Investigating fMRI neurofeedback score prediction from EEG signals: genetic algorithm applied to hyperparameter selection

Caroline Pinte, Claire Cury, Pierre Maurel

Investigating fMRI neurofeedback score prediction from EEG signals: genetic algorithm applied to hyperparameter selection

Caroline **Pinte** [1], Claire Cury [1], Pierre Maurel [1],

1 Univ Rennes, Inria, CNRS, Inserm, IRISA UMR 6074, Empenn ERL U 1228, F-35000 Rennes, France

Abstract

Simultaneous electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) acquisitions can provide more effective neurofeedback (NF) training due to their complementary temporal and spatial precision. However, MRI is expensive and can be draining for participants. Therefore, our goal is to reduce the reliance on MRI by developing a model that can predict fMRI NF scores from EEG signals alone, potentially eliminating the need for MRI. Yet, arbitrarily proposing a model architecture for such complex problems is challenging. So, in this study, we used a genetic algorithm to search for neural network architecture hyperparameters, specifically applied here to convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. The resulting architectures provided fMRI NF score predictions that, when combined with EEG NF scores, significantly matched the true bi-modal EEG-fMRI NF scores more closely than the EEG NF scores alone. This approach demonstrates the potential for enriching the EEG modality in a unimodal neurofeedback framework, thereby reducing the need for MRI. However, the predictions still lack precision. Therefore, this work thoroughly investigates the potential for enriching the EEG modality in a unimodal neurofeedback framework. Our code and models are available at `https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals`.

Keywords

Time Series Regression, 1D CNN, LSTM, Genetic Algorithm, EEG, fMRI, Neurofeedback

Article informations

1. Introduction

Neurofeedback (NF) is a non-invasive therapeutic technique that uses real-time monitoring of brain patterns to provide individuals with NF scores, feeding back information about their brain activity (Sitaram et al., 2017). The primary goal of neurofeedback in clinical settings is to enable individuals to self-regulate their brain function, leading to improvements in cognitive, emotional and behavioral functioning across various health conditions, such as motor recovery after stroke (Renton et al., 2017; Wang et al., 2018). Acquisitions are typically made through non-invasive modalities such as electroencephalography (EEG) or functional magnetic resonance imaging (fMRI).

EEG provides a direct measurement of real-time electrical potential changes in the brain using electrodes placed on the scalp. This equipment is known for its portability and affordability. While it offers excellent temporal resolution, operating within the millisecond range, its spatial resolution is limited to the centimeter range (Boly et al., 2016), notably due to the ill-posed inverse problem of source localization.

fMRI indirectly estimates brain activity by measuring variations in the blood oxygenation level-dependent (BOLD) signal, reflecting neurovascular activity. This activity generally occurs a few seconds after neural activation measured by EEG and is referred to as the hemodynamic response. In contrast to EEG, fMRI is non-portable and much more costly. While it offers better spatial resolution than EEG, in the millimeter range, its temporal resolution is inferior, typically in the second range.

The complementary nature of these resolutions quickly motivated the combination of EEG and fMRI. Despite the ongoing challenges in processing and integration, which persist to this day, simultaneous EEG-fMRI acquisitions offer multi-modal non-invasive measurements of brain activity applicable in many contexts (Abreu et al., 2018), including neurofeedback (Zotev et al., 2014; Mano et al., 2017; Ciccarelli et al., 2023). Several studies have investigated the relationship between EEG signals and BOLD activity (de Munck et al., 2007; Scheeringa et al., 2011; Magri et al., 2012; Portnova et al., 2018). However, the

correlations identified do not consistently establish a link between the two modalities, as the results are highly dependent on the task, brain region, and frequency bands considered.

In the context of motor imagery neurofeedback, the use of simultaneous EEG-fMRI acquisitions has resulted in higher and more specific activation compared to EEG neurofeedback alone, as demonstrated in Perronnet et al. (2017) and Cury et al. (2020a). However, as the use of MRI is very costly and burdensome for the participant, we aim to minimize its usage while maintaining the quality of the sessions. Thus, our goal is to develop a model capable of predicting fMRI NF scores from EEG signals alone, in order to enhance EEG NF scores during unimodal EEG neurofeedback sessions. This objective has already been investigated in a previous study of our lab (Cury et al., 2020b), where a sparse regression model was proposed to predict fMRI NF scores from EEG signals during motor imagery tasks, showing encouraging results in the development of individualized models for each participant. Now, to take a step towards real application in clinical settings, we seek to create a single global model, applicable to all participants. As our problem falls into the category of time series regression, we have at our disposal two widely used classes of machine learning models for such tasks: recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

RNNs (Hopfield, 1982) take into account past inputs through a feedback loop that incorporates both the current input and information from the previous input. This so-called short-term sequential memory is stored in the network's hidden state, which is updated with each new input, enabling it to capture the context and patterns of the sequence based on prior inputs. However, the original architecture had shortcomings, notably the vanishing gradient problem (Bengio et al., 1994). To overcome this issue and better retain long-term dependencies in the data, the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) architecture was introduced as an improved version of the RNN. RNN-LSTMs introduce a cell state with gates to retain important information over long sequences. However, despite their benefits, LSTMs are notoriously challenging to deploy effectively. The difficulty arises from the complexity of the architecture, leading to a greater risk of over-fitting, the requirement for a large training dataset, and the necessity of making numerous decisions concerning architecture and training hyperparameters (Greff et al., 2016). Consequently, performance can vary significantly based on these factors (Parmezan et al., 2019).

CNNs (LeCun et al., 1989) are a class of neural networks mainly used in computer vision due to their ability to find patterns and spatial hierarchies of features within images. The core principle of CNNs resides in convolution layers, which employ sets of learnable filters that slide across the input image and conduct convolution operations to extract spatial features. The first layers of the network detect basic features such as horizontal and vertical edges, while subsequent layers extract increasingly complex features such as objects or faces. Additionally, CNNs incorporate pooling layers to reduce the spatial dimensions of the feature maps produced, and fully connected layers to integrate the high-level features learned to perform classification or regression tasks. In the context of time series analysis, a specialized variant known as the one-dimensional convolutional neural network, or 1D CNN, can be employed. This type of CNN is designed for processing sequential data, such as time series datasets. It is worth noting that, contrary to LSTMs, CNNs are not explicitly designed to capture long-term dependencies. However, due to their simpler architecture, they typically offer quicker training times, often require less data for training, and can exhibit more stable performance (Zhang et al., 2015; Cura et al., 2020; Marinho et al., 2023).

When using neural networks, one of the most important issues is the design of the architecture. Most of them are designed manually, sometimes thanks to prior expertise, but often without any concrete justification for the choices made other than empirical exploration. In addition, in cases where users do not have sufficient expertise, for example on a problem like ours that has not yet been widely investigated, it becomes very difficult to achieve a high-performance network architecture as well as to justify the choices made by manually designing a network. One way of addressing this challenge is to adopt a well-known method, called the genetic algorithm, and apply it to the particular case of neural network architecture search. The genetic algorithm was introduced in 1975 by John Holland and his collaborators, gaining popularity in the 1990s, as indicated by the reissued work Holland (1992). It is an evolutionary algorithm inspired by the process of natural selection and genetic evolution and is used to solve all kinds of optimization problems. The general idea is to borrow from natural selection the concepts of reproduction, crossover, and mutation to iteratively evolve a population of potential solutions toward better solutions over the course of successive generations.

One of the many optimization problems that benefit from this approach is the search for neural network hyperparameters. Hyperparameters are values set by the user prior to the model training, distinct from parameters known as weights that are updated during the model learning pro-

cess. Examples of hyperparameters include those controlling the learning process, such as the learning rate and batch size, or those defining the model architecture, such as the number of hidden layers and the dropout rate. The idea of applying the genetic algorithm approach to automatically set hyperparameter values was quickly investigated (Miller et al., 1989), and then specialized for various types of networks and tasks, such as regression with feedforward neural networks (Benardos and Vosniakos, 2007), image classification with CNNs (Xie and Yuille, 2017; Sun et al., 2020), and natural language processing task with LSTMs (Gorgolis et al., 2019). Regarding the time series regression task, which is our focus in this work, genetic algorithms have occasionally been applied to some types of neural networks, including time-delay neural networks (TDNNs) (Hansen et al., 1999), LSTMs (Bouktif et al., 2018; Erden, 2023), and 1D CNNs (Chung and Shin, 2020). However, while all these works use the general principle of genetic algorithms, they tailor their approaches to the specific network type chosen, and consequently, the hyperparameters to be optimized. Often, these choices are influenced by the application context. Given that our problem is still relatively unexplored, we had no preconceived ideas about the best model type to use, necessitating a more general approach. Therefore, we decided to implement a genetic architecture search algorithm for our time series regression task without specializing in any single network type, allowing us the flexibility to explore several possibilities, such as LSTMs and CNNs.

