input is labeled with its corresponding output (Caruana and Niculescu-Mizil 2006). Alternatively, unsupervised learning techniques do not use outputs for learning, but rather learn the (unknown) structure of the data.

Semi-supervised learning methods use both labeled and unlabeled data, usually to learn the structure of the training data to become able to generate more (artificial) training points (Aznan *et al* 2019), that are used for conventional supervised learning in a second learning phase. Self-supervised learning (Jing and Tian 2019) is a similar approach that is used to learn the relevant structure in EEG data by first learning an unsupervised pretext task, after which the model is further trained on the target task with labeled data (Banville *et al* 2020, Kostas *et al* 2021). The remainder of this review will focus on supervised learning methods.

2.3. Neural networks

Out of the many supervised learning approaches that exist, and are reviewed by Caruana and Niculescu-Mizil (2006), neural networks currently appear as the most promising one. There are several reasons for this: the neural network formalism is quite general, which means that many specific ML architectures can be seen as neural networks, therefore leveraging the extensive knowledge we have on them. Neural networks are also able to learn mappings between inputs and outputs in a highly general way, making few assumptions. Neural networks are therefore applicable to many tasks, even if the input is a time sequence, has a large dimensionality, consists of images or portions of videos.

Training a neural network consists of finding θ, the set of parameters (synonym weights) that minimizes a *loss*. Training a neural network is therefore an *optimization problem*, for which many well-known algorithms and software libraries exist of which PyTorch[7], in Python and C++, and TensorFlow[8], in Python, are the most well-known. Other less-known libraries exist in various programming languages, such as MATLAB or Java, but they do not have as many tutorials and tools available as PyTorch and Tensorflow. On embedded systems, CMSIS-NN (Lai *et al* 2018), by ARM, makes neural networks amenable for deployment on small microcontrollers.

The multi-layer perceptron Rumelhart *et al* (1985), MLP is the neural architecture that is most often used. Other commonly used architectures include convolutional neural networks (CNN) and recurrent neural networks (RNN), which each introduce new types of layers. In practice, neural networks can combine many layers of different kinds. In the papers that we review in this article, great care is always given to explain and motivate the choice of neural architecture. Designing a neural network requires experience, as there is no systematic approach. More information on neural networks and their architectures is presented in the appendix, section 'Neural networks'. We refer readers interested in knowing more than what we present to books such as Goodfellow *et al* (2016) and Aggarwal (2018).

A neural network's usefulness is directly related to it is ability to generalize from training examples to the general problem that needs solving. Generalization is particularly important in the engineering and medical fields, where data is costly to acquire, as it allows a neural network to produce better predictions in production (on unseen data) with fewer training data points. With neural networks, the main avenue to increase generalization is to decrease *over-fitting*. Over-fitting happens when a neural network remembers exactly what training input should learn which training output, without having actually made sense of the data. The network achieves a training loss close to 0, but produces garbage output on the testing set.

To avoid over-fitting, Batch Normalization (Ioffe and Szegedy 2015) considers the input of every layer in a neural network, and normalizes it so that, in expectation, the inputs of every layer has a zero mean and a unit variance. Dropout (Srivastava *et al* 2014) does not modify the values that flow through a neural network, but instead randomly disables neurons every time the network is evaluated during training. Both these methods contribute to improving training speed and generalization of the network, and ensure that the model converges to an optimal loss. Both batch normalization (Tayeb *et al* 2019, Tam *et al* 2020) and dropout (Gautam *et al* 2020, Tortora *et al* 2020a) are often used in biosignal decoding papers, sometimes both at the same time. Other normalization techniques are possible, such as L1-normalization or clipping the gradients (Zhang *et al* 2019a), but they have been superseded by Batch Normalization and Dropout.

[7] https://pytorch.org.
[8] www.tensorflow.org/.

2.4. Deep learning

DL is a form of representation learning (Bengio *et al* 2013) that introduces more than one level of abstraction to the learned features (Goodfellow *et al* 2016). This means that from the initially learned features, new features at a higher level of abstraction are extracted. If one considers the steps of the pipeline as a graph, then this graph can become much larger in the number of consecutive steps, or deeper, than with other techniques, and is therefore referred to as DL.

Most modern DL models are neural networks, however, other ML techniques can also qualify as DL. The most common type of DL neural networks at this time are the previously described CNNs (LeCun *et al* 1989). The subsequent application of convolutional layers results in a high-level representation of the input as stipulated by Goodfellow *et al* (2016).

A DL method that is useful in biosignal decoding is transfer learning (Fahimi *et al* 2019, Kostas *et al* 2021). With Transfer Learning, a model is first trained on general data from which the model aims to learn the structure of the data. After pretraining, a fine-tuning phase will further train the model with data related to the specific problem that the model should solve. This approach is useful when faced with limited data availability for the target domain and specifically for biosignal decoding when calibrating a model for a new user. Transfer Learning allows for the training of much larger models than if the model was trained directly on the task-specific dataset.

3. Methods

3.1. Search strategy

To gather the literature related to practical applications of biosignal control that use DL in their decoding pipeline, a systematic procedure was followed. Due to the interdisciplinary nature of biosignal control research, no standard best-practice for systematic data gathering currently exists. To ensure all relevant literature was identified by the search strategy, the search was based on the PRISMA method for systematic reviews and meta-analyses (Moher *et al* 2016).

Articles that were retrieved according to this method were categorized in two classes. The first type of articles are those that discuss the implementation and evaluation of a biosignal based control system. The second type of papers are those that present a DL model that is evaluated on offline biosignal data. The latter type of papers focus on methods related to biosignal decoding without deploying their methods in a real-world control system. Most limit themselves to offline evaluation of the decoding performance of their algorithm. This review focuses on the former type of articles, which are presented and analyzed in the results section. Of the latter type of articles, those deemed most relevant to this review are cited where appropriate. The inclusion and exclusion criteria that were used to select papers of interest are summarized in table 1.

To gather the literature about control applications that use DL to decode biosignals, the **PubMed**, **Scopus** and **IEEE Explore** literature repositories were queried. In order to ensure that the latest state-of-the-art methods are included in this review, the **ArXiv** preprint service was also queried. The retrieval of documents from the selected databases was performed with a custom Python script that queries the API endpoints that are provided by the respective repositories. For each database, all provided search fields were used, except for Scopus where the title, abstract, and keyword fields were used. The initial search was performed in November 2020.

The query that was used to retrieve publications is the following: (**Brain Computer Interface OR Electroencephalogra* OR EEG OR Brain Machine Interface OR Biological Signal* OR Magnetoencephalogra* OR MEG OR Electromyogra* OR EMG OR Human Computer Interaction* OR HCI) AND (Deep Learning OR Convolutional Neural Network* OR Recurrent Neural Network OR Generative Adversarial Network* OR GAN OR Auto Encoder OR Transformer Network) AND (Motor Image* OR Control* OR Applied**). The resulting search strategy is depicted in figure 2.

3.2. Literature synthesis and ideal publication profile

The final list of selected publications were reviewed in detail and relevant data items were extracted from each publication. These extracted data were used to generate the tables and figures presented in section 4. Analyses were performed in a Jupyter notebook (Kluyver *et al* 2016) with the Python[9] programming language. Statistical analyses were performed with the Seaborn (Waskom 2021) and Pandas (Reback *et al* 2020) software packages. Figures in section 4 were generated using the Matplotlib (Hunter 2007) software library.

Assessing the quality of publications in the multidisciplinary field of biosignal control is a nontrivial matter. There are no standardized guidelines available at the time of writing and most of the guidelines from specific fields are not entirely applicable to publications pertaining to the implementation and validation of hardware prototypes and complete processing pipelines. Therefore, different elements of the quality assessment guidelines for relevant fields of research were combined to design an *ideal publication profile*, i.e. what are the important data that a publication should provide in order to be relevant to each field that is implicated in biosignal control research.

[9] www.python.org/.

Table 1. Inclusion and exclusion criteria for the selection of relevant biosignal decoding papers.

Inclusion criteria	Exclusion criteria
• Use a DL model in at least one step of the biosignal decoding pipeline. • The biosignal decoding implementation is validated with a real-world control application.	• Invasive methods to measure biosignals. • Medical applications, such as sleep staging or pathology detection, that are unrelated to control. • Low-quality publications that are almost impossible to reproduce due to limited information.

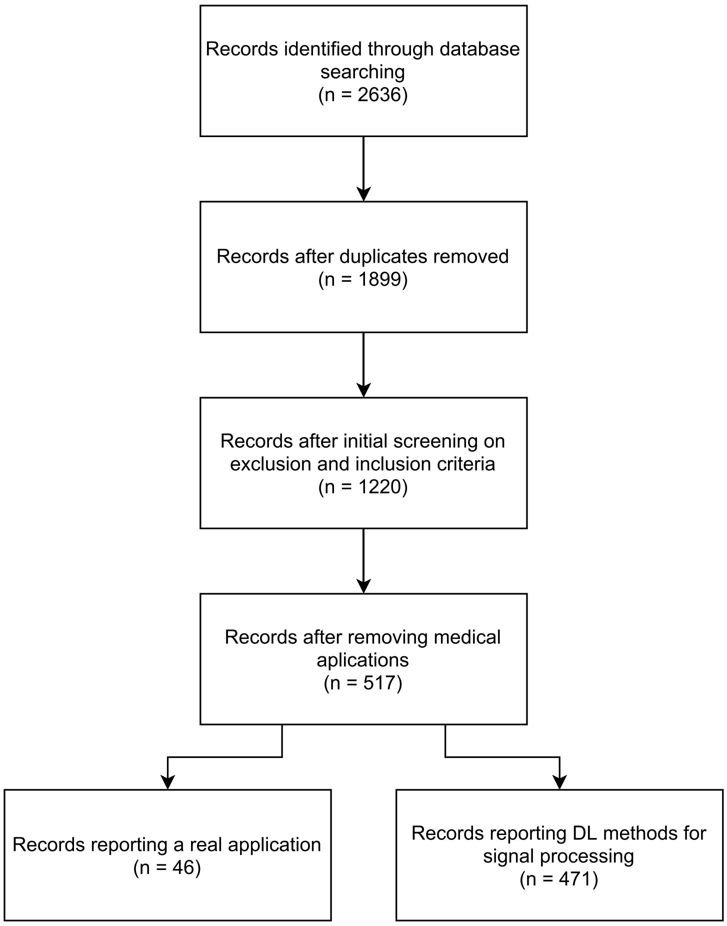


Figure 2. PRISMA search strategy flowchart.

Concretely, a publication that fits this ideal profile is easy to reproduce by providing the necessary code and data to deploy the developed system. The evaluation should also be easy to reproduce by detailing the tasks that were performed to evaluate prototypes. Additionally, a thorough evaluation of a POC prototype is required to assess the suitability for further development into a consumer-ready control system. Finally, if any of the information that is extracted for this review is missing or unclear, the publication is also marked as low quality.

Taking into account whether the publication was published in a conference or a journal was also considered as a criterion to fit the ideal publication profile. However, depending on the specific field, conferences might be considered as an equally impactful publication venue as opposed to the typical preference for publishing in journals above conferences. Therefore, it was determined that this aspect should not be a part of the ideal publication profile.

To provide a concrete example of the proposed ideal publication profile, a case study is performed with a publication that was selected from those that match the proposed profile most closely. This publication is discussed in the context of the proposed profile in section 4.2 and serves as the basis for a template of the ideal publication profile.

In this scoping review, the ideal publication profile criteria are used to classify the retrieved articles into three tiers according to how well they fit this profile. The lowest tier class is for papers that do not fit

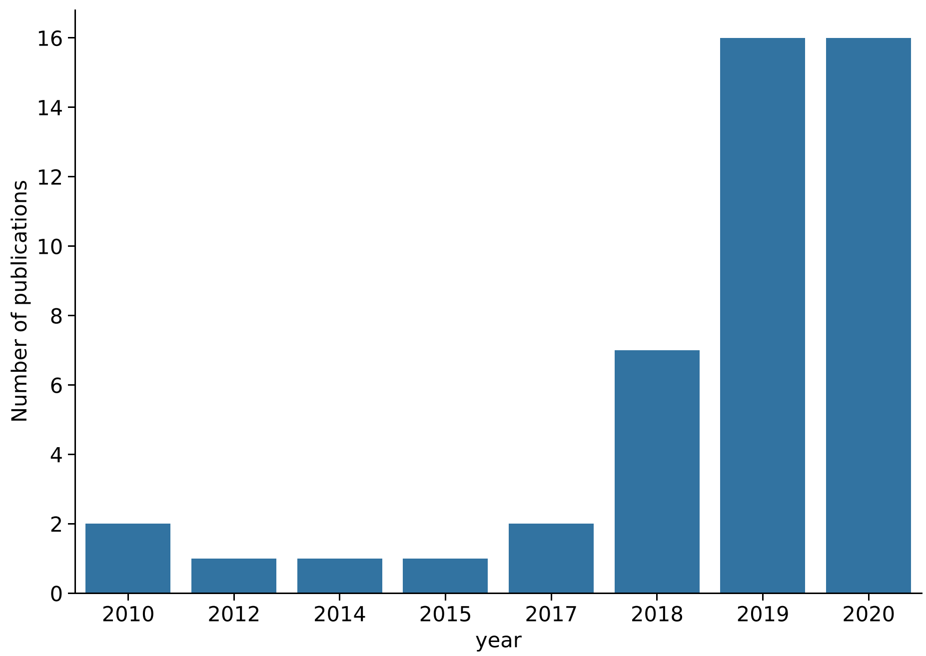


Figure 3. Publications per year about DL methods in biosignal control systems.

any of the ideal criteria, making them hard to reproduce and their evaluation is limited. The second tier are publications that do not have any issues, but could be improved by sharing code or data, or by performing a more thorough evaluation. Finally, those papers that match most of the ideal profile criteria are classified in the highest tier.