In this work, we introduce a genetic algorithm approach for searching architecture hyperparameters, designed to be applicable to various types of neural networks. We chose to apply it to LSTMs and CNNs in order to study the prediction of fMRI NF scores from EEG signals alone, using a global model approach, as opposed to subject-specific models. The goal of this approach is to reduce the reliance on the costly fMRI modality in a bi-modal neurofeedback context.

2. Materials

The pseudonymized data are available in BIDS format on the OpenNeuro platform: `https://openneuro.org/datasets/ds002338`. OpenNeuro is an open science database dedicated to storing datasets from human brain imaging research studies, providing a free and open platform for sharing data. This dataset consists of simultaneous EEG-fMRI acquisitions performed during a motor imagery neurofeedback task. It is described in Lioi et al. (2020) as the XP2 protocol and received approval by the Institutional Review Board. This section offers information about the participants, the experimental protocol, equipment, offline processing steps, and concludes with specifics regarding the computation of the NF scores used in our study.

2.1 Participants and protocol

We used data collected from 15 healthy subjects included in the XP2 protocol, described below. The original study (Perronnet et al., 2018) involved 17 subjects, but two individuals (identified as sub-xp202 and sub-xp203) were excluded from our analysis due to a lack of BOLD data. All participants were right-handed and had never taken part in a neurofeedback session. Each participant provided signed informed consent, including consent for the publication of their anonymized data.

This protocol comprises a single session, featuring three motor imagery (MI) neurofeedback runs. Initially, a MIpre run, which is a run where the subject engages in motor imagery without receiving any feedback, was used to identify the region of interest (ROI) for fMRI processing. Then, three neurofeedback runs were performed with a one-minute break between each run. A single run consists of eight blocks alternating between 20 seconds of rest with eyes open and 20 seconds of motor imagery task involving the right hand, with visual feedback. The session concluded with a MIpost block without feedback to evaluate the participant's performance. Finally, within this protocol, seven subjects received unidimensional (1D) feedback, while the remaining eight were provided with bidimensional (2D) feedback. While Cury et al. (2020a) showed that participant behavior differs between 1D and 2D feedback, we included all 15 subjects in our analysis, following the approach in Cury et al. (2020b). This was done to ensure an adequate amount of data given the already limited subject pool, under the assumption that this inclusion would not significantly impact the results.

2.2 Equipment

Data were obtained through a hybrid EEG-fMRI neurofeedback setup located at the Neurinfo platform (Rennes University Hospital, France), with detailed specifications provided in Mano et al. (2017). This platform facilitates EEG-fMRI acquisition, online processing, EEG-fMRI NF scores computation over time, and synchronisation before sending the visual feedback.

EEG data was recorded using a 64-channel extended international 10–20 EEG system solution from Brain Products (Brain Products GmbH, Gilching, Germany), which is MR-compatible. The signal was sampled at a rate of 5 kHz and a resolution of 0.5 µV, with FCz used as the reference electrode and AFz as the ground electrode.

fMRI acquisitions were conducted using a 3T Verio MRI running VB17 (Siemens Healthineers, Erlangen, Germany)

and equipped with a 12-channel receiver head coil. The acquisitions were carried out using echo-planar imaging (EPI) and covered the upper half of the brain with the following parameters: TR = 1s, TE = 23ms, resolution: 2×2×4 mm^3, number of 4-mm slices: 16, no slice gap.

2.3 Offline processing

Detailed information about preprocessing can be found in Lioi et al. (2020), leading to a signal sampled at 200 Hz for the EEG modality. As for fMRI acquisitions, MIpre runs mentioned earlier were used in each session to conduct a first-level general linear model (GLM) analysis. The resulting activation maps, voxel-wise family-wise error corrected at $p < 0.05$, were used to define two regions of interest (ROIs), each measuring $9 \times 9 \times 3$ voxels, centered around the maximum activation in the primary motor area (M1) and the supplementary motor area (SMA), respectively.

2.4 NF scores computation

The EEG NF scores were computed as a measure of event-related desynchronization (ERD), following the formula:

$$NF_{EEG}(t) = \frac{BP_{C3}(rest) - BP_{C3}(t)}{BP_{C3}(rest)} \quad (1)$$

where $BP_{C3}(t)$ represents the power in the 8–30 Hz frequency band of a Laplacian around C3 at time t and $BP_{C3}(rest)$ denotes the average power in the 8–30 Hz frequency band over the resting block preceding the neurofeedback training. $NF_{EEG}(t)$ quantifies the desynchronization occurring during motor imagery in relation to the baseline at rest. The EEG NF scores were converted into visual feedback every 250 ms, resulting in 1280 NF scores per run.

The fMRI NF scores were then calculated according to the following formula:

$$NF_{fMRI}(t) = \frac{B_{ROI}(t)}{B_{ROI}(rest)} - \frac{B_{BG}(t)}{B_{BG}(rest)} \quad (2)$$

where $B_{ROI}(t)$ represents the fMRI signal in the ROI (M1 area) selected during the calibration step at time t, divided by the corresponding signal averaged across the last 6 seconds of the preceding rest block. $B_{BG}(t)$ denotes the BOLD signal in a background lower slice, included to normalize by global BOLD signal changes. The fMRI NF scores were converted into visual feedback every second, resulting in 320 NF scores per run.

3. Methods

Our approach focuses on predicting fMRI neurofeedback scores from EEG signals. As illustrated in Figure 1, we operate in the context of bi-modal EEG-fMRI neurofeedback sessions. The long-term objective is to deploy this method during future neurofeedback sessions, where the model provides fMRI NF predictions without MRI acquisitions, to reduce its associated costs. The purpose of this section is to explain the construction of such a model. Firstly, we will describe the creation of the supervised learning dataset. Then, we will detail the search for the model architecture using a genetic algorithm. Finally, we will conclude by discussing the training and performance evaluation of the model.

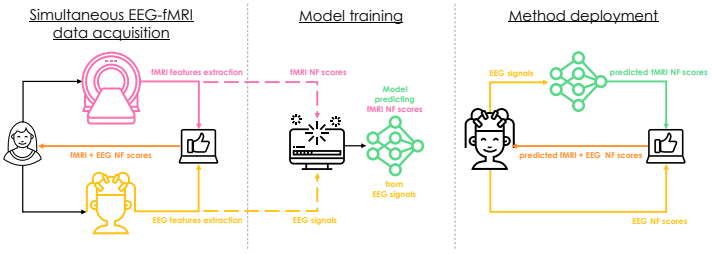


Figure 1: **Illustration summarizing our approach and objectives for predicting fMRI neurofeedback scores from EEG signals.** Firstly, simultaneous EEG-fMRI data are acquired during neurofeedback sessions. This data is used to build a model capable of predicting fMRI NF scores from EEG signals alone. Our ultimate goal is to use the previous model learned from the bi-modal sessions to enhance the EEG unimodal sessions by proposing an improved NF score that incorporates the model fMRI predictions.

3.1 Formatting of the dataset

We have decided to set up two ways of generating our dataset in order to compare both approaches. Firstly, we created supervised learning samples directly from the raw signals by associating relevant parts of the EEG signal with the corresponding fMRI NF scores, which is the outcome we aim to predict. Alternatively, we extracted features from the raw EEG signals, potentially facilitating model learning, and similarly associated relevant segments with the corresponding fMRI NF scores.

3.1.1 From raw signals to supervised learning samples

Initially, we had raw EEG signals acquired with 64 channels at a sampling rate of 200 Hz over 320 seconds per run, resulting in 64,000 points per channel and per run. The first decision we made involved dimensionality reduction, selecting signals only from 25 channels referred to as motor electrodes ('F3', 'F4', 'C3', 'C4', 'Fz', 'Cz', 'FC1', 'FC2', 'CP1', 'CP2', 'FC5', 'FC6', 'CP5', 'CP6', 'F1', 'F2', 'C1', 'C2', 'FC3', 'FC4', 'CP3', 'CP4', 'C5', 'C6', 'CPz'), chosen for their proximity to the motor area under study (see Figure 2).