4. Results

4.1. Literature review

Following the systematic search strategy described in section 3.1, 46 papers remain that fit the selection criteria that were listed in table 1. Figures in subsequent sections include all papers, regardless of their ideal publication profile tier, unless stated otherwise.

Figure 3 emphasizes the increased interest in DL methods for biosignal control systems. This increase is in line with the rise in the use of DL methods in other fields of research. It demonstrates that research interest in using DL methods as a decoding algorithm for biosignal control increases yearly.

Twenty six publications (57%) are conference papers and nineteen (41%) are journal articles. The missing two percent is due to one publication being a preprint (George *et al* 2020) at the time of retrieval.

The application that is most common for EEG control is robot arm control. A total of five publications implement such a control system (Kuhner *et al* 2019, Shim *et al* 2019, Zied *et al* 2019, Alex *et al* 2020, Jeong *et al* 2020). Four of the retrieved publications use EEG signals to control a drone. Kobayashi and Ishizuka (2019) and Ishizuka *et al* (2020) implement a quadcopter control system, while Zhuang *et al* (2021) use EEG for controlling a ground vehicle and Aznan *et al* (2019) use it for controlling a mobile robot.

Another common application of EEG decoding are BCI spellers. Three publications in our result set are of this type (Nguyen and Chung 2018, Zhang *et al* 2019b, 2020a). Other applications of EEG control include driving assist (Lu *et al* 2020, Mourad *et al* 2020), embedded movement recognition (Wang *et al* 2020), smart home control (Zhang *et al* 2017), image reconstruction (Hernandez-Carmona and Penaloza 2019) and virtual reality control (Bevilacqua *et al* 2014, Karácsony *et al* 2019). One publication uses EEG for wheelchair control (Zgallai *et al* 2019) and another combines EEG with functional near-infrared spectroscopy to play a computer game (Makhrov and Denisova 2018).

Prosthesis (Shima and Tsuji 2010, Li *et al* 2017, George *et al* 2018, 2020, Teban *et al* 2018, Wan *et al* 2018, Jafarzadeh *et al* 2019, Liu *et al* 2019, Gautam *et al* 2020, Tam *et al* 2020) and exoskeleton (Orlando *et al* 2010, Xiang *et al* 2012) control exclusively uses EMG, with hybrid methods being used by Li *et al* (2018) and Ren *et al* (2019) respectively. Two publications use EMG for wheelchair control (Stroh and Desai 2019, Zhou *et al* 2019) and two also for operating a robotic arm (Hu *et al* 2015, Song *et al* 2020). Other applications that use EMG include drones (Redrovan and Kim 2018), real-time gesture recognition (Cote-Allard *et al* 2020, Zanghieri *et al* 2020), human-robot collaboration (Hanafusa and Ishikawa 2020), smartphone interaction (Cotton 2020) and virtual reality movement (Chiu *et al* 2019). There is

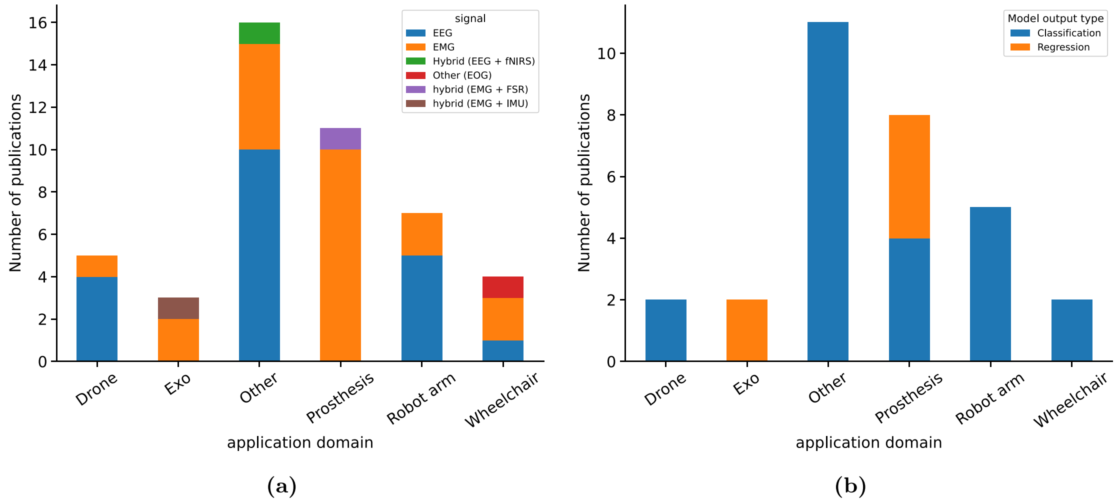


Figure 4. Distribution of applications found in the retrieved publications (a) divided by the signal that is used to decode user intention (EEG = electroencephalogram, EMG = electromyogram, fNIRS = functional near infrared spectroscopy, EOG = electrooculogram, FSR = force-sensing resistor, IMU = inertial measurement unit) (b) divided by whether the used model outputs classes or regression values (excluding those in the lower-tier ideal publication profile).

finally one special case where the authors used electrooculogram, i.e. eye movement, to operate a wheelchair (Ramakrishnan *et al* 2020).

Figure 4(a) displays the distribution of control signals over different applications. For the selected publications, exoskeletons and prostheses exclusively use EMG signals for control, with some publications combining EMG with another sensor type in a hybrid control system. Conversely, while EMG is still utilized in the other applications, EEG is the most common signal for these applications. Only for wheelchair control an equivalent mix of signals are utilized.

The application domains presented in figure 4(a) consist of high-level categories that can encompass multiple concrete applications. For example, a drone can be the well-known quadcopter (Redrovan and Kim 2018), but this can also refer to ground vehicles (Lu *et al* 2020) and robots (Aznan *et al* 2019) that can move according to the operator's commands.

All retrieved publications that discuss exoskeletons target the upper limbs (Orlando *et al* 2010, Xiang *et al* 2012, Ren *et al* 2019). Similarly, all prostheses applications aim for the control of a prosthetic hand (Shima and Tsuji 2010, Li *et al* 2017, 2018, George *et al* 2018, Wan *et al* 2018, Jafarzadeh *et al* 2019, Hanafusa and Ishikawa 2020, Tam *et al* 2020). In the retrieved publications, the number of degrees of freedom that are provided by the device vary among the devices used.

The *Other* category includes applications such as speller systems that allow paralyzed persons to communicate (Nguyen and Chung 2018, Zhang *et al* 2019b, 2020a), virtual reality control systems (Bevilacqua *et al* 2014, Karácsony *et al* 2019), driving assist systems (Lu *et al* 2020, Mourad *et al* 2020), and smart home control (Zhang *et al* 2017). Except for spellers, which are all EEG-based, these applications are the type that target biosignal control for everyday use by able-bodied users.

Depending on the application, the choice, and design of a high-level or low-level controller will determine if the decoding model should output classes or regression values. Figure 4(b) demonstrates that classification is the most common approach to signal decoding with prostheses also using regression in half of the publications. All publications on exoskeleton control use regression as the model output. Note that this figure does not include those publications that are deemed low-quality according to our ideal publication profile.

Distributing the publications based on whether their model outputs a class or regression value yields figure 5. This figures demonstrates that for EEG decoding classification is used exclusively in the retrieved publications. For EMG decoding, four out of twelve publications used regression. Note that this figure also excludes publications belonging to the low quality ideal publication profile category.

Another important aspect of the biosignal decoding pipeline is the choice of DL model that does the actual decoding of the signal. Figure 6(a) presents the distribution of DL model types over the selected publications. It indicates that CNN is by far the most widely used network architecture. Second are RNNs which can deal with varying length input sequences, but are more complex to use. Another interesting finding is that one publication uses reinforcement learning in combination with RNN (Zhang *et al* 2019b).

When considering the type of model with regards to the application domain, we can observe from figure 6(b) that CNNs are still the most common model for every application except exoskeletons. This is in line with the general distribution of model

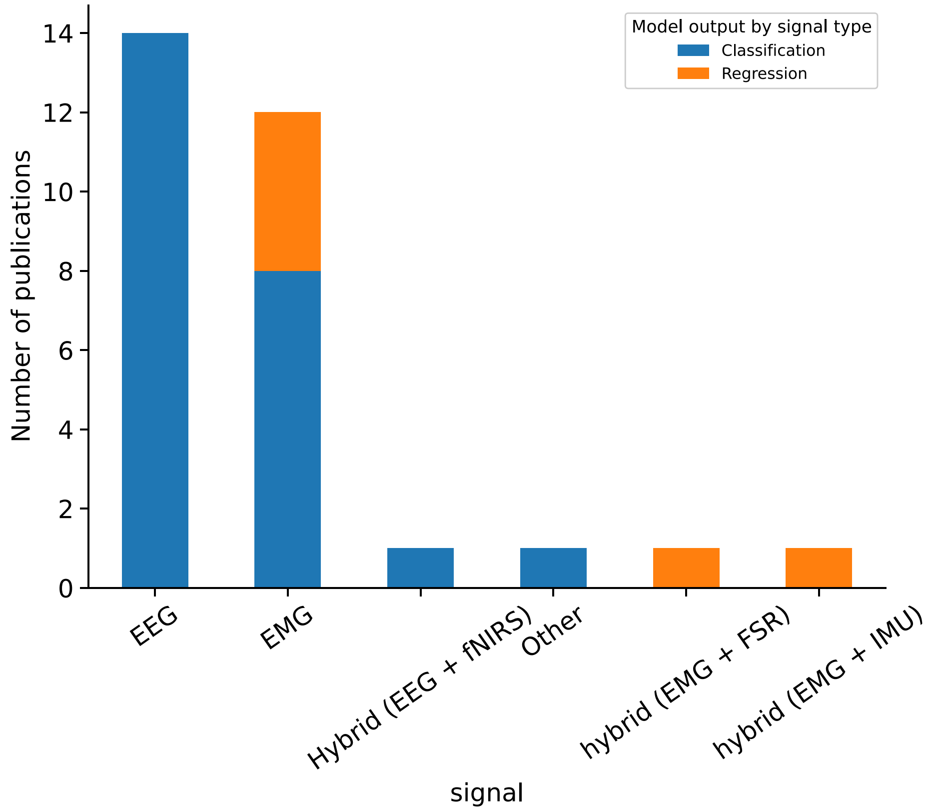
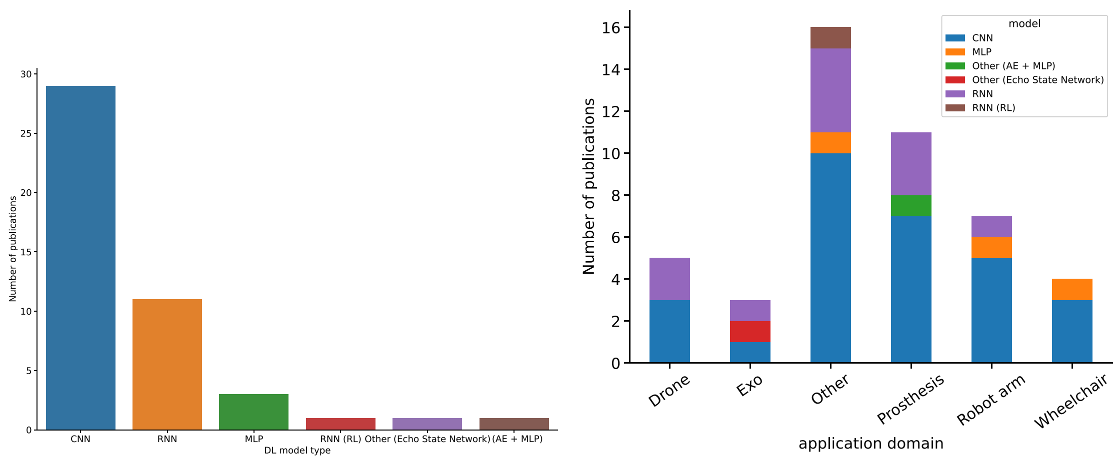


Figure 5. Distribution of classification or regression over the different signal types.



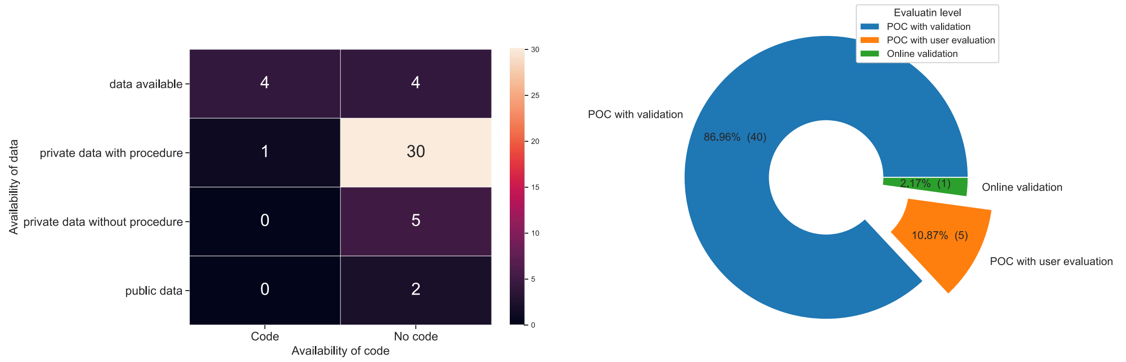
(a) (b)

Figure 6. (a) Distribution of DL models used to decode user intention in the published biosignal decoding pipelines (CNN = convolutional neural network; MLP = multilayer perceptron; RNN = recurrent neural network; RL = reinforcement learning; AE = autoencoder). (b) Distribution of applications, divided by the model used for decoding.

types and the disparity in exoskeletons could be attributed to the small sample size of the publication data.