To facilitate reproducibility of the study, we want to give extra details about where to find the fMRI NF scores

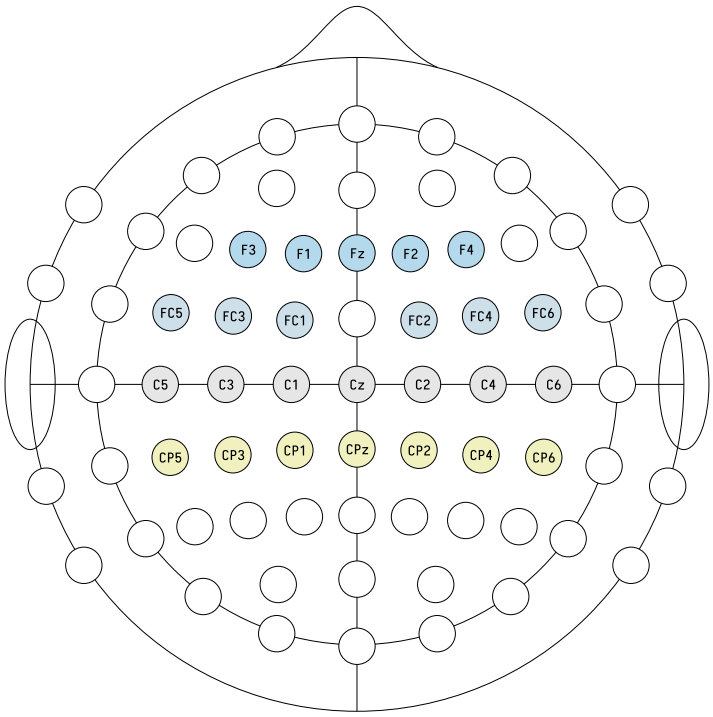


Figure 2: **Channels selected for the creation of the samples.** The colored electrodes on the diagram correspond to the channels selected for creating our samples due to their proximity to the motor area under study. The electrode corresponding to the empty space between Cz and Fz, known as FCz, was not selected here as it was used as the reference electrode, as described in section 2.2.

that we used. The derivatives of our previously described data available on OpenNeuro include NF scores calculated for both the primary motor cortex area (M1) and the supplementary motor area (SMA). M1 is primarily responsible for the execution of voluntary movements, whereas SMA is involved in the planning and coordination of complex movements. Additionally, for each of these NF targets, both raw NF scores and smoothed NF scores are provided. Raw NF scores were calculated according to the formula presented in section 2.4, whereas the smoothed version of the NF scores was computed over the preceding three volumes. For this study, we decided to use the raw NF scores calculated for the M1 area as the outcome we aim to predict.

We initially had 320 fMRI NF scores per run, representing 1 NF score per second. For the following steps, we increased this number to 1280 NF scores per run by linear interpolation, resulting in 4 NF scores per second. The reasoning behind this comes from the fact that the EEG NF scores have a size of 1280 per run. Therefore, having the same size for both fMRI and EEG NF scores will prove useful for the analysis in the results section 4. Furthermore, this idea of harmonizing towards 1280, rather than 320, serves as a form of data augmentation where we artificially increase the number of samples to be given to the model during training. We then proceeded with a calibration step. This involved retrieving the fMRI NF scores from the three runs of the same subject, calculating the 70th percentile for these three runs, dividing the scores by this value, and finally clipping the scores to be within the range of 0 to 1. This corresponds most closely to the real scores shown to the subjects during neurofeedback. In the following, this outcome we aim to predict will be referred to as true fMRI NF scores, as opposed to our predicted fMRI NF scores.

Next, we proceeded to create the supervised learning samples by associating each true fMRI NF score with a corresponding segment of the EEG data. Since fMRI NF scores are derived from the BOLD signal in the motor ROI, as well as from the signal averaged across the last 6 seconds of the previous rest block as explained in section 2.4, we decided to select the equivalent of 6 seconds of the EEG signal preceding the corresponding fMRI NF score, along with the last 6 seconds of the preceding rest block. Consequently, this selection does not allow for the creation of samples for the first 6 seconds, resulting in the exclusion of the initial 24 fMRI NF scores out of the total 1280 per run. This process is illustrated in Figure 3.

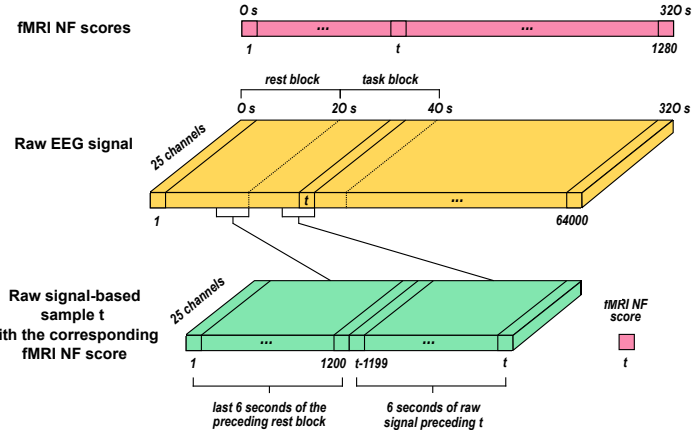


Figure 3: **From raw signals to supervised learning samples.** The sample corresponding to the fMRI NF score at time t is created by combining the last 6 seconds of raw EEG signal from the preceding rest block with the 6 seconds of the raw EEG signal preceding t.

3.1.2 From extracted features to supervised learning samples

In this approach, we initially had the same materials at our disposal: the raw EEG signals from the 25 selected channels and the calibrated fMRI NF scores. The additional step involves extracting features from the EEG signal instead of using the raw signal. We chose to extract the bandpower in the alpha range (8-12 Hz) and the beta range (12-30 Hz)

over a 2-second window with a shift of 0.05 seconds. In the previous approach, we used the channels as features, resulting in 25 features per time point. Here, we calculate both alpha and beta power bands for each of the 25 channels, resulting in a total of 50 features per time point. In the same manner, we then associated each fMRI NF score with the corresponding segment of the EEG data. As the previous approach, the segment consisted of the equivalent of 6 seconds of EEG bandpowers preceding the corresponding fMRI NF score, along with the last 6 seconds of bandpowers from the preceding rest block. In the end, this approach does not allow for the creation of samples for the first 6 seconds either. Additionally, 2 more seconds are excluded due to the feature extraction window size, resulting in the exclusion of the initial 32 fMRI NF scores out of the total 1280 per run. This process is illustrated in Figure 4.

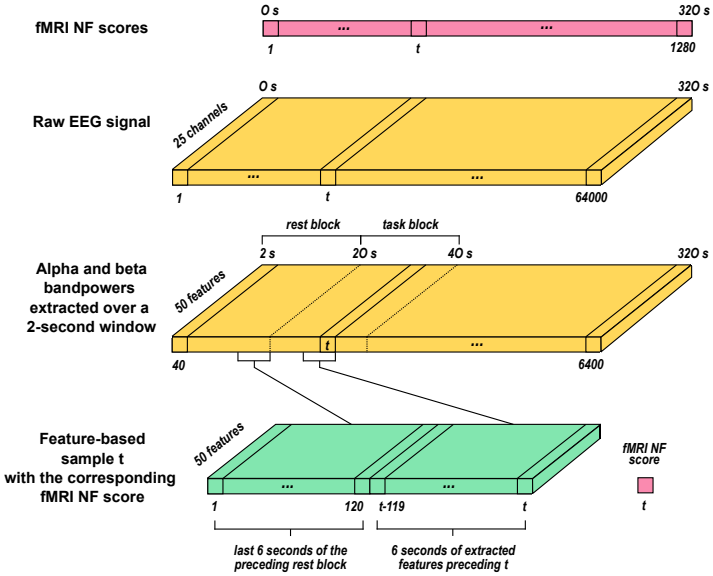


Figure 4: **From extracted features to supervised learning samples.** Features are extracted by computing alpha (8-12 Hz) and beta (12-30 Hz) bandpowers over a 2-second window with a shift of 0.05 seconds. The sample corresponding to the fMRI NF score at time t is created by combining the last 6 seconds of extracted EEG features from the preceding rest block with the 6 seconds of extracted EEG features preceding t.

3.2 Genetic search for neural network architecture

We proposed a genetic algorithm framework with the aim of making it adaptable to several types of network architectures, such as CNNs or LSTMs. The population in the genetic search consists of individuals. An individual is a set of architecture hyperparameter values, which differs according to the type of network chosen. These hyperparameter values are used to build a model architecture, which is then evaluated to score the individual and ultimately select the best architecture. The full training and testing of the model based on the chosen network took place at a later and independent stage.

3.2.1 Genetic search algorithm overview

A genetic algorithm involves three main components: the initialization of the population, the scoring of individuals, and the evolution of the population. Our genetic search, outlined in Algorithm 1, takes into account a pre-selected set of hyperparameter values that depends on the type of architecture we aim to optimize. Additionally, it needs the data described above, a fixed number of individuals constituting the population, and finally, the number of generations to be processed. Detailed specifications are provided in section 3.2.5.

The genetic search begins by initializing the individuals, each corresponding to a set of hyperparameters which values are taken at random from the pre-selection, as described in Algorithm 2. For each generation, the algorithm iterates through the population, training a neural network model with hyperparameters defined by the current individual. The performance, referred to as score, of each individual is then assessed using the mean squared error (MSE) metric, as outlined in Algorithm 3. After evaluating all individuals in a generation, the population undergoes evolution, which involves processes such as the selection of the best individuals in this generation as parents, crossover to create offspring, and mutation of some hyperparameter values also known as genes, as detailed in Algorithm 4. This evolution creates a new population for the next generation. The loop continues until we reach the desired number of generations, and at the end of the process, the algorithm returns the individual with the best performance from the last generation.