Certain control systems also perform their decoding on embedded hardware, which introduces additional constraints regarding computational resources and power consumption. Eight publications perform their decoding on embedded hardware as opposed to a dedicated computer with specialized hardware. One paper even performs decoding in the cloud (Zhang *et al* 2019b).

There is a wide variety in the devices used for the acquisition of each signal. For EEG, the number of recorded channels (electrodes) varies from a single channel to 64 channels, with sampling frequencies ranging from 128 to 1000 Hz. For the types of electrodes used, there is no dominating type with both wet and dry, and active and passive electrodes being



(a) (b)

Figure 7. Reproducibility criteria for the publications. (a) Availability of code and data. (b) Pie chart of the proportion of publications that perform evaluation up to a certain level. The lowest level of evaluation is *Online validation* where real-time decoding was implemented but not used in a control system. The second level is *POC with validation* which means that while a proof-of-concept (POC) was implemented, it was only tested without assessing the user experience. The highest level of evaluation is *POC with user evaluation* where a user study with multiple participants is performed to assess the user experience of the control system.

used. For EMG, a distinction is made between separate electrode placement and sleeves or armbands consisting of electrode arrays, which are both used. Sampling frequencies are significantly higher for EMG devices with frequencies ranging from 1000 Hz up to 24 kHZ, with the exception of the Myo armband which is an armband consisting of eight EMG channels that are sampled at 200 Hz. (Ramakrishnan *et al* 2020) attached five electrodes to the participant's face to measure EOG and use it as the control signal, where EOG is typically used for artifact removal.

The following publications included a hybrid signal acquisition setup. Makhrov and Denisova (2018) combined 20 EEG channels with fNIRS at the F7, Fp1, G, Fp2 and F7 locations. In Li *et al* (2018) they measure EMG with a Myo armband in combination with force sensing resistors on the finger tips to determine grasping strength based on the combination of both signals. The final publication that takes a hybrid approach is Ren *et al* (2019), where two Myo armbands are measuring EMG and the device's builtin IMU sensors are leveraged to control the joint angles of an arm exoskeleton.

4.2. Ideal publication profile

In this subsection, figures and analyses related to the ideal publication profile are presented. Figure 7(a) displays a heatmap of the number of publications based on the availability of code and data that are used for experiments. It indicates that there are only four publications that provide both their code and data. Thirty publications share neither their code nor their data. However, these publications describe the data gathering procedure in detail, which allows reproduction at the cost of having to perform new experiments with human participants to gather the necessary data.

The evaluation levels of the selected publications are displayed on the pie chart in figure 7(b). This chart indicates that the majority of publications limit the validation of their POC to a simple test to demonstrate that their prototype works. Only five papers perform a full user evaluation of their prototype.

To further illustrate the ideal publication profile, we now present a case study of one publication that falls into the highest tier of our categorization. We provide arguments why this publication fits the ideal profile in contrast to other publications. The chosen publication for this case study is Kuhner *et al* (2019), which share all their code and provide a detailed description of their data gathering procedure, even though they did not share their data directly. One of the other aspects that sets this publication apart is the level of detail with which they describe their control system (in addition to sharing its implementation). The most important aspect that makes this into a model publication is the evaluation methodology. This publication is one of the few that perform a user study by putting the system into the hands of multiple users and evaluating the user experience in addition to objective performance metrics such as path optimality (number of steps taken compared to minimal number of steps necessary) and time to perform the task. Their user study is also the most extensive in both the number of participants and the diversity of tasks that are performed to evaluate the control system.

4.3. Prototype evaluation

While there are no standardized guidelines regarding the evaluation of prototypes, good-practice examples can serve as a template for the recommended evaluation methods for a biosignal control system. For this purpose, table 2 lists the five publications mentioned in figure 7(b) that were found to perform user studies

Table 2. Publications that include a user evaluation of their control system and characteristics of this evaluation.

Citation	Biosignal	Participant characteristics	Evaluation tasks	Evaluation metrics
Redrovan and Kim (2018)	EMG	Five users without great experience in quadrotors operation (no gender/age information)	Fly quadcopter in a predetermined flight cycle	Time to complete individual operation
Nguyen and Chung (2018)	EEG	Eight healthy volunteers, who have no problem with visual impairment (6M/2F, aged 24–32 years)	Spell the word **SPELLER** using the system	Number of commands Completion time (minutes) Accuracy Information transfer rate (bits min^{-1})
Karácsony *et al* (2019)	EEG	10 subjects (8M/2F, average age 25.3 $\pm$ 3.4 years)	Subjects are seated with their hands resting on the table and equipped with the BCI-VR system. Subjects play a VR game where three trials were performed, namely 2, 3 and 4-classes for control, each for 5 min, with 30 s break.	Score in the game (i.e. how well users could play the game) Questionnaire where users report experience on a Likert scale.
Shim *et al* (2019)	EEG	Five healthy right-handed participants (aged 25–31 years)	Perform predefined robot arm movements	Success rate
Kuhner *et al* (2019)	EEG	Online decoding: four healthy participants, all right-handed (1M/3F, aged 26.75 $\pm$ 5.9 years) User study: 20 able-bodied participants (17M/3F, aged 25–45 years) real-world scenarios: same as user study (not clear how many)	Online decoding: simulated goal formulation with the GUI User study: five predefined, simulated scenarios: move the robot to the garden, drink beer using a beer mug, arrange a red flower in a red vase, place a red rose on the couch table, and give a red wine glass with red wine to your friend real-world scenarios: fetch and carry task with disturbances; Drinking task	Online decoding: accuracy of the control; time it took the participants to define a goal; number of steps used to define a goal; path optimality, i.e. ratio of minimally possible number of steps to number of steps used; average time per step Participant ratings: rate if the displayed control opportunities offered by the GUI comply to their expectations from 1 to 5; rate the overall intuitiveness of the GUI in the range of 1–5; subjective quality of object references ranging from 1 to 5 real-world scenarios: accuracy and execution time

for their evaluation, which is an essential part of the ideal publication profile.

The selection of participants for a user study strongly depends on the target application. This is regrettably not the case in any of the encountered publications. However, we noticed that several papers do not evaluate the device on a target population. For example, both Nguyen and Chung (2018) and Kuhner *et al* (2019) only include able-bodied participants while their applications target paralyzed persons.

5. Discussion

The goals of this scoping review are to provide an overview of the literature discussing biosignal control with DL and to identify gaps in the state-of-the-art. Section 5.1 discusses the results from section 4 and proposes the potential implications of these findings. Section 5.2 provides recommendations for the development of practical biosignal control with DL in the form of a guide. Finally, section 5.3 presents the future perspectives of DL for biosignal decoding.

5.1. Implications from results

One remarkable finding is a substantial increase in publications targeting DL for intention decoding which is apparent in recent years. This trend can be related to the recent successes of DL in other fields of research, such as audio processing and computer vision (Baevski *et al* 2020, Srinivas *et al* 2021). Additionally, the development of custom software, such as the frameworks mentioned in section 2.2, and hardware, such as NVIDIA® Jetson™[10] or Intel® Neural Compute Stick[11] among others, facilitates the deployment of DL methods in real-world settings.

EMG is mainly applied for the control of exoskeletons and prostheses, because of the ease to map muscle activations to the degrees of freedom of the device and to integrate EMG sensors in the devices, while adding EEG sensors would require the addition of external acquisition devices to the system. We also noted that all prosthesis and exoskeleton applications targeted the upper extremities, which can be explained by the fact that decoding upper body movements is easier than lower body movement (Gandhoke *et al* 2019), as first discovered by Penfield and Boldrey (1937). For prostheses and exoskeletons, the limb that is replaced or assisted by the respective device is also of importance, as this will influence the design of the device and the degrees of freedom that are supported by the device.

The typical sampling rate for EMG acquisition is 1 kHz or higher. However, successful control was also achieved with the Myo armband[12] (discontinued), which supports a sampling rate of 200 Hz. This would likely be the sampling rate to expect in future commercial devices for everyday use. There is a trend towards electrode arrays that can be placed on a sleeve or an armband instead of single (or bipolar) electrodes. This can be attributed to the added robustness regarding electrode placement and drift of these methods, as the overall patterns that are related to a movement should remain largely the same (Farina *et al* 2010). Another advantage of sleeves and armbands is the ease of donning and doffing, and no need for skin preparation, with recent research even integrating this technology in E-textiles (Yin *et al* 2021). While the signal quality will generally be lower than with research-grade electrodes, these advantages make sleeves and armbands more suited for everyday use.

EEG is exclusively used in classification tasks, as demonstrated by figure 5, where the decoded intention can be mapped to a low-level device action or to a high-level command that is composed of a sequence of low-level actions. This sequence can be fixed or generated on-the-fly from the current device state to the requested device state. Such an approach decouples the number of values or classes that need to be predicted from the degrees of freedom of the device. Therefore, the applications that are controlled with EEG generally provide a discrete number of commands through their control interface.

There is also an interest in hybrid signal decoding for multiple applications. This approach is often useful in dealing with the low SNR of biosignals. The introduction of a redundant source of information can allow models to learn patterns that would otherwise be lost in the noise through sensor fusion (Makhrov and Denisova 2018, Ren *et al* 2019). Alternatively, the secondary signal can have a complementary purpose to add features to the system. In Li *et al* (2018), the force-sensing resistor sensors are used to provide tactile feedback when grasping with a prosthetic hand to improve the sense of agency of the user.

The dominance of CNN with regard to model choice can be attributed to the relative ease of use and the popularity of this architecture in other research fields that use DL. While RNN architectures have been successful in closely related fields such as speech recognition and natural language processing, they have only seen limited deployment in a biosignal decoding context. Typically, CNNs also have less trainable parameters which makes them less sensitive to small datasets and lower their computational requirements. Other architectures were investigated for biosignal decoding, but state-of-the-art research mostly focuses on CNN architectures (Buongiorno *et al* 2019, Roy *et al* 2019).

Unfortunately, only five papers fit the ideal profile that we defined for publications on control applications that use DL methods. From a reproducibility perspective, most publications can be reproduced from the provided descriptions. However, an ideal publication would share both their code and data. It is easier to reproduce an experiment if a public dataset is used or if the gathered data is provided. Open access of all relevant resources should be the goal for any publication that presents an implementation or new data.

Also related to the ideal publication profile, is the level of evaluation for the implemented prototype. Ideally, this should at least be a basic user study to validate that the control system improves the user experience of the targeted user group compared to alternative approaches. Only a small proportion of publications perform a user study, which emphasizes that there is a clear gap in current evaluation methodologies. The heterogeneity in evaluation methodologies also makes control system designs harder to compare without reimplementing the full control system. Therefore, there is a need for standardized evaluation guidelines for each application type. Having such guidelines should encourage researchers to adhere to a detailed user evaluation for POC prototypes and allows for the comparison of control systems for the same application types.

Furthermore, the evaluation tasks should also be aligned with the targeted use cases of the application. A large variety of tasks and the inclusion of edge-cases will provide most value towards the transfer of results from laboratory to real-life conditions. Performance of the control system is typically assessed by the time to complete the task or by the number of

[10] www.nvidia.com/en-us/autonomous-machines/embedded-systems/product-development/.
[11] https://software.intel.com/content/www/us/en/develop/hardware/neural-compute-stick.html.
[12] www.youtube.com/watch?v=A8lGstYAY14.

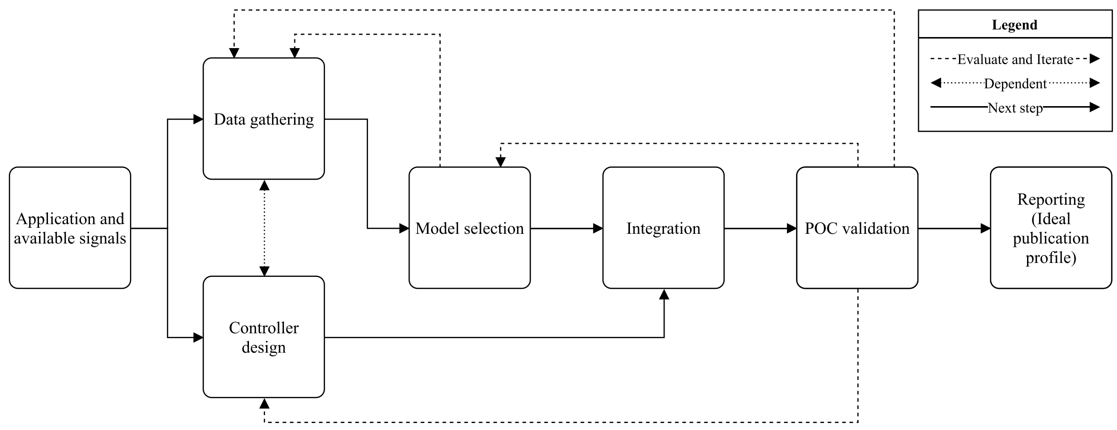


Figure 8. Flowchart for the recommended steps and their order for designing and implementing a biosignal control system.

steps/commands required to reach the end goal. Success rate can serve as an initial indication for a pilot study, but a complete user study should not solely rely on this metric as a measure of application performance. Ideally, objective metrics related to task completion should be supplemented with a subjective evaluation by participants, as the end goal should be the best user experience, which is a subjective concept. Possible approaches to subjective evaluation of the user experience are suggested in section 5.2.

The case study in section 4.2 proposes a template for an ideal publication that reports results from the evaluation of a biosignal control system. By following this template and performing extensive user-studies as presented in section 4.3 one can ensure optimal reproducibility. Taking all this together should bring us closer to the standardized guidelines that should enable the comparison of control systems without the need for new experiments.

5.2. A literature-based guide to DL for biosignal control

Building a complete biosignal control system requires several important decisions. This section will go through every step of the process and discusses insights, best practices, and pitfalls for each step. An overview of the recommended approach is depicted in figure 8.

Before starting the design of a biosignal control system, the application requirements and the available signals should be analyzed. As established in section 4.1, there is a close relationship between the device type that will be controlled and the signals that can be used for the control system. The number of commands depends on the number of degrees of freedom of the device. The choice of signals is also dependent on the user, e.g. EMG signals cannot be integrated in paralyzed individuals. After a thorough evaluation of the application requirements, the design of the controller software and the data gathering procedure can begin. Methods of software engineering are well-established in this regard and ensure a thorough requirements analysis (Kotonya 1998).

The first step in any ML application is the acquisition of relevant data to train the model. While public data is important for benchmarking and comparison of models, this data will usually not be sufficient when the aim is to implement a real-time application that will operate in real-world conditions. Therefore, data gathering will often be necessary to acquire data that reflects the application and enables the training of a performant DL model. However, to know the type of data that will be required, an initial design of the controller interface is necessary to determine the experimental protocol for data gathering. One should also keep in mind that DL methods need sufficient data, and data augmentation methods might be necessary (Corley and Huang 2018, Wang *et al* 2018). The more trainable parameters that a DL model has, the larger the amount of necessary data will be. For example, the original Transformer model consisted of 65 million trainable parameters and required 4.5 million sentence pairs to learn English–German translation (Vaswani *et al* 2017). Such extensive training also increases the time to train the model, even with high-performance hardware.

It should also be noted that supervised learning methods need a ground-truth label to train a model. This means that data gathering with the aim to use the data to train a ML model, will typically necessitate additional measurement devices to extract this ground truth. This should inform the position (in time) and labels of markers, and the size of a trial window. When movement is involved, motion capture solutions are often used for this purpose (Jeong *et al* 2020). Data engineering is another important aspect that needs consideration when gathering data for ML applications, as this will determine the expected data formatting when going to the real-time setting.

Next, a choice should be made whether to implement a high-level controller that will map a user command to a sequence of device actions or a low-level controller that can directly use the output from the decoder model to perform a device action. The former requires more implementation work, but allows for a reduction of the number of commands that need decoding when dealing with a high number of degrees of freedom (Kuhner *et al* 2019). The latter only needs to send the command to the device, but requires an output for each possible degree of freedom of the device (George *et al* 2020).

The choice of DL model is highly dependent on the application requirements and how the data is preprocessed. CNNs can often work with raw data that is only cleaned in the preprocessing step, while other models will typically necessitate feature extraction before passing the input to the ANN (Schirrmeister *et al* 2017). Alternatively, RNNs are also used for biosignal decoding, but these architectures typically need more technical knowledge to deploy and evaluate. The literature review clearly shows that CNN is the most deployed model and that biosignal decoding models are typically rather shallow, which can be attributed to the limited availability of data.

To obtain a good insight into the decoding performance of a DL model, one should evaluate more than just the classification accuracy on some test data. As stated in the background section, all possible evaluation metrics should be assessed to properly evaluate any ML model. However, offline (open-loop) validation of the model is insufficient to determine its' suitability for integration into a real-world (closed-loop) control system. It is generally accepted in ML that underspecification is a major issue when evaluating the performance of a model for a real-world application, because additional confounding factors affect the application in a real-world setting (D'Amour *et al* 2020). This is also a significant issue for biosignal control where many additional factors come into play when operating in a real-world setting. Reviews by Bi *et al* (2019), Rashid *et al* (2020)and Al-Saegh *et al* (2021) among others discuss the differences between offline and online systems in more detail. Therefore, stress tests should be performed to evaluate the model under conditions that reflect real-life conditions. Unfortunately, such stress tests have not been developed yet in the context of biosignal control. Finally, a thorough evaluation of the real-time performance is also necessary, which is currently lacking in many papers, as demonstrated by figure 2.

Integrating the selected model into a full control system is not a trivial matter either. For some applications, the model will be deployed on embedded hardware that is part of the controlled device. In this case, the different software components will run on separate hardware components, for which communication protocols will need implementing. Additionally, simply passing predictions from the model will often be insufficient in a real control system. A control strategy will be required that buffers the predictions that can then be used to generate commands for the device (Kuhner *et al* 2019, Tam *et al* 2020). This enables faster and more robust processing of the user intention.

Efforts are being made to develop standardized platforms that facilitate the deployment of control systems into real or simulated environments. Previous research solutions attempted to create common ecosystems for neurorobotic applications, in terms of open source frameworks such as the Neurorobotics platform (Falotico *et al* 2017), ROS-Health (Beraldo *et al* 2018) and its successor ROS-Neuro (Tonin *et al* 2019). ROS-Neuro was already succesfully used in a DL context (Valenti *et al* 2020, 2021). For more general applications, the OpenVibe platform provides several environments and integrates with a large variety of devices for BCI control experiments (Renard *et al* 2010).

Decoding user intent is often not the only aspect of a biosignal control system. Feedback to the user is also important to engender a feeling of agency for the user. This feedback should manifest within a certain timeframe to avoid latency issues that might diminish the sense of agency (Caspar *et al* 2021). Typically, feedback takes the form of tactile responses and audiovisual cues that indicate successful decoding of a command. This additional feedback can be provided before the effect of the user's command can be observed. Integrating biosignals with other interaction modalities is also essential for certain applications such as robotic arm control. Keeping the number of commands low is often favorable to provide an ideal user experience. For example, (Kuhner *et al* 2019) use depth sensing cameras to track object and robot positions, which allows users to think of which object they want to grasp, without communicating the object's position to the robot control system.

User-training will also play an important role in the practical deployment of an application (Kuhner *et al* 2019). Users will not want to use a system that requires a long training time before being usable on a regular basis. Due to the non-stationary nature of the signals, users might have to re-train themselves and re-calibrate their device on every usage. These challenges are highlighted by the Cybathlon competition where one of the events is a race where participants have to complete a BCI task in the shortest time (McFarland and Wolpaw 2018, Perdikis *et al* 2018, Hehenberger *et al* 2021). Therefore, user training should be considered when designing the final control system, which is extensively discussed by Roc *et al* (2021).

Finally, the implemented POC should be evaluated for usability. Simply reporting the decoding performance (even in real-time) is not sufficient to validate a DL model that is intended for use in a biosignal control system. Since the end goal is to provide the optimal user experience, this should be thoroughly evaluated. At lower technology readiness level validation studies should be conducted, whereas at higher technology readiness level user evaluation studies can be planned with bigger samples. Until now, no standardized procedures exist that would allow application-agnostic comparison of a control system, however Novak and Riener (2015) already proposed standardized evaluation procedures for exoskeletons and prostheses, and a systematic review by De Bock *et al* (2022) reviews benchmarking of occupational exoskeletons.

Besides evaluating objective performance of an application, user experience should also be considered, as well as the user intent and acceptance of the novel technology. Elprama *et al* (2020) investigate methods to evaluate user acceptance in the context of industrial exoskeletons. Methods from human-computer interaction (Dix *et al* 2003) and usability engineering (Nielsen 1994) could suggest possible approaches to validate a control application and determine which questions to ask participants for an ideal evaluation of the user experience. Since user opinions can be subjective, it is essential to have a heterogeneous group of participants, as mentioned in section 4.3.

We end this subsection with a summary of general advice that can aid in the development of a control system and ensure that published results are reproducible and comparable. General tips:

- *Software/Hardware co-design*: when aiming for embedded decoding of user intent, it is recommended to design the hardware setup and software decoding pipeline together (Tam *et al* 2020, Wang *et al* 2020). This can avoid issues when deploying the control system to the application setting and forces the software design to take hardware constraints into account.
- *Limit required preprocessing*: extensive preprocessing will yield clean signals with a high SNR, but this usually comes at the cost of expensive computational requirements that take resources and time. It will often be necessary to balance preprocessing requirements with latency and power consumption constraints.
- *Perform user studies*: a user study is the only way to really evaluate the user experience of a control system. While validation is sufficient for an initial prototype, evaluation in the lab will be necessary at a higher technology readiness level. Finally, in-field evaluation is necessary to assess user experience in real-world operating conditions.
- *Share resources*: sharing resources such as code and data can assist other researchers that are starting the development of their own control system and enhance the reproducibility of the results. This also enables a better comparison of different approaches.
- *Use standardized validation procedures*: using standardized validation methods allows for the comparison of different approaches without the need to fully reproduce the experiments. This also enables the systematic review of methods, which could assist in identifying best practices.

5.3. Future perspectives

Recent advances in DL methods in related fields can be leveraged to make real-time decoding more reliable, faster and use less resources. There are still several methods that are yet to be applied to biosignal decoding. For example self-supervised learning was recently used (Kostas *et al* 2021) for biosignal decoding, and there are multiple advancements in transfer learning (Srinivas *et al* 2021) that could help in mitigating the small data issue, with some techniques even allowing transfer between two completely different settings (Lu *et al* 2021). By leveraging these advancements, the calibration of the control system to a new user and the required user training could be minimized thanks to the strong generalization power of DL. In combination with the advent of embedded DL, these results indicate that DL will likely become the preferred method for biosignal decoding.

Another interesting development in the research and development of ML pipelines is AutoML (He *et al* 2021). This new approach to the design of ML pipelines allows for automated exploration of alternative methods for the different components of the pipeline. While this has not been developed yet in the context of biosignal control, it has proven to be a valuable tool for researchers and companies that are lacking the expertise to develop the whole pipeline themselves. Companies such as Google are now providing AutoML solutions as part of their cloud infrastructure[13]. Having such a framework for biosignal decoding pipeline development could significantly boost the deployment of real-world applications.

Developing standardized methods for both data gathering and prototype validation should also contribute towards the commercialization of biosignal control systems. The current lack of standardized data gathering procedures means that engineers need to design a new experimental protocol for each new POC. An advantage of standardized data gathering procedures is that data engineering for ML training could be automated.

The same limitations apply for the design of the experimental protocol of a validation or evaluation study. When the control systems for a specific application are evaluated with the same procedures, different approaches can be better compared without

[13] https://cloud.google.com/automl.

the need to reproduce the baseline. The development of stress tests that benchmark the model performance under specific conditions could also be a great contribution. For example, stress tests that evaluate the performance of a model across different sensor devices or when using different numbers of sensors could be useful to determine whether additional data gathering is necessary when new acquisition devices become available. Another example would be to evaluate the performance of a model on data from a participant that was not included in the training data, which could be an initial benchmark for the expected amount of calibration that will be necessary for new users of the control system.

Finally, new advancements in sensor and robotics hardware should improve decoding performance and user comfort even further. Current medical systems are accurate, but require a setup that is not acceptable for a consumer device. To get biosignal control accepted in the mainstream market, the control systems should be plug-and-play, requiring minimal user setup. Once this is achieved, it should only be a matter of making the hardware affordable and developing worthwhile applications. Afterwards, biosignal control should become possible for consumer applications such as VR control and everyday computer interaction.

6. Conclusion

This scoping review presents an overview of the theoretical background that is necessary to understand DL and biosignal decoding methods, and defines the scope of the field of biosignal control with DL decoding methods. After reviewing publications that applied DL for biosignal decoding in a POC application, gaps in the current state-of-the-art were identified and discussed. A bottom-up guide to implementing a biosignal control system with DL was extracted from the literature.

Biosignal control is becoming an increasingly viable alternative to classic interaction modalities, especially for robotic devices that are directly controlled by an operator. However, while the technology is already being used in medical devices, it is currently limited to stationary settings within controlled environments where biosignals can be measured with medical-grade sensors. Conversely, mobile commercial sensor systems exist, but they are unreliable for more advanced intention decoding purposes.

To bridge this gap, DL can be used to decode user intention from biosignals. DL methods have received increased attention in recent years thanks to the success of these methods in other research fields. However, DL is a complex and data-intensive method that usually requires more computational resources than the classical ML techniques. Even so, thanks to new advancements in both hardware and software technologies, DL will likely become the go-to method for user intent decoding from biosignals.

Finally, there currently is a lack of standardized protocols for evaluating research prototypes. Standardization of these protocols should allow for a better comparison of the performance of biosignal control systems and assist researchers in determining the necessary steps to validate their prototypes. Leveraging advances in DL and standardizing evaluation procedures should bring us closer to commercial biosignal control for everyday use.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors would like to thank Dr Bruno Tassignon for his advice concerning the design of the utilized systematic literature search strategy. We would also like to acknowledge María Alejandra Díaz for her assistance with the screening of publications.

Appendix. Detailed background

One of the objectives of this article is to allow experts in non-computer science domains to be able to assess the use of DL techniques for solving biosignal decoding tasks. Instead of enumerating a dense list of citations, leaving most of the work of understanding and summarizing DL papers to the reader, we propose in the following sections a guided walk-through of DL and related fields.

The biosignal decoding pipeline

In biosignal control, the biosignals related to voluntary activity from a subject are measured to enable a user to control a device with their thoughts and movements. To identify the desired action from the user's biosignal activity, the signals are processed by a pipeline that consists of the steps depicted in figure A.1.

As the first step in every biosignal decoding pipeline, a digital representation of the signals of interest is acquired through sensors that are represented as data channels. However, biosignals typically have a low SNR. This means that there will be more noise present in the acquired data, making it harder to extract useful information. Therefore, a preprocessing step is necessary to improve SNR and yield a 'clean' signal that eliminates artifacts, such as those resulting from power supplies or unrelated muscle activity. This is typically achieved by filtering out frequencies that fall outside the frequency spectrum

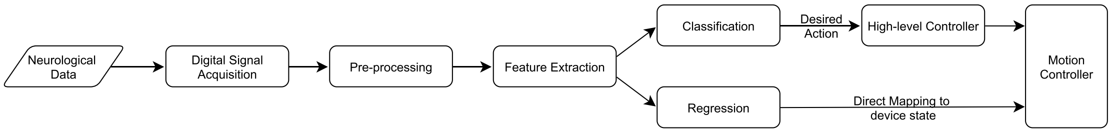


Figure A.1. Pipeline for decoding user commands from a biosignal.

that is known to be associated with the measured signal. For more information on filters and digital signal processing, interested readers are referred to Orfanidis (1996). Another important preprocessing step is windowing the continuous signal into discrete segments. Several windowing methods exist and are reviewed by Podder *et al* (2014).