3.2.2 Initialization of the population

For the initialization step, we need the set of hyperparameters and their pre-selected set of values, as well as the desired number of individuals within a population.

The initialization algorithm consists of assigning randomly those values to each newly created individual. In this manner, it iterates over each individual in the population and, for each individual, iterates over each hyperparameter. During this process, it selects a value at random for each hyperparameter from a pre-defined set of possible values. This random creation process provides some diversity in the initial population, which is the starting point of a broad exploration of the hyperparameters space. Once values for all hyperparameters are chosen for each individual, the algorithm returns the population for the first generation.

Algorithm 1: Genetic search

Input: A pre-selected set of hyperparameter values to search for, the data, the number of individuals in the population, the number of generations.

Output: The best individual.

$P_0 \leftarrow$ Initialize the individuals in the population as described in Algorithm 2.
for *each generation* n **do**
  for *each individual* i **do**
    $Score_i \leftarrow$ Evaluate the individual's performance, as described in Algorithm 3.
  end
  if *generation* $<$ *number of generations* **then**
    $P_n \leftarrow$ Evolve the population, as described in Algorithm 4.
  end
end
Return the individual that has the best score out of the last generation.

Algorithm 2: Initialization of the population

Input: The pre-selected set of hyperparameter values to search for, the number of individuals in the population.

Output: A population.

for *each individual* i **do**
  for *each hyperparameter* h **do**
    $i_h \leftarrow$ Select a value at random from the pre-selection.
  end
end
Return the population consisting of the desired number of individuals.

3.2.3 Scoring of individuals

During this crucial step, the individuals taken as input undergo training and scoring using the same training, early stopping, and scoring datasets. These datasets consist of 13 subjects for training, 1 subject for early stopping, and 1 subject for scoring.

The method begins by creating a neural network architecture based on the hyperparameter values specified in the individual. Then, it proceeds to train it on the training dataset, using the early stopping dataset to mitigate underfitting and overfitting. Once training is done, the model is assessed on the scoring dataset by generating predictions and computing the mean squared error (MSE) with true fMRI NF scores. The mean of these errors, called a model score, serves as a quantitative measure of the individual's performance, reflecting how well the neural network, designed with the specified hyperparameters from the individual, can accurately predict the desired output. The lower the score, the better the individual's performance.

Algorithm 3: Scoring of individuals

Input: An individual i, the training dataset, the early stopping dataset, the scoring dataset.

Output: A score which represents the performance of the individual.

$Model_i \leftarrow$ Initialize the neural network architecture with hyperparameter values taken from individual i.
$Model_i \leftarrow$ Train the model with the training and early stopping datasets.
$Score_i \leftarrow$ Evaluate the model with the scoring dataset.
Return the MSE between fMRI NF predictions and true fMRI NF scores over the scoring dataset.

3.2.4 Evolution of the population

Finally, for the last major step, we once again need the set of hyperparameters, as well as the previous population whose individuals have been trained and evaluated.

The evolution process involves selecting the top n individuals from the previous population as parents. To introduce diversity, i new individuals are randomly created and added to the list of parents. Specifications are provided in section 3.2.5. These parents are then included in the new population. To maintain a consistent number of individuals across generations, the population is supplemented with offspring generated from these parents. To create an offspring, two parents are chosen randomly, and a crossover process occurs, where the value of each hyperparameter is randomly selected from either parent. A small chance of mutation is then introduced, meaning there is a probability p that one hyperparameter value is replaced by a new one chosen at random from a pre-selected set of values. This combination of selection, crossover, and mutation aims to create a new population with a mix of well-performing individuals from the previous generation and potentially new individuals that explore different regions of the hyperparameter space. The algorithm concludes by returning the updated population, ready for the next generation of the genetic search.

3.2.5 Implementation details

To begin with, we will describe the choices made for the inputs to the aforementioned algorithms. Firstly, an important step is to define the search space from which the

Algorithm 4: Evolution of the population

Input: The pre-selected set of hyperparameter values to search for, the previous population.

Output: Updated population after performing selection, crossover, and mutation operations on the previous population.

$Parents \leftarrow$ Select the best individuals.
$Parents \leftarrow$ Add new individuals, randomly created in the same manner as outlined in Algorithm 2.
$Population \leftarrow$ Add parents to the new population.
for *each remaining space in the population* **do**
  $Offspring \leftarrow$ Take two distinct parents at random, then perform crossover by randomly selecting the value of each hyperparameter from one of the two parents.
  $Offspring \leftarrow$ Mutate with a probability p one hyperparameter value of the offspring.
  $Population \leftarrow$ Add the offspring to the new population.
end
Return the new population.

genetic algorithm will select values to create individuals. The search space varies depending on the type of architecture being searched for.

- LSTM:
  - Number of layers: [1, 2, 3]
  - Number of nodes: [1, 2, 3, 4, 5, 10, 20, 30, 40, 50]
  - Number of neurons in the dense layer: [32, 64, 128, 256, 512]
  - Dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Kernel regularizers value: [0.0005, 0.001, 0.005, 0.01]
- CNN:
  - Number of convolutional layers: [2, 3, 4]
  - Number of filters in the first layer: [16, 32, 64, 128]
  - Kernels size: [3, 9, 25, 65, 95, 125]
  - Number of neurons in the dense layer: [64, 128, 256, 512]
  - Spatial dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Kernel regularizers value: [0.0005, 0.001, 0.005, 0.01]

These values are referred to as the pre-selection. For both architecture types, we excluded values that we empirically considered either too low or too high while maintaining a wide amplitude, such as dropout values of 0.1 and 0.9, and regularizer values below 0.0005 and above 0.01. Additionally, considering hardware limitations, we opted for reasonable values. For LSTMs, this included a maximum of 3 layers, a maximum of 50 nodes per layer, and a maximum of 512 neurons in the dense layer. For CNNs, we opted for a maximum of 4 convolutional layers, a maximum of 128 filters in the first layer, a kernel size no greater than 125, and a maximum of 512 neurons in the dense layer. It's important to note that the architecture search space was restricted to configurations achievable with the chosen pre-selected hyperparameters. LSTM models were constrained to have the same number of nodes for all layers. For CNNs, we chose a consistent pattern for the number of filters, doubling the number from the preceding layer. This rule applied starting with the first layer, which was the only layer under consideration in the search process.

Regarding the rest of the genetic search algorithm inputs, we reiterate that the datasets used in the genetic search comprised the three runs of 13 subjects for the training dataset, the three runs of one subject for the early stopping dataset, and the three runs of one subject for the scoring dataset. The creation of these samples has been detailed in section 3.1. Finally, we set the number of individuals in the population to 10, and the number of generations to 10. All input values were chosen expecting a balance between having ample research depth and accommodating hardware limitations.

Secondly, regarding the training hyperparameters, we fixed the maximum number of epochs at 300 and implemented early stopping with a patience of 10 and restoration of the best weights. The batch size was set to 32, the training loss function used was mean squared error, and the Adam optimizer was used with an initial learning rate of 1e-05.

Finally, regarding the choices made for the evolution of the population, given that we fixed the number of individuals in the population to 10, we chose to keep the top 2 individuals as parents and introduce 1 randomly created individual into the parents' list. Consequently, to achieve the final population size of 10 individuals, 7 offspring were generated, each with a 20% probability of mutation occurring in a randomly selected hyperparameter value.

3.3 Post-genetic model for fMRI NF scores prediction

3.3.1 Post-genetic model training

Once the best architecture design was selected, we proceeded with the final phase: model training. From the preceding step, we retained only the architecture hyperpa-

rameters of the individual with the best performance. We did not retain the weights of the trained models, as genetic search and model training were designed to be separate phases. For training hyperparameters, we used similar values to those employed during the genetic search phase. Early stopping was implemented with a patience of 10 and restoration of the best weights, the batch size was set to 32, the training loss function used was mean squared error, and the Adam optimizer was used with an initial learning rate of 1e-05. However, we fixed the maximum number of epochs at 500 to allow more freedom for the models to converge. This phase was less time-consuming than the genetic search phase since we only had to train 15 models, as explained in the next section, compared to the 100 models (i.e., 10 generations with 10 models each) in the genetic search phase.

3.3.2 Post-genetic model evaluation

To evaluate our method, we used the selected neural network architecture and conducted 15 training processes on different data arrangements, resulting in 15 different models trained on the same architecture. Notably, the partition used during the genetic search corresponds to the 15th fold of the model evaluation conducted here.