From the resulting (clean) window, referred to as a trial in the context of biosignals, features are extracted that represent the information that is contained in the signal. Depending on the chosen signal, several feature extraction methods exist. However, DL methods automatically handle feature extraction when given raw data, which is detailed in section 'Deep learning'. Specific techniques fall outside of the scope of this review and are therefore not discussed and interested readers are referred to Rashid *et al* (2020) for an overview of existing feature extraction methods.

To identify the user intent from the extracted features (or raw signal data), an algorithm, which typically uses ML, computes a value or class label that is associated to the desired action. With a classification method, the predicted class is passed to a high-level controller that maps the predicted class to an action that the device can perform. For example, by mapping a specific movement intention class to a graphical user interface action (Kuhner *et al* 2019). When using regression, the predicted value could either be a threshold to perform a fixed action, or directly mapped to a continuous value that is understood by the low level motion controller. A common regression control strategy maps exoskeleton joint angles to those of the user (Ren *et al* 2019).

Designing a good controller is an essential step in applying biosignal control. On the one hand, the controller determines what type of model should be used to process the measured signal. On the other hand, the interaction capabilities with the device are entirely determined by the supported controller interface.

Biosignal modalities

Several types of biosignals can be used to control electronic devices. The most commonly used signal in BCI applications is the electroencephalogram (EEG), which measures electrical activity from firing neurons. It has a high temporal resolution and can be measured with relatively low-cost devices. Multiple control paradigms exist for EEG signals that influence the design of a control system. Several types of EEG paradigms can be distinguished and are discussed in more detail in other reviews of Ramadan and Vasilakos (2017) and Rashid *et al* (2020).

EEG can also be measured with implanted electrodes to yield a higher SNR and an increased spatial resolution. Modalities include the electrocorticogram where electrodes are implanted on the top of the cortex, right under the skull. There is also the possibility to implant electrodes directly in the brain matter (intracortical EEG) to further increase SNR and spatial resolution. However, such invasive approaches fall outside of the scope of this article.

For robotic control, the electromyogram (EMG), which measures electrical activity from muscle contractions, is most commonly used, because it is relatively simple to incorporate in existing robotic systems and it is easier to identify which muscle is active and subsequently map this activity to an action of the controlled device (Simão *et al* 2019). The choice between using EEG and EMG generally depends on the application.

Other biosignals are also leveraged, but to a lesser extent, as there are drawbacks that currently make them impractical compared to EEG and EMG. Alternative signals include magnetoencephalogram (Gross 2019), functional near infrared spectroscopy (Naseer and Hong 2015), and functional Magnetic Resonance Imaging (Sitaram *et al* 2007). Lesser known signal types are also becoming usable, such as acoustic resonance (Norman *et al* 2021), which uses the Doppler effect to detect changes in brain activity, and Photoplethysmogram (Han *et al* 2020), which uses optical reflections to monitor brain activity, in a similar way to near infrared spectroscopy. Another possibility is to monitor spinal cord activity using magnetospinography (Sakaki *et al* 2020) if one is only interested in lower-limb activity.

Finally, some systems combine multiple signals to take advantage of the redundant information that might be lost when using only one signal type, but can reliably be extracted from another signal type. Such systems are referred to as hybrid BCI. One example of such a hybrid system is to combine EEG and EMG for movement detection (Leeb *et al* 2011, Loopez-Larraz *et al* 2018, Tortora *et al* 2020b), as this allows detection of motor planning from the EEG signal and determining the exact movement more reliably from EMG.

Machine learning

AI is a wide domain of research in Computer Science. It has no precise definition, but Nilsson (2009) proposes to define AI as any method that makes a computer able to perform an intelligent task, that is, acting on inputs in the pursuit of a goal.

This definition of AI encompasses many subfields, such as planning and optimization (scheduling tasks, finding the shortest route in a city), natural language processing (text-to-speech, speech-to-text, parsing, understanding, translation), logic reasoning and inference (expert systems), and machine learning, on which we now focus.

ML is a subfield of AI which consists of algorithms that extract knowledge from data and involves two steps. First, a *learning* phase consists of summarizing the data in some machine form. Once knowledge is extracted, a *prediction* phase uses it to produce a value to solve a specific task in a computer system.

A typical ML pipeline consists of several main steps, being preprocessing, feature extraction, prediction, and sometimes postprocessing. Depending on the task, preprocessing can sometimes be omitted, as much as it could be an extensive and important part of the pipeline. Postprocessing is generally rare, as algorithms are usually trained to directly predict the value(s) of interest.

The most common form of ML is supervised learning, in which we assume that the data is presented as a set of input-output pairs, a *dataset*, which we call *labeled data*, as each input is labeled with its corresponding output. For instance, inputs can be fragments of signals acquired by electrodes, and outputs can be corresponding annotations, such as the intent of the user at that time, or a movement that was performed (Yohanandan *et al* 2018).

During learning, the supervised learning model will consider the input-output pairs, and will try to minimize the error between what it *predicts* for a given input, and the actual output for that pair. Methods and algorithms that allow to concretely perform this learning operation are presented in the following sections, and are extensively reviewed by Caruana and Niculescu-Mizil (2006).

Once the algorithm has learned, a form of summary of the input-output pairs has been produced. It can be the weights of a neural network, decision boundaries in a decision tree, means and variances in a Gaussian Mixture Model (Rasmussen *et al* 1999), or the equation of a line in Support Vector Machines. This knowledge can be used to perform *predictions* on new inputs, previously unseen by the algorithm but drawn from the same distribution (i.e. if the algorithm has been trained on brain signals, it must predict on brain signals, not electrocardiogram data).

The objective of supervised learning is to achieve high-quality predictions on previously unseen data. The main difficulty is that this previously unseen data usually has no known output (it only consists of inputs). It is therefore impossible to directly evaluate how good the prediction of a supervised learning algorithm is. Methods to estimate this quality exist, and are presented in the book by Aggarwal (2018). They mainly consist of using a *test set*: some of the input-output pairs are kept aside, not used for training, but are used after training to compute the difference between the output predicted by the algorithms on unseen data, and the known actual output. The ability for an algorithm to make good predictions on the test set is known as its *generalization* ability.

Unsupervised learning techniques do not use outputs for learning, but rather learn the (unknown) structure of the data. For example, clustering methods will try to form groups in the training data according to some criterion and try to determine which group a previously unseen example belongs to after training. Semi-supervised learning methods use both labeled and unlabeled data. Their objective is to learn a supervised learning task, even in cases where only a small amount of training data is available. Usually, semi-supervised approaches learn the structure of the training data to become able to generate more (artificial) training points (Aznan *et al* 2019), that are used for conventional supervised learning in a second learning phase. Self-supervised learning (Jing and Tian 2019) is a similar approach which is currently gaining traction in the larger ML community. This technique was previously used to learn the relevant structure in EEG data by first learning an unsupervised pretext task, after which the model is further trained on the target task with labeled data (Banville *et al* 2020, Kostas *et al* 2021). The remainder of this review will focus on supervised learning methods.

Neural networks

Out of the many supervised learning approaches that exist, and are reviewed by (Caruana and Niculescu-Mizil 2006), neural networks currently appear as the most promising one. There are several reasons for this: the neural network formalism is quite general, which means that many specific ML architectures can be seen as neural networks, therefore leveraging the extensive knowledge we have on them. Neural networks are also able to learn mappings between inputs and outputs in a highly general way, making few assumptions. Neural networks are therefore applicable to many tasks, even if the input is a time sequence, has a large dimensionality, consists of images or portions of videos.

Despite the name of neural networks, that may imply a type of opaque and magical approach, neural networks are a perfectly sound mathematical approach to ML. Neural networks consist of a parametric function $\hat{y} = f(x, \theta)$, a function that computes an output $\hat{y}$ from an input x and a parameter θ. The exact equation of the function can be anything, but usually consists of a sequence of matrix multiplications and activation functions, also known as a Perceptron (Ramchoun *et al* 2016). Convolutional neural networks (CNN) are often used when the input x is an image (LeCun *et al* 1995). Recurrent neural networks are able to process inputs that are sequences of numbers (or images) of varying length (Hochreiter and Schmidhuber 1997). In section 'Neural architectures', we provide more details about these neural network architectures, and identify the numbers that appear in their equations, that can be learned collectively form θ.

Training a neural network consists of finding θ, the set of parameters (synonym weights) that minimizes a *loss*. Training a neural network is therefore an *optimization problem*, for which many well-known algorithms and software libraries exist. We now describe what these algorithms do for readers interested in a detailed mathematical background, but practical applications of neural networks do not depend on the understanding of this section, but can rely on the libraries we present later in this section.

In most supervised learning settings, the loss is the Mean Squared Error, $\mathcal{J} = \mathbb{E}_{x,y \sim D}(y - f(x, \theta))^2$. Minimizing the Mean Squared Error leads to finding θ such that the outputs predicted by the network are as close as possible to the actual outputs, as appear in the dataset D. Other losses exist, and are used in specific cases, such as when the output of a neural network is a discrete probability distribution.

The algorithm that is the most often used to train a neural network is gradient descent. The set of operations (matrix multiplications, hyperbolic tangents, etc) in a neural network are all differentiable. It is therefore possible to compute $\nabla_\theta \mathcal{J}$, the gradient of the loss with regards to the parameters of the neural network. By updating the parameters in the opposite direction of the gradient, $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{J}$, with α a positive learning rate close to 0, the neural network becomes slightly more accurate at predicting outputs. Stochastic Gradient Descent consists of computing that gradient, and moving the parameters, repeatedly until the accuracy of the neural network stops improving (Bottou 2012). Recent advances, such as the Adam optimizer (Kingma and Ba 2014), still follow the same approach, but move the parameters in a slightly smarter way, leading to faster training of the neural network.

By explaining that neural networks are parametric functions, and that training them consists of computing the gradient of a loss with regard to the parameters and using it to update the parameters so that the loss decreases, we hope to have demystified neural networks, that are often presented as magical black-boxes in the industrial and medical domain, or in the media. To further help people interested in using neural networks for biosignal decoding applications, we also mention that building neural networks (defining the function f they compute), computing the gradients, and performing Stochastic Gradient Descent, is all automated in several well-regarded software libraries. PyTorch[14], in Python and C++, and TensorFlow[15], in Python, are the most well-known. Other less-known libraries exist in various programming languages, such as MATLAB or Java, but they do not have as many tutorials and tools available as PyTorch and Tensorflow. On embedded systems, CMSIS-NN (Lai *et al* 2018), by ARM, makes neural networks amenable for deployment on small microcontrollers. Using one of these frameworks is a must when implementing neural networks in any setting to avoid unnecessary work to implement the low-level components, and ensure their correct implementation.

Neural architectures

In this section, we discuss several neural architectures, that define the actual computations performed in a neural network, to map its input to the output. The choice of neural architecture is mostly influenced by the kind of input given to the network, and, in the next subsections, we will detail when to use which neural architecture.

Multi-layer perceptron (MLP)

The multi-layer perceptron (Rumelhart *et al* 1985) is the neural architecture that is most often used. It assumes that the input x of the neural network is a one-dimensional vector of floating-point values. Multi-layer perceptrons (MLPs) can therefore take as input sensor readings, or fixed-length windows of signals.

An MLP has a structure inspired from how neurons are organized in the neo-cortex of the human brain, hence the historical use of the name 'neural network'. The network is organized in a sequence of layers, each layer consisting of neurons (see figure A.2). Each neuron of a given layer is connected to all neurons of the previous layer, through connections whose importance (or weights) are learnable (part of the set of parameters of the network). Practically, a neuron produces a floating-point value, computed as $\sigma(\sum_j w_j h_j)$. j represents an index that enumerates the neurons in the previous layer, w_j is

[14] https://pytorch.org.

[15] www.tensorflow.org/.

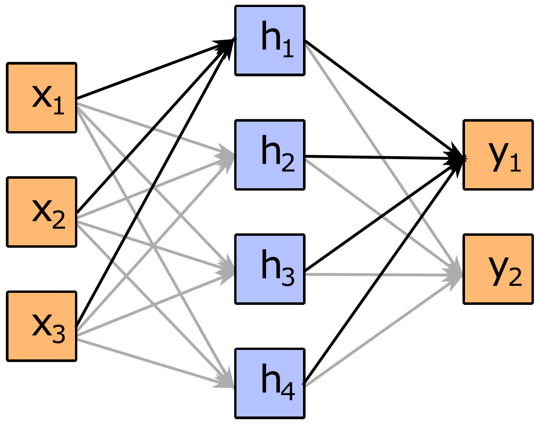


Figure A.2. Graphical depiction of a simple multi-layer perceptron with one hidden layer. Each arrow has a learnable weight (a parameter of the network). In this particular network, the output $y_1 = w_{h11}h_1 + w_{h21}h_2 + w_{h31}h_3 + w_{h41}h4 = w_{h11}\sigma(w_{11}x_1 + w_{21}x_2 + w_{31}x_3) + w_{h21}\ldots$.