Table 1: **Details of the composition of the 15 folds used.** Each fold consists of three datasets: training, early stopping, and test. The three runs from the same subject are always grouped together within the same dataset.

Fold ID	Test	Early stopping	Training
Fold 1	sub-xp201	sub-xp204	all 13 others
Fold 2	sub-xp204	sub-xp205	all 13 others
Fold 3	sub-xp205	sub-xp206	all 13 others
Fold 4	sub-xp206	sub-xp207	all 13 others
Fold 5	sub-xp207	sub-xp210	all 13 others
Fold 6	sub-xp210	sub-xp211	all 13 others
Fold 7	sub-xp211	sub-xp213	all 13 others
Fold 8	sub-xp213	sub-xp216	all 13 others
Fold 9	sub-xp216	sub-xp217	all 13 others
Fold 10	sub-xp217	sub-xp218	all 13 others
Fold 11	sub-xp218	sub-xp219	all 13 others
Fold 12	sub-xp219	sub-xp220	all 13 others
Fold 13	sub-xp220	sub-xp221	all 13 others
Fold 14	sub-xp221	sub-xp222	all 13 others
Fold 15	sub-xp222	sub-xp201	all 13 others

Our data consists of 15 subjects, each with 3 NF runs of 1256 samples for the raw signal-based approach and 1248 samples for the extracted features-based approach. Using these subjects, we performed 15 different permutations, referred to as folds, in a leave-one-subject-out cross validation manner. Each fold corresponds to a different partitioning where one subject is used to test the model and evaluate its performance, another one to apply an early stopping strategy to avoid overfitting, and the rest to learn the model weights. They are respectively referred to as the test, early stopping, and training datasets. So, in fold i, the test dataset consisted of the 3 runs of subject i, the early stopping dataset of the 3 runs of subject $i+1$, and the training dataset of the 3 runs of all the other subjects. The detailed list of permutations is provided in Table 1. Such an approach facilitated the training and evaluation of 15 models built with the same architecture, enhancing the robustness of our method's performance assessment.

Finally, to assess the performance of these models, we predicted the fMRI NF scores for each sample of each test run and compared them with true fMRI NF scores using mean squared error (MSE). The average error for these test runs represent the performance of each of the 15 trained models, and the average of these performances represents the overall performance of our method. Then, to assess the applicability of our method in real-life conditions, as presented in the deployment section of Figure 1, we proceed to the next step. Our ultimate goal is to generate NF scores that integrate both our fMRI NF predictions and EEG NF scores, aiming to approximate true bi-modal EEG-fMRI NF scores without fMRI acquisitions. Therefore, we combined both our predictions and true fMRI NF scores with the calibrated EEG NF scores (presented in section 2.4 and calibrated in the same way as fMRI NF scores in section 3.1). This combination is achieved by computing the mean between the fMRI NF scores and the EEG NF scores. Since both have been calibrated between 0 and 1, the combined scores also fall within this range. Then, we calculated the error between the fMRI NF predictions averaged with EEG NF scores and the true bi-modal EEG-fMRI NF scores. It is worth noting that the resulting average errors will be lower than the MSE between predicted and true fMRI NF scores since EEG NF scores are added on both sides. To be exact, this operation is equivalent to dividing those MSEs by 4. The purpose of this step is to obtain an estimate of the quality of this new virtually bi-modal NF score. Since our fMRI NF predictions are intended to enhance EEG NF scores, their combination should exhibit a lower error with the true EEG-fMRI NF scores than EEG-NF scores alone for us to validate our method.

4. Results

4.1 Results overview

As the results of this work are substantial and quite detailed, we begin with an overview to clarify the structure of this section. We will present the results of the models

created using two types of neural network architectures: LSTM and 1D CNN. These models were trained using two different data preparation approaches described in section 3.1: extracted features-based samples and raw signal-based samples. We start by summarizing and illustrating the performances corresponding to the fMRI NF predictions of these four configurations in Table 2 and Figure 5. Then, we will examine the final results, which incorporate EEG NF scores, in Table 3.

To begin with, Table 2 allows us to assess the performance of the four configurations using two metrics: mean squared error (MSE) and Pearson's correlation. MSE is the main metric used here, while Pearson's correlation offers additional insights into the shape of the predictions. The values presented here correspond to the error and correlation between the fMRI NF predictions and the true fMRI NF scores, representing model performance directly at the output.

In summary, the mean MSEs between fMRI NF predictions and true fMRI NF scores across the four configurations are fairly similar, but it is the CNN architecture with extracted features samples that stands out with the lowest value (0.1586). Regarding Pearson's correlations, the values are overall very low. While correlation isn't necessarily the best metric in this case, given that both predictions and true scores can feature frequent spikes and fluctuations, it serves as a useful secondary metric. It allows us to evaluate the appearance of the predictions, which is an important aspect when judging the quality of the results, as we will see in the following. We note that the two CNN configurations perform slightly better in this regard, with correlations around 0.1, while the LSTM configurations show correlations close to zero.

Then, we summarize the performance of our four configurations side by side in Figure 5. At first glance, the differences between the configurations are not obvious. We begin by comparing the neural network types: for LSTMs, the difference between the extracted features approach and the raw signal approach is not significant. However, for CNNs, the extracted features approach performs significantly better than the raw signal approach. Next, we compare network types within each approach: for the raw signal approach, the difference between LSTM and CNN is not significant, while for the extracted features approach, CNN significantly outperforms LSTM. Further details will be provided in the following sections.

Next, Table 3 presents what we refer to as the final results, compared against two baselines (n°1 and n°2) to be surpassed, which allow us to evaluate the success of the experiments. The principle behind these final results, described in section 3.3.2, consists of combining both our predictions and true fMRI NF scores with the corresponding EEG NF scores, following the goal of approximating true bi-modal EEG-fMRI NF scores without requiring fMRI acquisitions. The values presented here thus correspond to the mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores. These values do not provide any additional information compared with the Table 2 (as mentioned earlier, this operation is equivalent to dividing those values by 4), but they are necessary for comparison with the two baselines.

The first baseline (noted baseline n°1), also introduced in section 3.3.2, corresponds to the mean MSE between

Table 2: **Comparison of fMRI NF predictions across all configurations.** First column: Mean MSE between fMRI NF predictions versus true fMRI NF scores (corresponding to the pink (left) boxplot in the following figures). Second column: Mean Pearson's correlation between fMRI NF predictions versus true fMRI NF scores.

Configuration	**Mean MSE between predicted and true fMRI NF scores**	**Mean correlation between predicted and true fMRI NF scores**
LSTM extracted features samples	0.1673 (± 0.0177)	0.0418 (± 0.1114)
LSTM raw signal samples	0.1707 (± 0.0560)	-0.0162 (± 0.0572)
CNN extracted features samples	0.1586 (± 0.0212)	0.1022 (± 0.1956)
CNN raw signal samples	0.1692 (± 0.0299)	0.1151 (± 0.2283)

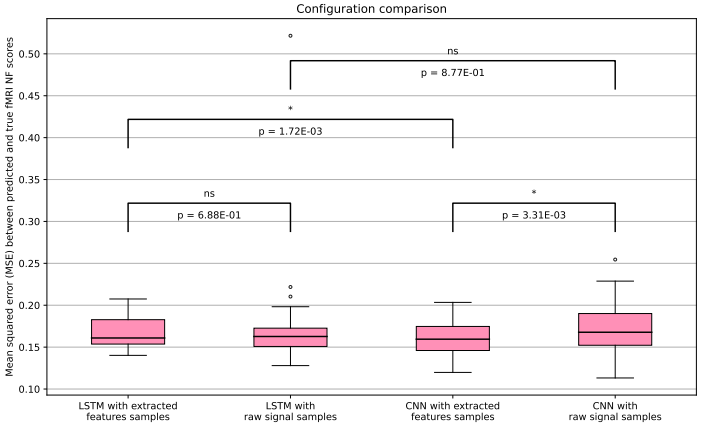


Figure 5: **Results for all test subjects across all folds using the mean squared error (MSE) metric for all configurations.** Each boxplot represent the MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores. The p-value from a paired t-test is displayed for key comparisons (*: significant, ns: not significant).

EEG NF scores and true bi-modal EEG-fMRI NF scores. The purpose of this comparison is to verify if our fMRI NF predictions are actually enhancing EEG NF scores. Ideally, the combination should yield a lower error when compared to the true EEG-fMRI NF scores than using EEG NF scores alone.

The second baseline (noted baseline n°2) represents the mean MSE between the mean of true fMRI scores averaged with the EEG NF scores versus true bi-modal EEG-fMRI NF scores. The idea is to mimic a perfectly centered flat prediction. In fact, the calculation is close to the variance of true fMRI NF scores (with the combination with EEG NF scores), but we have chosen to explain it this way to connect it with what we are trying to verify. Indeed, the purpose of this second comparison is to verify if our fMRI NF predictions are more accurate than simply predicting the mean of the true fMRI scores. The story of why we decided to look at our results with baseline n°2 will be told in the following.

So, let's analyse Table 3. We first observe that baseline n°1 and n°2 values differ slightly between the extracted features and raw signal categories. Theoretically, these values should be identical. This difference can be explained by the dataset formatting process, as presented in section 3.1. The number of samples created is not exactly the same (with 8 fewer samples per run for the extracted features approach), meaning that the true fMRI NF scores are also of slightly different sizes. This explains the minor variation between the means.

When comparing our final results with baseline n°1, we find that all configurations outperform it, symbolizing that adding our fMRI NF predictions to EEG NF scores brings us significantly closer to the true bi-modal EEG-fMRI NF scores compared to using EEG NF scores alone. This is excellent news, as it suggests that our models effectively enhance EEG neurofeedback by incorporating fMRI predictions. However, as we will see in section 4.3, the LSTM configuration using raw signal-based samples unexpectedly produced flat-looking predictions, often centered around the mean of the true fMRI NF scores. Even more surprisingly, the mean MSE between these predictions and the true scores was very close to that of the LSTM models using extracted features-based samples, which produced predictions with greater amplitude. This observation raised the question of whether merely surpassing baseline n°1 is sufficient for our method to be considered satisfactory.

Therefore, we turn our attention to baseline n°2. Here, we observe that none of the four configurations manage to outperform it. As we will see in the next sections, the LSTM with raw signal approach and the CNN with extracted features approach have lower performance than baseline n°2, but not significantly so (with a p-value of 0.0882 and 0.2039 respectively (details provided in their respective sections), which exceeds the 0.05 threshold, indicating that we cannot reject the null hypothesis of identical average scores). The comparison with baseline n°1 showed us that we were indeed predicting something interesting from the EEG signals alone, despite considerable differences between the two modalities. However, these predictions were in fact no better than simply predicting the average of the true fMRI NF scores.

Table 3: **Comparison of final results across all configurations.** First column: Mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot in the following figures). Second column: Mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot in the following figures), referred to as baseline n°1. Third column: Mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot in the following figures), referred to as baseline n°2.

Config.	**Mean MSE for final results**	**Baseline n°1 (MSE)**	**Baseline n°2 (MSE)**
LSTM extracted features samples	0.0418 (± 0.0044)	0.0829 (± 0.0198)	0.0384 (± 0.0037)
LSTM raw signal samples	0.0427 (± 0.0140)	0.0831 (± 0.0196)	0.0388 (± 0.0037)
CNN extracted features samples	0.0397 (± 0.0053)	0.0829 (± 0.0198)	0.0384 (± 0.0037)
CNN raw signal samples	0.0423 (± 0.0075)	0.0831 (± 0.0196)	0.0388 (± 0.0037)

In the next four sections, we will examine and discuss the results of each configuration presented in Tables 2 and 3. For each section, we will provide: (1) an illustration of the architecture identified through genetic search, (2) learning curves for the 15 trained folds, (3) overall performance represented by boxplots using the mean squared error (MSE) metric, (4) prediction examples showcasing the highest and lowest errors across all folds, and (5) an analysis figure of the relationship between the quality of predictions and the quality of the EEG input signals.

4.2 LSTM model with extracted features samples as inputs

This section presents the results of applying our method to the LSTM network type using extracted features-based input data. After running the genetic algorithm for hyperparameter optimization, the architecture hyperparameters found are as follows: one LSTM layer with 40 units, followed by a dense layer with 256 neurons. Regularization was applied to both the LSTM and dense layers with a value of 0.001, and the dropout rate for the dense layer was set at 0.3. An illustration of this network is provided in Figure 6.

The use of only one LSTM layer, combined with a relatively large number of units, suggests a focus on capturing simple patterns in the input sequence. The dense layer, with its substantial number of neurons, contains the majority of the model's weights, and could be prone to overfitting. However, the regularization applied, while moderate, helps mitigate this risk by penalizing large weights. Additionally, the modest dropout rate in the dense layer further reduces the likelihood of overfitting by randomly setting a fraction of the layer's output units to zero during training. The relatively low regularization and dropout values indicate that significant overfitting mitigation was not necessary, which is typical for one-layer models like this one. This suggests that during the genetic search, this simpler network might have identified trends in the data better than bigger models, possibly because the input features were pre-extracted, simplifying the task. This interpretation will be contrasted with the results from raw signal-based samples in the next section.

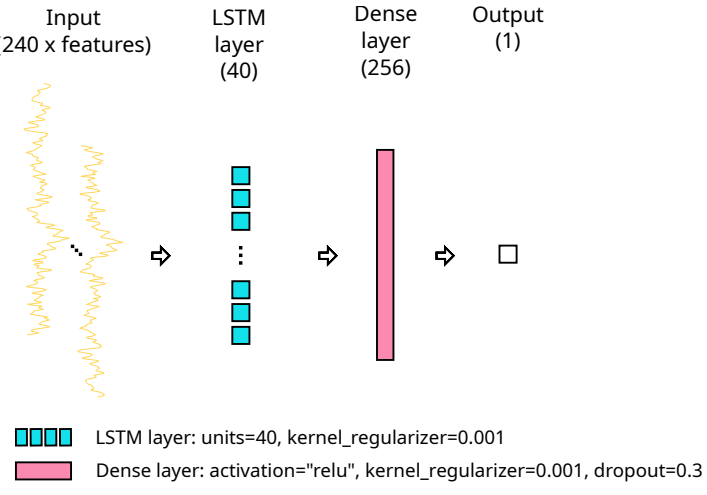


Figure 6: **Architecture of the LSTM found through the genetic search, using extracted features samples as input.**

Then, as described in section 3.3.2, we trained 15 different models using the selected architecture by permuting the data within the training, early stopping, and testing datasets to achieve a more robust evaluation of our method. Figure 7 displays the 15 learning curves, depicting the loss calculated on both the training and early stopping datasets. The learning curve generated from the training dataset provides insight into how well the model learns, while the curve derived from the early stopping dataset, which is not used for updating weights, assesses the model's generalization ability. This validation curve allows us to halt the learning process before overfitting occurs, thanks to the early stopping mechanism.

The figure shows a general trend of good convergence between the training and validation loss curves across the 15 folds. However, in some instances, the validation loss is noisier and does not decrease as much as the training loss. This can be attributed to several factors. Firstly, the fact that the validation loss is higher than the training loss may suggest that, while the model fits the training data well, it faces some challenges in generalizing to unseen data in the validation set. It typically indicates early signs of overfitting. Secondly, the noise in the validation loss could be due to variability in the data distribution between the training and validation datasets. Since the data is permuted across different folds, some validation subsets might include more challenging samples. This case demonstrates the importance of early stopping, which stops training before the model diverges too far from the equilibrium.

Now, we present the comprehensive results following the complete application of our method. As previously explained, we consider the predictions from all 15 trained models on their respective test subjects. Figure 8 shows four key comparisons using the mean squared error (MSE) metric. Firstly, we directly compare the predicted fMRI NF scores to the true fMRI NF scores (corresponding to the pink (left) boxplot), providing a direct evaluation of model performance. Secondly, we compare our fMRI NF predictions averaged with EEG NF scores to the true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot). While adding the same EEG NF scores on both sides naturally reduces the MSE, it allows us to evaluate the predictions in a context more relevant to our goal of improving unimodal EEG neurofeedback sessions. To evaluate this final result, we include baseline n°1 (corresponding to the purple (center right) boxplot), representing the MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, which has the purpose of verifying if our predictions averaged with EEG NF scores align more closely with the true bi-modal EEG-fMRI NF scores than the EEG NF scores alone. Lastly, we provide baseline n°2 (corresponding to the blue (right) boxplot), representing the MSE between the mean of true fMRI scores averaged with EEG NF scores versus the true bi-modal EEG-fMRI NF scores, which has the purpose of verifying if our fMRI NF predictions are more accurate than simply predicting