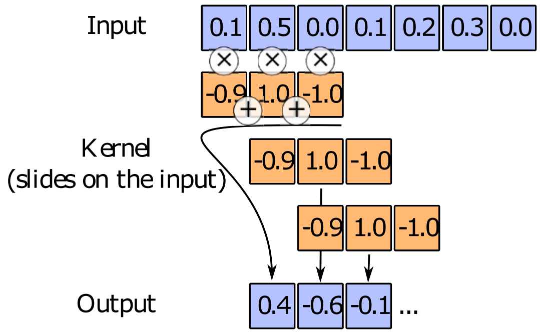


Figure A.3. A one-dimensional convolution (as defined in the neural network literature) of an input with a 3-elements kernel. The kernel slides on the input, leading to the computation of a succession of partial dot products.

the learnable weight of the connection between the jth neuron in the previous layer, and h_j is the value of that neuron. σ is an *activation function*, whose purpose is to make the neural network a non-linear function. Common activation functions include the hyperbolic tangent, the sigmoid function $\left(\frac{\tanh(x)+1}{2}\right)$, or ReLU, the rectified linear unit $(\max(x,0))$.

Compute-efficient implementations of neural networks observe that the output of a neuron is computed from a dot product (between the output of every neuron in the previous layer, and the weights of the connections), and that computing the value of every neuron of a given layer can be implemented as a matrix multiplication between a *weight matrix*, and the values of all the neurons of the previous layer. As such, from a mathematical standpoint, an MLP is a differentiable function that computes a series of matrix multiplications and activation functions.

The last questions that remain are how to choose how many layers should be included in an MLP, how many neurons each layer should consist of, and what the activation functions should be. There is unfortunately no recipe for choosing the structure of a neural network, but the book on Deep Learning (Goodfellow *et al* 2016) mentions a few approaches that, by looking at what has to be learned, allows to produce a reasonable neural architecture.

Convolutional neural networks

The layers of MLPs perform a matrix multiplication that allows every floating-point value in the previous layer to influence every floating-point value in the next layer, in an independent way. This maximizes the representative power of the network (the complexity of the functions it can learn), but is also costly to compute, and may be too flexible for fast and efficient learning. Convolutional layers (LeCun *et al* 1995) replace this matrix multiplication with a signal correlation operation: the input (1-, 2- or 3-dimensional) is correlated with a *learnable kernel*, leading to the output of the layer. Figure A.3 shows an example of a 1-dimensional correlation between an input and a 3-elements kernel. Keen observers can see that the operation is indeed a (cross) correlation, and not a convolution, as the name 'convolutional neural networks' may indicate. A convolution would have flipped the kernel before performing the same operation. This small difference may have historically been missed by researchers on CNNs, who then named their networks improperly.

The main advantage of convolutional layers is that each value produced only depends on a small set of values from the previous layer. This increases compute-efficiency (less multiplications have to be performed), and allows the network to develop *shift invariance*, the ability to recognize the presence of small patterns in the input, regardless of where they are in the input, and how many of them there are (see figure A.4). This increases the generalization ability of the network. Convolutional layers play an important aspect in the domain of *representation learning* (Bengio *et al* 2013) where extracting high-level representations from the input is learned together with the classification, as opposed to manual feature engineering. Zeiler and Fergus (2013) provide a good explanation on how this concept can be visualized in the context of image classification.

Recurrent neural networks (RNN)

MLPs are also sometimes called *feed-forward neural networks*, as each layer only depends on the previous layer. There is no 'backwards' connection that goes from a layer i to layer $i-1$. Recurrent neural networks introduce these backward connections, with the long short-term memory (LSTM) being the most common case (Hochreiter and Schmidhuber 1997). LSTM networks are quite complicated to understand, and a precise definition is outside the scope of this article. The general intuition is that the backward connections

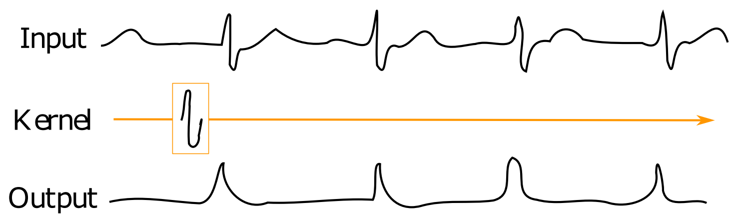


Figure A.4. A one-dimensional convolution can be used to detect patterns in a one-dimensional signal (here a heartbeat in an ECG). Every position at which the pattern matches is identified by the convolution, regardless of their location (*shift invariance*) and number of repetitions.

allow the network to 'remember' some information about an input, and use it to influence the prediction it will make when presented with the next input. Inputs presented to a neural network are therefore not seen as an unordered bag of independent inputs anymore (each of them receiving its independent prediction), but now form an ordered sequence. This allows the input x to be an element in a sequence of elements (for instance, one image in a video, or one instantaneous reading of 10 electrodes in a sequence of electrode readings). Then, one simply presents a sequence of inputs to the network, that at the end predicts an output that depends on all inputs that have been presented to it. LSTM networks are often used for sequence-to-sequence tasks, such as machine translation, signal processing, visual question answering and video processing.

In practice, neural networks can combine many layers of different kinds. It is not unusual to find neural networks that start with a few convolutional layers, to detect patterns independently of where they are in the input (such as edges in an image, or features in a 1D signal), then have one LSTM layer to be able to make sense of sequences of inputs, followed by a few feed-forward MLP-like layers to map what the LSTM layer learned to actual outputs. In the papers that we review in this article, great care is always given to explain and motivate the choice of neural architecture. Designing a neural network requires experience, as there is no systematic approach. We refer readers interested in knowing more than what we present here to books such as Goodfellow *et al* (2016) and Aggarwal (2018).

Deep learning

DL is a form of representation learning that introduces more than one level of abstraction to the learned features (Goodfellow *et al* 2016). This means that from the initially learned features, new features at a higher level of abstraction are extracted. If one considers the steps of the pipeline as a graph, then this graph can become much larger in the number of consecutive steps, or deeper, than with other techniques, and is therefore referred to as deep learning.

Most modern DL models are neural networks, however, other ML techniques can also qualify as DL. The most common type of DL neural networks at this time are the previously described CNNs (LeCun *et al* 1989). The subsequent application of convolutional layers results in a high-level representation of the input as stipulated by Goodfellow *et al* (2016). For example, in an object classification task, the network learns to extract primitive shapes from the raw input (a matrix of pixel values) in the first layer and then learns to extract objects from these primitive features in the next layer.

A DL method that is useful in biosignal decoding is transfer learning (Fahimi *et al* 2019, Kostas *et al* 2021). With transfer learning, a model is first trained on general data from which the model aims to learn the structure of the data. After pretraining, a fine-tuning phase will further train the model with data related to the specific problem that the model should solve. This approach is useful when faced with limited data availability for the target domain and specifically for biosignal decoding when calibrating a model for a new user. Transfer learning allows for the training of much larger models than if the model was trained directly on the task-specific dataset.

Data preprocessing

Until now, we considered an abstract representation of the data, with x the input and y the desired output. In practice, neural networks assume x and y to be multi-dimensional arrays of floating-point values. Going from raw data, as acquired by sensors (such as electrodes), to these arrays of floating-point values therefore requires application-specific code, usually being referred to as a *data preprocessing pipeline*.

Conceptually, given enough layers and neurons, and the proper architecture, a neural network can learn any mapping from inputs to outputs (Sonoda and Murata 2017) (they are universal function approximators). This means that any method of acquiring a signal, and representing it as floating-point values will eventually allow the network to make sense of the inputs, and learn something. However, a more careful design of the inputs allows to improve two important properties of the neural network: learning speed (important when the network is used in an adaptive system that learns as it is being used) and generalization power (the ability of making high-quality predictions for unseen inputs, even if training on a small amount of input-output pairs).

Designing the input, also called *feature engineering*, is highly domain-specific. In the signal processing literature, especially in settings that consider EEG data, the following preprocessing steps are commonly used for feature engineering:

(a) *Signal filtering*: applied on the signals, filters remove frequencies of the signals, to only keep those of interest. This is a form of noise removal, in which the expert designer knows that some frequencies never convey information and can only be noise. There are many different types of filters, which fall outside of the scope of this publication. For more information on filters and digital signal processing, interested readers are referred to (Orfanidis 1996).
(b) *Windowing*: when specific events in a signal (such as a spike or pattern) matters more than the overall shape of the complete signal, windowing allows to split a signal into fixed-length, usually overlapping, sub-sequences. Having the network focus on small sub-sequences allows it to be faster (less compute intensive, as less data is being processed), and generalize better, as a small number of easily-recognizable patterns (on which the network focuses) can appear in various positions in longer signals (that the network does not have to bother with). Several windowing methods exist, and are reviewed by Podder *et al* (2014). Jeong *et al* (2020) and Nguyen and Chung (2019) use Hamming windowing.
(c) *Feature extraction*: this final step is highly variable and depends on the exact context (sleep staging, Motor Imagery detection, epilepsy seizure detection, etc) in which the signal should be decoded. In general, DL models have been shown to perform better when the input is the raw (preprocessed) signal that is still represented as time-series of samples for each signal channel. One of the most commonly used feature extraction methods is the Fourier transform, which allows to decompose a temporal signal (a sequence of signal readings over time) to a sum of sinuses of various frequencies. The Fourier transform transforms data from the time domain to the frequency domain. This transform is loss-less and invertible, which means that it does not destroy information. It allows the neural network to more easily focus on the existence of a particular frequency in a signal, instead of having to make sense of the entire (time-domain) signal.

Other (more minor) preprocessing steps exist, which are not presented in this section. For a detailed review of possible feature extraction methods, we refer interested readers to Rashid *et al* (2020).

Improving generalization

We have introduced how neural networks work, how to train them, and how to design data acquisition pipelines that allow for efficient learning. We now focus on methods that, given a neural network, allow to increase its generalization power. Generalization is particularly important in the engineering and medical fields, where data is costly to acquire, as it allows a neural network to produce better predictions in production (on unseen data) with fewer training data points.

With neural networks, the main avenue to increase generalization is to decrease *over-fitting*. Over-fitting happens when a neural network remembers exactly what training input should learn which training output, without having actually made sense of the data. The network achieves a training loss close to 0, but produces garbage output on the testing set. It is like a small child that learns how to read words, and remembers that card number 7 is pronounced 'cat', without actually looking at the word written on the card, or being able to read at all.

Batch normalization (Ioffe and Szegedy 2015) considers the input of every layer in a neural network, and normalizes it so that, in expectation, the inputs of every layer has a zero mean and a unit variance. Intuitively, this normalization prevents ludicrously large or small values from appearing inside the network, which makes it 'behave better' or 'be smoother' (so, easier to train, and better at generalization). The actual mathematical way in which batch normalization works is however still unknown, with recent papers providing the first insights (Santurkar *et al* 2018).

Dropout (Srivastava *et al* 2014) does not modify the values that flow through a neural network, but instead randomly disables neurons every time the network is evaluated during training. The main motivation behind Dropout is to avoid one particular neuron in the network to learn how to compensate (and thus cancel out) another particular neuron in the network. When neurons are constantly randomly disabled and re-enabled, they all have to learn independently from each other. More mathematically, Dropout leads to a neural network that is made of a different set of neurons every time it is evaluated. This leads to a large ensemble of 'sub-networks', all trained on different datapoints. Ensembles of function approximators such as this are known to help with generalization (Dietterich 2000).

Both batch normalization (Tayeb *et al* 2019, Tam *et al* 2020) and Dropout (Gautam *et al* 2020, Tortora *et al* 2020a) are often used in biosignal decoding papers, sometimes both at the same time. Other normalization techniques are possible, such as L1-normalization or clipping the gradients (Zhang *et al* 2019a), but they have been superseded by Batch Normalization and Dropout.

ORCID iDs

Arnau Dillen https://orcid.org/0000-0001-5840-8920
Denis Steckelmacher https://orcid.org/0000-0003-1521-8494
Kyriakos Efthymiadis https://orcid.org/0000-0002-2235-5115
Kevin Langlois https://orcid.org/0000-0002-7991-0976
Albert De Beir https://orcid.org/0000-0002-8570-6613
Uros Marusic https://orcid.org/0000-0002-7420-2137
Bram Vanderborght https://orcid.org/0000-0003-4881-9341
Ann Nowé https://orcid.org/0000-0001-6346-4564
Romain Meeusen https://orcid.org/0000-0001-7553-9957
Fakhreddine Ghaffari https://orcid.org/0000-0002-0928-7963
Olivier Romain https://orcid.org/0000-0002-2172-1865
Kevin De Pauw https://orcid.org/0000-0002-6901-7199