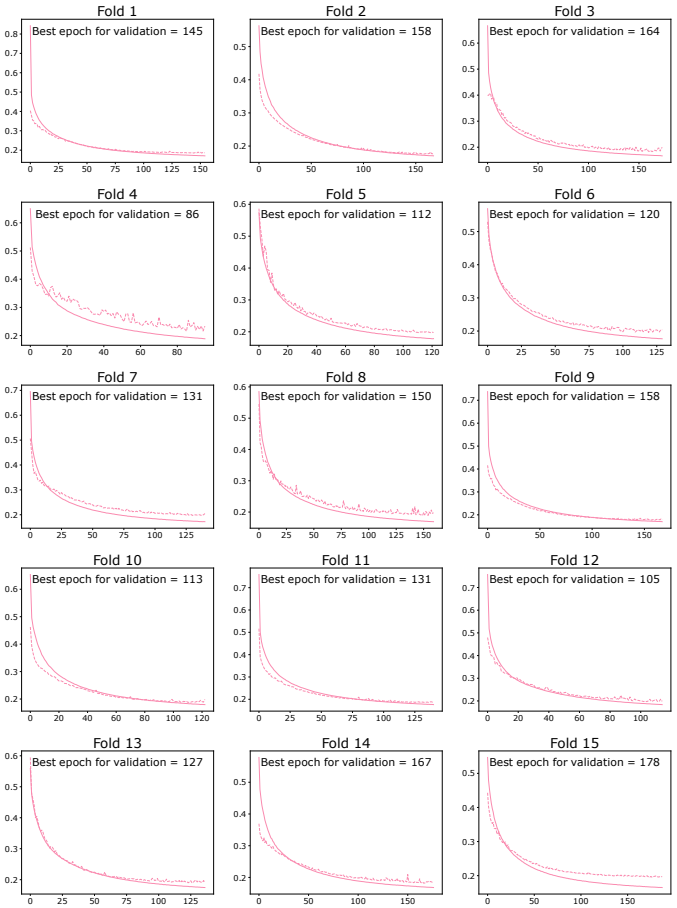


Figure 7: **Learning curves for the 15 folds of the LSTM approach with extracted features samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

the mean of the true fMRI scores.

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1673(\pm 0.0177)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0418(\pm 0.0044)$. These MSE values are not interpretable on their own, which is why we compare the final results with baseline n°1 and n°2. For baseline n°1 (purple boxplot), the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores is $0.0829(\pm 0.0198)$. By having a significantly ($p = 4.12e{-}18 < 0.05$ using a paired t-test) lower MSE when the predictions are averaged with EEG NF scores compared to EEG-only NF scores, the goal of enhancing EEG neurofeedback with predicted fMRI NF scores is reached. What is predicted, even if not perfect, seems to carry information that was not in the EEG NF scores alone. However, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0384(\pm 0.0037)$. Our predictions averaged with EEG NF scores are thus significantly ($p = 2.07e{-}4 < 0.05$ using a paired t-test) less accurate than the mean of true fMRI scores averaged with EEG NF scores.

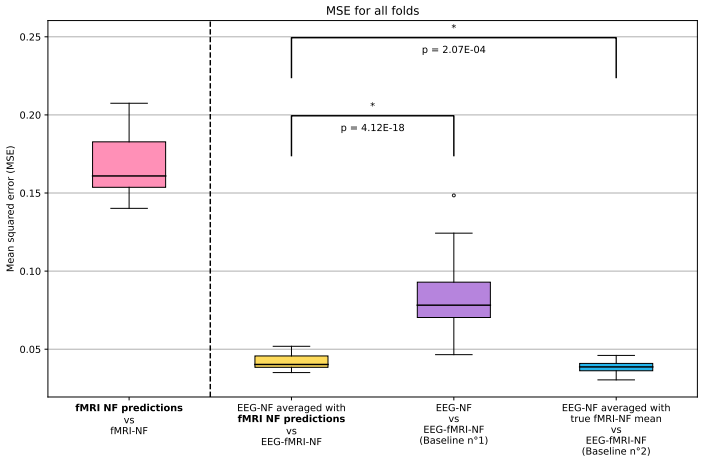
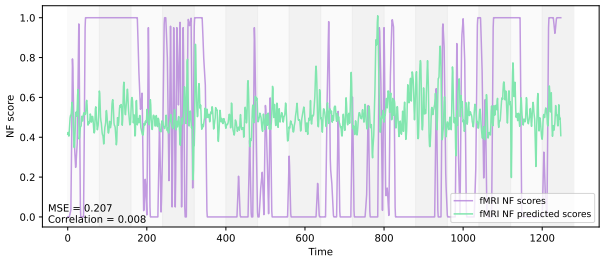


Figure 8: **Results for all test subjects across all folds using the mean squared error (MSE) metric for the LSTM with extracted features samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

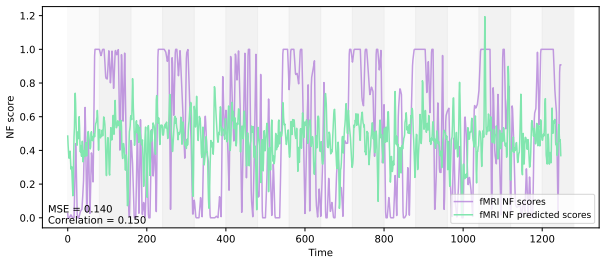
To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 9 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp213 run 2, which represents the highest error, and sub-xp218 run 3, which represents the lowest error.

In the example with the highest MSE of 0.207 (a), it is noticeable that the true fMRI NF scores exhibit an irregular pattern, not quite following the expected rest/task alternation. The predictions are spiky but centered. It appears that the model struggled to accurately predict these scores. The irregularity of the true fMRI NF scores may have contributed to the model's difficulty in generalizing to this run, leading to this high MSE value.

In contrast, for the lowest MSE example (b), the true fMRI NF scores follow more clearly the expected rest/task trend. The model's predictions, though still very spiky, seem to follow a bit better the true fMRI NF scores, resulting in a lower MSE of 0.140. We could think that the closer adherence to the expected trend in the true fMRI NF scores made it easier for the model to generalize to this run, leading to a more accurate performance.



(a) Prediction for sub-xp213 run 2



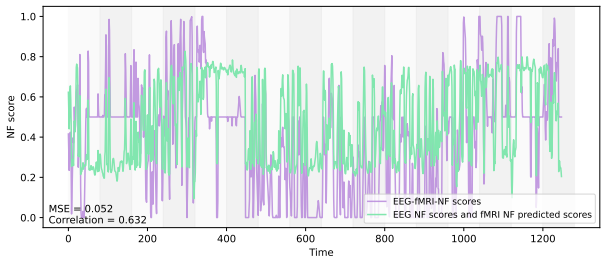
(b) Prediction for sub-xp218 run 3

Figure 9: **Examples of predictions made using an LSTM model with extracted features samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp213 run 2, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp218 run 3, representing the lowest error across all folds.

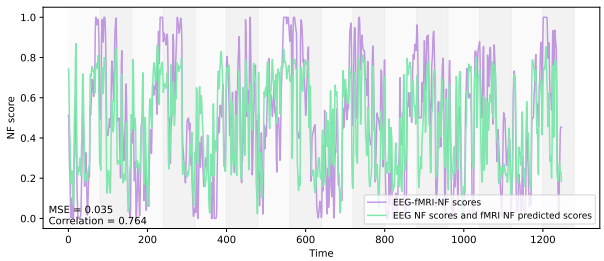
As described in section 3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 10 presents the predictions for the same subjects used in Figure 9, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores exhibit an irregular pattern, deviating significantly from the expected rest/task alternation. The presence of a few plateaus at 0.5 suggests that while one modality had an NF score of 1, the other had an NF score of 0, which is not ideal. This discrepancy, where EEG NF scores sometimes oppose the true fMRI NF scores, could indicate a lower quality of EEG signals. We will explore this possibility with the next figure. The model's predictions in this case somewhat follow the true scores, mainly due to the addition of EEG NF scores on both sides, but exhibit clear offsets and numerous spikes, leading to a mean MSE of 0.052.

For the lowest MSE example (b) with a value of 0.035, the true bi-modal EEG-fMRI NF scores follow more closely the expected rest/task trend, though they remain somewhat spiky, but to a reasonable degree. Here, the model's predictions align more closely with the true scores, capturing both the overall trend and amplitude more effectively. This better alignment could suggest that when the true bi-modal scores have a clearer rest/task pattern, the model is better at generalizing and producing accurate predictions. However, this is not exactly our goal, as we aim for the model to predict accurate fMRI NF scores regardless of whether the participant performed well during the rest and task blocks or struggled more. The next figure will investigate this idea further.