References

Aggarwal C C 2018 *Neural Networks and Deep Learning: A Textbook* (Berlin: Springer)(https://doi.org/10.1007/978-3-319-94463-0)
Al-Saegh A, Dawwd S A and Abdul-Jabbar J M 2021 Deep learning for motor imagery EEG-based classification: a review *Biomed. Signal Process. Control* **63** 102172
Alex J S R, Haque A, Anand A, Gautam A and Venkatesan N 2020 A deep learning approach for robotic arm control using brain-computer interface *Int. J. Biol. Biomed. Eng.* **14** 8
Ameri A, Akhaee M A, Scheme E and Englehart K 2019 Regression convolutional neural network for improved simultaneous EMG control *J. Neural Eng.* **16** 036015
Aznan N K N, Connolly J D, Moubayed N A and Breckon T P 2019 Using variable natural environment brain–computer interface stimuli for real-time humanoid robot navigation *2019 Int. Conf. Robotics and Automation (ICRA)* (*Montreal, QC, Canada, 20–24 May 2019*) pp 4889–95
Aznan N K, Atapour-Abarghouei A, Bonner S, Connolly J D, Al Moubayed N and Breckon T P 2019 Simulating brain signals: creating synthetic EEG data via neural-based generative models for improved SSVEP classification *2019 Int. Joint Conf. Neural Networks (IJCNN)* (*Budapest, Hungary, 14–19 July 2019*) (https://doi.org/10.1109/IJCNN.2019.8852227) (IEEE)
Baevski A, Zhou H, Mohamed A and Auli M 2020 wav2vec 2.0: a framework for self-supervised learning of speech representations (arXiv:2006.11477 [cs, eess])
Banville H, Chehab O, Hyvärinen A, Engemann D A and Gramfort A 2020 Uncovering the structure of clinical EEG signals with self-supervised learning (arXiv:2007.16104)
Bengio Y, Courville A and Vincent P 2013 Representation learning: a review and new perspectives *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1798–828
Beraldo G, Castaman N, Bortoletto R, Pagello E, Millán J D R, Tonin L and Menegatti E 2018 ROS-health: an open-source framework for neurorobotics *2018 IEEE Int. Conf. Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)* (*Brisbane, QLD, Australia, 16–19 May 2018*) pp 174–9
Bevilacqua V, Tattoli G, Buongiorno D, Loconsole C, Leonardis D, Barsotti M, Frisoli A and Bergamasco M 2014 A novel BCI-SSVEP based approach for control of walking in virtual environment using a convolutional neural network *2014 Int. Joint Conf. Neural Networks (IJCNN)* (https://doi.org/10.1109/IJCNN.2014.6889955)
Bi L, Feleke A G and Guan C 2019 A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration *Biomed. Signal Process. Control* **51** 113–27
Bottou L 2012 Stochastic gradient descent tricks *Neural Networks: Tricks of the Trade* Lecture Notes in Computer Science (LNCS) 7700 (Berlin: Springer) pp 430–45
Buongiorno D, Cascarano G D, Brunetti A, De Feudis I and Bevilacqua V 2019 A survey on deep learning in electromyographic signal analysis *Intelligent Computing Methodologies* Lecture Notes in Computer Science 11645 (Berlin: Springer) pp 751–61
Caruana R and Niculescu-Mizil A 2006 An empirical comparison of supervised learning algorithms *Proc. 23rd Int. Conf. Machine Learning* (*Pittsburgh, Pennsylvania, USA 25–29 June 2006*) pp 161–8
Caspar E A, De Beir A, Lauwers G, Cleeremans A and Vanderborght B 2021 How using brain-machine interfaces influences the human sense of agency *PLoS One* **16** e0245191
Chiu C-F, Shih T K, Lin C-Y, Hui L, Utaminingrum F and Yang T-N 2019 Application of hand recognition system based on electromyography and gyroscope using deep learning *2019 12th Int. Conf. Ubi-Media Computing (Ubi-Media)* (*Bali, Indonesia, 5–8 August 2019*) (https://doi.org/10.1109/Ubi-Media.2019.00027)
Corley I A and Huang Y 2018 Deep EEG super-resolution: upsampling EEG spatial resolution with generative adversarial networks *2018 IEEE EMBS Int. Conf. Biomedical and Health Informatics, BHI 2018* (*Las Vegas, NV, USA, 4–7 March 2018*) (IEEE) pp 100–3
Cote-Allard U, Gagnon-Turcotte G, Phinyomark A, Glette K, Scheme E J, Laviolette F and Gosselin B 2020 Unsupervised domain adversarial self-calibration for electromyography-based gesture recognition *IEEE Access* **8** 177941–55
Cotton R J 2020 Smartphone control for people with tetraplegia by decoding wearable electromyography with an on-device convolutional neural network *2020 8th IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)* (*New York, NY, USA, 29 November–1 December 2020*) (https://doi.org/10.1109/BioRob49111.2020.9224367)
Craik A, He Y and Contreras-Vidal J L 2019 Deep learning for electroencephalogram (EEG) classification tasks: a review *J. Neural Eng.* **16** 031001
D'Amour A *et al* 2020 Underspecification presents challenges for credibility in modern machine learning (arXiv:2011.03395)
De Bock S *et al* 2022 Benchmarking occupational exoskeletons: an evidence mapping systematic review *Appl. Ergon.* **98** 103582
Dietterich T G 2000 Ensemble methods in machine learning *Int. Workshop on Multiple Classifier Systems* (*Cagliari, Italy, 21–23 June 2000*) (Springer) pp 1–15
Dix A, Finlay J, Abowd G D and Beale R 2003 *Human-Computer Interaction* (London: Pearson Education)
Elprama S A, Vannieuwenhuyze J T A, De Bock S, Vanderborght B, De Pauw K, Meeusen R and Jacobs A 2020 Social processes: what determines industrial workers' intention to use exoskeletons? *Hum. Factors* **62** 337–50

Fahimi F, Zhang Z, Goh W B, Lee T S, Ang K K and Guan C 2019 Inter-subject transfer learning with an end-to-end deep convolutional neural network for EEG-based BCI *J. Neural Eng.* **16** 026007

Falotico E *et al* 2017 Connecting artificial brains to robots in a comprehensive simulation framework: the neurorobotics platform *Front. Neurorobot.* **11** 2

Farina D, Lorrain T, Negro F and Jiang N 2010 High-density EMG E-textile systems for the control of active prostheses *2010 Annual Int. Conf. IEEE Engineering in Medicine and Biology* (*Buenos Aires, Argentina, 31 August–4 September 2010*) pp 3591–3

Gandhoke G S, Belykh E, Zhao X, Leblanc R and Preul M C 2019 Edwin Boldrey and Wilder Penfield's homunculus: a life given by Mrs. Cantlie (in and out of realism) *World Neurosurg.* **132** 377–88

Gautam A, Panwar M, Wankhede A, Arjunan S P, Naik G R, Acharyya A and Kumar D K 2020 LoCoMo-Net: a low-complex deep learning framework for sEMG-based hand movement recognition for prosthetic control *IEEE J. Trans. Eng. Health Med.* **8** 2100812

George J A, Brinton M R, Duncan C C, Hutchinson D T and Clark G A 2018 Improved training paradigms and motor-decode algorithms: results from intact individuals and a recent transradial amputee with prior complex regional pain syndrome *2018 40th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)* (*Honolulu, HI, USA, 18–21 July 2018*) (https://doi.org/10.1109/EMBC.2018.8513342)

George J A, Neibling A, Paskett M D and Clark G A 2020 Inexpensive surface electromyography sleeve with consistent electrode placement enables dexterous and stable prosthetic control through deep learning (arXiv:2003.00070)

Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)

Gross J 2019 Magnetoencephalography in cognitive neuroscience: a primer *Neuron* **104** 189–204

Han C-H, Kim E and Im C-H 2020 Development of a brain-computer interface toggle switch with low false-positive rate using respiration-modulated photoplethysmography *Sensors* **20** E348

Hanafusa M and Ishikawa J 2020 Human-adaptive impedance control using recurrent neural network for stability recovery in human-robot cooperation *2020 IEEE 16th Int. Workshop on Advanced Motion Control (AMC)* (*Kristiansand, Norway, 14–16 September 2020*) (https://doi.org/10.1109/AMC44022.2020.9244389)

He X, Zhao K and Chu X 2021 AutoML: a survey of the state-of-the-art *Knowl.-Based Syst.* **212** 106622

Hehenberger L, Kobler R J, Lopes-Dias C, Srisrisawang N, Tumfart P, Uroko J B, Torke P R and Müller-Putz G R 2021 Long-term mutual training for the CYBATHLON BCI race with a tetraplegic pilot: a case study on inter-session transfer and intra-session adaptation *Front. Hum. Neurosci.* **15** 70

Hernandez-Carmona D and Penaloza C I 2019 Deep learning based image reconstruction from brain data during object-grasping task with a BMI *2019 IEEE Int. Conf. Advanced Robotics and its Social Impacts (ARSO)* (*Beijing, China, 31 October–2 November 2019*) (https://doi.org/10.1109/ARSO46408.2019.8948782)

Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

Hu Y, Lin G, Yang C, Li Z and Su C-Y 2015 Manipulation and grasping control for a hand-eye robot system using sensory-motor fusion *2015 IEEE Int. Conf. Robotics and Biomimetics (ROBIO)* (*Zhuhai, China, 6–9 December 2015*) (https://doi.org/10.1109/ROBIO.2015.7418792)

Hunter J D 2007 Matplotlib: a 2D graphics environment *Comput. Sci. Eng.* **9** 90–95

Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *32ndInt. Conf. Machine Learning (ICML 2015)* (*Lille France, 6–11 July 2015*) vol 37 (PMLR) pp 448–56

Ishizuka K, Kobayashi N and Saito K 2020 High accuracy and short delay 1ch-SSVEP quadcopter-BMI using deep learning *J. Robot. Mechatronics* **32** 738–44

Jafarzadeh M, Hussey D C and Tadesse Y 2019 Deep learning approach to control of prosthetic hands with electromyography signals *2019 IEEE Int. Symp. on Measurement and Control in Robotics (ISMCR)* (*Houston, TX, USA, 19–21 September 2019*) (https://doi.org/10.1109/ISMCR47492.2019.8955725)

Jeong J-H, Shim K-H, Kim D-J and Lee S-W 2020 Brain-controlled robotic arm system based on multi-directional CNN-BiLSTM network using EEG signals *IEEE Trans. Neural Syst. Rehabil. Eng.* **28** 1226–38

Jing L and Tian Y 2019 Self-supervised visual feature learning with deep neural networks: a survey (arXiv:1902.06162)

Karácsony T, Hansen J P, Iversen H K and Puthusserypady S 2019 Brain computer interface for neuro-rehabilitation with deep learning classification and virtual reality feedback *Proc. 10th Augmented Human Int. Conf. 2019—AH2019* (*Reims France, 11–12 March 2019*) p 22

Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

Kluyver T *et al* 2016 *Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows* (Amsterdam: IOS Press) pp 87–90

Kobayashi N and Ishizuka K 2019 LSTM-based classification of multiflicker-SSVEP in single channel dry-EEG for low-power/high-accuracy quadcopter-BMI system *2019 IEEE Int. Conf. Systems, Man and Cybernetics (SMC)* (*Bari, Italy, 6–9 October 2019*) (https://doi.org/10.1109/SMC.2019.8914015)

Kostas D, Aroca-Ouellette S and Rudzicz F 2021 BENDR: using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data (arXiv:2101.12037)

Kotonya G 1998 *Requirements Engineering: Processes and Techniques* Worldwide Series in Computer Science 10 (New York: Wiley) p 294

Kuhner D *et al* 2019 A service assistant combining autonomous robotics, flexible goal formulation and deep-learning-based brain–computer interfacing *Robot. Auton. Syst.* **116** 98–113

Lai L, Suda N and Chandra V 2018 CMSIS-NN: efficient neural network kernels for arm cortex-M CPUs (arXiv:1801.06601)

LeCun Y *et al* 1995 Convolutional networks for images, speech and time series *The Handbook of Brain Theory and Neural Networks* **vol 3361** (Cambridge, MA: MIT Press) p 1252 (available at: www.iro.umontreal.ca/~lisa/pointeurs/handbook-convo.pdf)

LeCun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1989 Backpropagation applied to handwritten zip code recognition *Neural Comput.* **1** 541–51

Leeb R, Sagha H, Chavarriaga R and Millán J D R 2011 A hybrid brain-computer interface based on the fusion of electroencephalographic and electromyographic activities *J. Neural Eng.* **8** 25011–6

Li C, Ren J, Huang H, Wang B, Zhu Y and Hu H 2018 PCA and deep learning based myoelectric grasping control of a prosthetic hand *Biomed. Eng. Online* **17** 107

Li X, Huang Q, Zhu J, Sun W and She H 2017 A novel proportional and simultaneous control method for prosthetic hand *J. Mech. Med. Biol.* **17** 1750120

Liu H, Zhang Z, Qian J, Wang W and Kang K 2019 Hand gesture recognition based on deep neural network and sEMG signal *2019 IEEE Int. Conf. Robotics and Biomimetics (ROBIO)* (*Dali, China, 6–8 December 2019*) (https://doi.org/10.1109/ROBIO49542.2019.8961445)

Loopez-Larraz E, Birbaumer N and Ramos-Murguialday A 2018 A hybrid EEG-EMG BMI improves the detection of movement intention in cortical stroke patients with complete hand paralysis *2018 40th Annual Int. Conf. IEEE*

Engineering in Medicine and Biology Society (EMBC) (*Honolulu, HI, USA, 18–21 July 2018*) pp 2000–3
Lu K, Grover A, Abbeel P and Mordatch I 2021 Pretrained transformers as universal computation engines (arXiv:2103.05247 [cs])
Lu W, Wei Y, Yuan J, Deng Y and Song A 2020 Tractor assistant driving control method based on EEG combined with RNN-TL deep learning algorithm *IEEE Access* **8** 163269–79
Makhrov S S and Denisova E N 2018 On-board brain-computer interface based on the recognition of patterns of brain activity through a convolutional neural network *2018 Systems of Signals Generating and Processing in the Field of on Board Communications* (*Moscow, Russia, 14–15 March 2018*) (https://doi.org/10.1109/SOSG.2018.8350620)
McFarland D J and Wolpaw J R 2018 Brain–computer interface use is a skill that user and system acquire together *PLoS Biol.* **16** e2006719
Merlin Praveena D, Angelin Sarah D and Thomas George S 2020 Deep learning techniques for EEG signal applications–a review *IETE J. Res.* 1–8
Moher D *et al* 2016 Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement *Rev. Esp. Nutr. Hum. Diet.* **20** 148–60
Mourad A, Wissa V and Sherif H E 2020 Vehicle control by using brain signals driving system for disabled people *2020 6th Int. Conf. Mechatronics and Robotics Engineering (ICMRE)* (*Barcelona, Spain, 12–15 February 2020*) (https://doi.org/10.1109/ICMRE49073.2020.9064969)
Naseer N and Hong K-S 2015 fNIRS-based brain-computer interfaces: a review *Front. Hum. Neurosci.* **9** 3
Nguyen T-H and Chung W-Y 2018 A single-channel SSVEP-based BCI speller using deep learning *IEEE Access* **7** 1752–63
Nguyen T-H and Chung W-Y 2019 Detection of driver braking intention using EEG signals during simulated driving *Sensors* **19** 2863
Nielsen J 1994 *Usability Engineering* (San Mateo, CA: Morgan Kaufmann Publishers) p 362
Nilsson N J 2009 *The Quest for Artificial Intelligence* (Cambridge: Cambridge University Press)
Norman S L, Maresca D, Christopoulos V N, Griggs W S, Demene C, Tanter M, Shapiro M G and Andersen R A 2021 Single-trial decoding of movement intentions using functional ultrasound neuroimaging *Neuron* **109** 1554–66.e4
Novak D and Riener R 2015 A survey of sensor fusion methods in wearable robotics *Robot. Auton. Syst.* **73** 155–70
Orfanidis S J 1996 *Introduction to Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall) p 798
Orfanidis S J 1996 *Introduction to Signal Processing* (London: Pearson Education)
Orlando M F, Akolkar H, Dutta A, Saxena A and Behera L 2010 Optimal design and control of a hand exoskeleton *2010 IEEE Conf. Robotics, Automation and Mechatronics* (*Singapore, Singapore, 28–30 June 2010*) (https://doi.org/10.1109/RAMECH.2010.5513211)
Penfield W and Boldrey E 1937 Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation *Brain* **60** 389–443
Perdikis S, Tonin L, Saeedi S, Schneider C and Millán J D R 2018 The Cybathlon BCI race: successful longitudinal mutual learning with two tetraplegic users *PLoS Biol.* **16** e2003787
Podder P, Khan T Z, Khan M H and Muktadir Rahman M 2014 Comparative performance analysis of hamming, hanning and blackman window *Int. J. Comput. Appl.* **96** 1–7
Ramadan R A and Vasilakos A V 2017 Brain computer interface: control signals review *Neurocomputing* **223** 26–44
Ramakrishnan J, Mavaluru D, Sakthivel R S, Alqahtani A S, Mubarakali A and Retnadhas M 2020 Brain-computer interface for amyotrophic lateral sclerosis patients using deep learning network *Neural Comput. Appl.* Special Issue on New Trends in Brain Computer Interface 1–15
Ramchoun H, Idrissi M A J, Ghanou Y and Ettaouil M 2016 Multilayer perceptron: architecture optimization and training *Int. J. Interact. Multimedia Artif. Intell.* **4** 26–30
Rashid M, Sulaiman N, Abdul Majeed A P P, Musa R M, Nasir A F A, Bari B S and Khatun S 2020 Current status, challenges and possible solutions of EEG-based brain-computer interface: a comprehensive review *Front. Neurorobot.* **14** 25
Rasmussen C E *et al* 1999 The infinite Gaussian mixture model *12th Int. Conf. Neural Information Processing Systems* (*Denver, CO, November 29–December 4, 1999*) *Advances in Neural Information Processing Systems* vol 12 pp 554–60
Reback J *et al* 2020 pandas-dev/pandas: pandas 1.0.3 *Zenodo* (available at: https://zenodo.org/record/3715232)
Rechy-Ramirez E J and Hu H 2015 Bio-signal based control in assistive robots: a survey *Digit. Commun. Netw.* **1** 85–101
Redrovan D V and Kim D 2018 Hand gestures recognition using machine learning for control of multiple quadrotors *2018 IEEE Sensors Applications Symp. (SAS)* (*Seoul, Korea (South), 12–14 March 2018*) (https://doi.org/10.1109/SAS.2018.8336782)
Ren J-L, Chien Y-H, Chia E-Y, Fu L-C and Lai J-S 2019 Deep learning based motion prediction for exoskeleton robot control in upper limb rehabilitation *2019 Int. Conf. Robotics and Automation (ICRA)* (*Montreal, QC, Canada, 20–24 May 2019*) (https://doi.org/10.1109/ICRA.2019.8794187)
Renard Y, Lotte F, Gibert G, Congedo M, Maby E, Delannoy V, Bertrand O and Lécuyer A 2010 OpenViBE: an open-source software platform to design, test and use brain–computer interfaces in real and virtual environments *Presence* **19** 35–53
Rim B, Sung N-J, Min S and Hong M 2020 Deep learning in physiological signal data: a survey *Sensors* **20** 969
Roc A, Pillette L, Mladenovic J, Benaroch C and N'Kaoua B 2021 A review of user training methods in brain computer interfaces based on mental tasks *J. Neural Eng.* **18** 011002
Roy Y, Banville H, Albuquerque I, Gramfort A, Falk T H and Faubert J 2019 Deep learning-based electroencephalography analysis: a systematic review *J. Neural Eng.* **16** 051001
Rumelhart D E, Hinton G E and Williams R J 1985 Learning internal representations by error propagation *Technical Report* (San Diego, CA: Institute for Cognitive Science, California University)
Sakaki K *et al* 2020 Evaluation of neural activity by magnetospinography with 3D sensors *Clin. Neurophysiol.* **131** 1252–66
Santurkar S, Tsipras D, Ilyas A and Madry A 2018 How does batch normalization help optimization? *Proc. 32nd Int. Conf. Neural Information Processing Systems* (*Montréal, Canada, 3–8 December 2018*) pp 2488–98
Schirrmeister R T, Springenberg J T, Fiederer L D J, Glasstetter M, Eggensperger K, Tangermann M, Hutter F, Burgard W and Ball T 2017 Deep learning with convolutional neural networks for EEG decoding and visualization *Front. Neurorobot.* **38** 5391–420
Shim K-H, Jeong J-H, Kwon B-H, Lee B-H and Lee S-W 2019 Assistive robotic arm control based on brain-machine interface with vision guidance using convolution neural network *2019 IEEE Int. Conf. Systems, Man and Cybernetics (SMC)* (*Bari, Italy, 6–9 October 2019*) (https://doi.org/10.1109/SMC.2019.8914058)
Shima K and Tsuji T 2010 Classification of combined motions in human joints through learning of individual motions based on muscle synergy theory *2010 IEEE/SICE Int. Symp. on System Integration* (*Sendai, Japan, 21–22 December 2010*) (https://doi.org/10.1109/SII.2010.5708346)
Simão M, Mendes N, Gibaru O and Neto P 2019 A review on electromyography decoding and pattern recognition for human-machine interaction *IEEE Access* **7** 39564–82
Sitaram R, Caria A, Veit R, Gaber T, Rota G, Kuebler A and Birbaumer N 2007 fMRI brain-computer interface: a tool for neuroscientific research and treatment *Comput. Intell. Neurosci.* **2007** 25487
Song W, Wang W and Jiang F 2020 A control system of SEMG signal based on deep learning *Int. J. Circuits Syst. Signal Process.* **14** 386–91

Sonoda S and Murata N 2017 Neural network with unbounded activation functions is universal approximator *Appl. Comput. Harmon. Anal.* **43** 233–68

Srinivas A, Lin T-Y, Parmar N, Shlens J, Abbeel P and Vaswani A 2021 Bottleneck transformers for visual recognition (arXiv:2101.11605 [cs])

Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58

Stroh A and Desai J 2019 Human-centered deep learning neural network trained myoelectric controller for a powered wheelchair *2019 IEEE Int. Symp. on Measurement and Control in Robotics (ISMCR)* (*Houston, TX, USA, 19–21 September 2019*) (https://doi.org/10.1109/ISMCR47492.2019.8955734)

Tam S, Boukadoum M, Campeau-Lecours A and Gosselin B 2020 A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning *IEEE Trans. Biomed. Circuits Syst.* **14** 232–43

Tayeb Z, Fedjaev J, Ghaboosi N, Richter C, Everding L, Qu X, Wu Y, Cheng G and Conradt J 2019 Validating deep neural networks for online decoding of motor imagery movements from EEG signals *Sensors* **19** 210

Teban T-A, Precup R-E, Lunca E-C, Albu A, Bojan-Dragos C-A and Petriu E M 2018 Recurrent neural network models for myoelectric based control of a prosthetic hand *2018 22nd Int. Conf. System Theory, Control and Computing (ICSTCC)* (*Sinaia, Romania, 104–12 October 2018*) (https://doi.org/10.1109/ICSTCC.2018.8540720)

Thomas J, Maszczyk T, Sinha N, Kluge T and Dauwels J 2017 Deep learning-based classification for brain-computer interfaces *2017 IEEE Int. Conf. Systems, Man and Cybernetics (SMC)* (*Banff, AB, Canada, 54–8 October 2017*) (IEEE) pp 234–9

Tonin L, Beraldo G, Tortora S, Tagliapietra L, Millán J D R and Menegatti E 2019 ROS-Neuro: a common middleware for BMI and robotics. The acquisition and recorder packages *2019 IEEE Int. Conf. Systems, Man and Cybernetics (SMC)* (*Bari, Italy, 6–9 October 2019*) pp 2767–72

Tortora S, Ghidoni S, Chisari C, Micera S and Artoni F 2020a Deep learning-based BCI for gait decoding from EEG with LSTM recurrent neural network *J. Neural Eng.* **17** 046011

Tortora S, Tonin L, Chisari C, Micera S, Menegatti E and Artoni F 2020b Hybrid human-machine interface for gait decoding through Bayesian fusion of EEG and EMG classifiers *Front. Neurorobot.* **14** 89

Valenti A, Barsotti M, Bacciu D and Ascari L 2021 A deep classifier for upper-limbs motor anticipation tasks in an online BCI setting *Bioengineering* **8** 21

Valenti A, Barsotti M, Brondi R, Bacciu D and Ascari L 2020 ROS-neuro integration of deep convolutional autoencoders for EEG signal compression in real-time BCIs (arXiv:2008.13485 [cs, eess, stat])

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* (*Long Beach, CA, United States 4–9 December 2017*) vol 30 pp 5999–6009

Wan Y, Han Z, Zhong J and Chen G 2018 Pattern recognition and bionic manipulator driving by surface electromyography signals using convolutional neural network *Int. J. Adv. Robot. Syst.* **15** 172988141880213

Wang F, Zhong S H, Peng J, Jiang J and Liu Y 2018 Data augmentation for EEG-based emotion recognition with deep convolutional neural networks *Multimedia Modeling* Lecture Notes in Computer Science 10705 (Berlin: Springer) pp 82–93

Wang X, Hersche M, Tomekce B, Kaya B, Magno M and Benini L 2020 An accurate EEGNet-based motor-imagery brain-computer interface for low-power edge computing *2020 IEEE Int. Symp. on Medical Measurements and Applications (MeMeA)* (*Bari, Italy, 1 June–1 July 2020*) (https://doi.org/10.1109/MeMeA49120.2020.9137134)

Waskom M L 2021 seaborn: statistical data visualization *J. Open Source Softw.* **6** 3021

Xiang K, Guo S and Song Z 2012 An electromyography-driven central pattern generator model for robotic control application *2012 IEEE Int. Conf. Robotics and Biomimetics (ROBIO)* (*Guangzhou, China, 11–14 December 2012*) (https://doi.org/10.1109/ROBIO.2012.6491270)

Yin L *et al* 2021 A self-sustainable wearable multi-modular E-textile bioenergy microgrid system *Nat. Commun.* **12** 1542

Yohanandan S A, Kiral-Kornek I, Tang J, Mshford B S, Asif U and Harrer S 2018 A robust low-cost EEG motor imagery-based brain-computer interface *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, EMBS* (*Honolulu, HI, USA, 18–21 July 2018*) (IEEE) pp 5089–92

Zanghieri M, Benatti S, Burrello A, Kartsch V, Conti F and Benini L 2020 Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor *IEEE Trans. Biomed. Circuits Syst.* **14** 244–56

Zeiler M D and Fergus R 2013 Visualizing and understanding convolutional networks (arXiv:1311.2901)

Zgallai W, Brown J T, Ibrahim A, Mahmood F, Mohammad K, Khalfan M, Mohammed M, Salem M and Hamood N 2019 Deep learning AI application to an EEG driven BCI smart wheelchair *2019 Advances in Science and Engineering Technology Int. Conf. (ASET)* (*Dubai, United Arab Emirates, 26 March–10 April 2019*) (https://doi.org/10.1109/ICASET.2019.8714373)

Zhang D, Yao L, Chen K, Wang S, Chang X and Liu Y 2020a Making sense of spatio-temporal preserving representations for EEG-Based human intention recognition *IEEE Trans. Cybern.* **50** 3033–44

Zhang J, He T, Sra S and Jadbabaie A 2019a Why gradient clipping accelerates training: a theoretical justification for adaptivity (arXiv:1905.11881)

Zhang X, Yao L, Huang C, Sheng Q Z and Wang X 2017 Intent recognition in smart living through deep recurrent neural networks *Neural Information Processing* Lecture Notes in Computer Science 10635 (Berlin: Springer) pp 748–58

Zhang X, Yao L, Wang X, Monaghan J, McAlpine D and Zhang Y 2020b A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers *J. Neural Eng.* **18** 031002

Zhang X, Yao L, Zhang S, Kanhere S, Sheng M and Liu Y 2019b Internet of things meets brain-computer interface: a unified deep learning framework for enabling human-thing cognitive interactivity *IEEE Internet Things J.* **6** 2084–92

Zhou X, Wang F, Wang J, Wang Y, Yan J and Zhou G 2019 Deep learning based gesture recognition and its application in interactive control of intelligent wheelchair *Intelligent Robotics and Applications* (Cham: Springer International Publishing) pp 547–57

Zhuang J, Geng K and Yin G 2021 Ensemble learning based brain–computer interface system for ground vehicle control *IEEE Trans. Syst. Man Cybern.* A **51** 5392–404

Zied T, Juri F, Nejla G, Christoph R, Lukas E, Xingwei Q, Yingyu W, Gordon C and Jörg C 2019 Validating deep neural networks for online decoding of motor imagery movements from EEG signals *Sensors* **19** 210