(a) Prediction for sub-xp213 run 2



(b) Prediction for sub-xp218 run 3

Figure 10: **Examples of final results made using an LSTM model with extracted features samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

So, to further analyze our results, we aim to understand why our method performs differently across subjects. To address this question, we hypothesize that the quality of the EEG signal might be a significant factor contributing to these differences. Other sources of variation might in-

clude the participant's affinity for one modality over the other during the bi-modal session (e.g., if the participant is less responsive to fMRI, predicting random scores becomes challenging). We focus on the quality of the EEG signal since it serves as the input for our models. We assume that: if a participant has EEG NF scores that follow the expected rest/task trend, it is because the participant is responding well to neurofeedback and that the EEG signal captures well this information, and therefore is of good quality. So, to evaluate the quality of EEG NF scores, reflecting the quality of our input EEG signals, we used the t-statistic measure between the task and rest values of these scores. A high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero means that the participant responded better to neurofeedback during rest than during the task, implying that the EEG signal might not be of high quality and/or that the participant did not understand the instructions (e.g., they might be thinking of the task during rest). Figure 11 shows the performance of our fMRI NF predictions against the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

Firstly, we look at the t-statistic between the task and rest blocks of the EEG NF scores (abscissa). This analysis will be valid for all four sections, as the same values of EEG NF scores are used each time. We observe that all test runs across all folds have t-statistics ranging from approximately -10 to 25. Although it's difficult to define an exact threshold at which a run is considered to have a "good" rest/task trend in general, we can at least observe that the vast majority of runs have a positive t-statistic. However, a substantial number are still close to zero, indicating a more lukewarm neurofeedback response. Finally, sub-xp211 appears to be a clear outlier, with very negative t-statistics for 2 of its runs and a t-statistic close to 0 for the last run.

Then, regarding the MSE between predicted and true fMRI NF scores (ordinate), we observe that values range from approximately 0.14 to 0.21. The lowest MSE example run that we observed earlier, from sub-xp218, indeed corresponds to a better t-statistic value (t-stat ≈ 15) than the highest MSE run (t-stat ≈ 5). A trend could be observed in the $0 - 15$ t-statistic range, where higher MSEs are closer to a t-statistic value of 0 and lower MSEs are associated with increasing t-statistics. However, the correlation coefficient value of -0.085 does not allow us to conclude that a higher t-statistic is systematically linked to a lower MSE. In fact, some runs with t-statistics in the 20-25 range have some of the highest MSEs. Therefore, in this section, it appears that the quality of the EEG NF scores cannot be considered as a factor contributing to the difference of performance between runs.

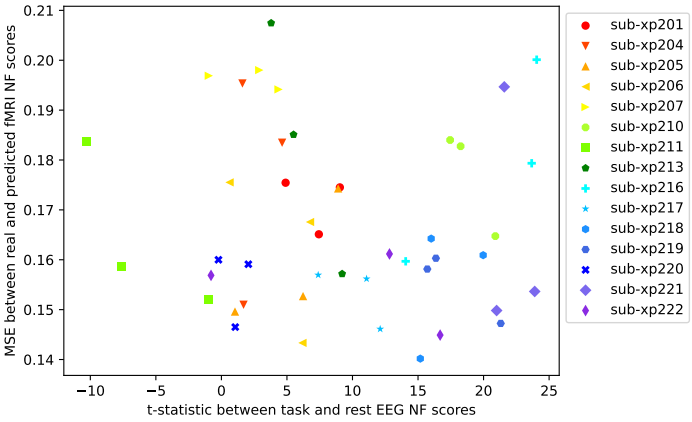


Figure 11: **Analysis for the LSTM approach with extracted features samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.085.

4.3 LSTM model with raw signal samples as inputs

This section presents the result of our method applied to the LSTM network type with raw signal-based input data. After running the genetic algorithm to search for architecture hyperparameters, the architecture found includes three LSTM layers with 4 units each, followed by a dense layer containing 512 neurons. Kernel regularizers applied to the LSTM and dense layers were set to 0.01. Finally, the dropout rate for the dense layer was set at 0.2. An illustration of this network is available in Figure 12.

The architecture identified by the genetic algorithm is notably more complex than the one used for extracted features inputs. The presence of three LSTM layers provides a deeper network structure designed to capture more intricate patterns within the raw signal inputs. However, having only 4 units per layer might be considered a bit too small. The dense layer, containing 512 neurons, offers significant capacity for processing the output of the LSTM layers. While this substantial number of neurons contributes to a more powerful model, it also increases the risk of overfitting. To mitigate this risk, the regularization applied to both the LSTM and dense layers is relatively high. However, the dropout rate in the dense layer is surprisingly modest. The similar performances shown in Table 2 for the raw signal and the extracted features approach suggests that although this model is more complex, it may

face greater challenges in extracting meaningful patterns directly from the raw signal inputs. It could also indicate that the model is working harder to achieve a similar level of performance as the simpler model that uses extracted features inputs.

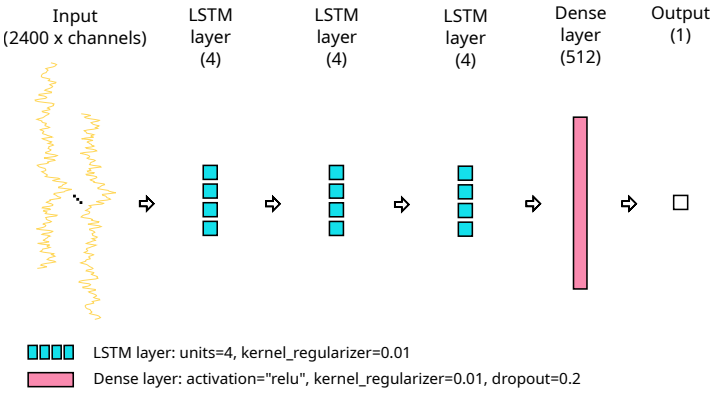


Figure 12: **Architecture of the LSTM found through the genetic search, using raw signal samples as input.**

Figure 13 shows the learning curves for the LSTM models trained on raw signal inputs, based on the same evaluation method described previously. As before, these curves illustrate both the training and validation losses across 15 models, providing insights into the model's learning and generalization.

The curves here present an interesting observation: in some instances, the validation loss is unexpectedly lower than the training loss. This behavior, though less common, can be attributed to several factors. Firstly, it might suggest that the regularization techniques, such as dropout or kernel regularizers, are having a strong impact on the training process. Since regularization is only applied during training, it could result in a higher training loss compared to the validation. Secondly, as before, some of the validation subsets may contain samples that are inherently easier to predict than those in the training data. Overall, this may indicate that the data is complex and challenging to model.

Same as the previous section, Figure 14 presents four key comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

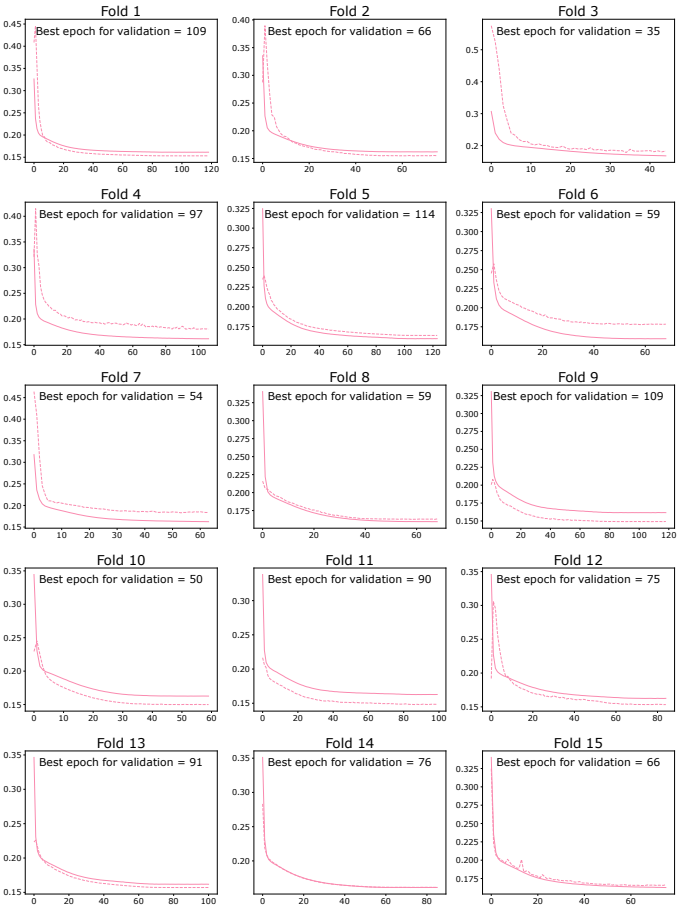


Figure 13: **Learning curves for the 15 folds of the LSTM approach with raw signal samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1707(\pm 0.0560)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0427(\pm 0.0140)$. Thirdly, for baseline n°1: the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (purple boxplot) is $0.0831(\pm 0.0196)$. Despite slightly higher MSE values for predictions and final results compared to the extracted features approach, the predictions averaged with EEG NF scores still significantly ($p = 4.17e{-}15 < 0.05$ using a paired t-test) outperform baseline n°1. Fourthly, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores ver-

sus true bi-modal EEG-fMRI NF scores is $0.0388(\pm 0.0037)$. Here, as the p-value computed using a paired t-test is $p = 0.0882 > 0.05$, we cannot reject the null hypothesis of identical average scores, meaning that our predictions averaged with EEG NF scores have a similar accuracy as the mean of true fMRI scores averaged with EEG NF scores.

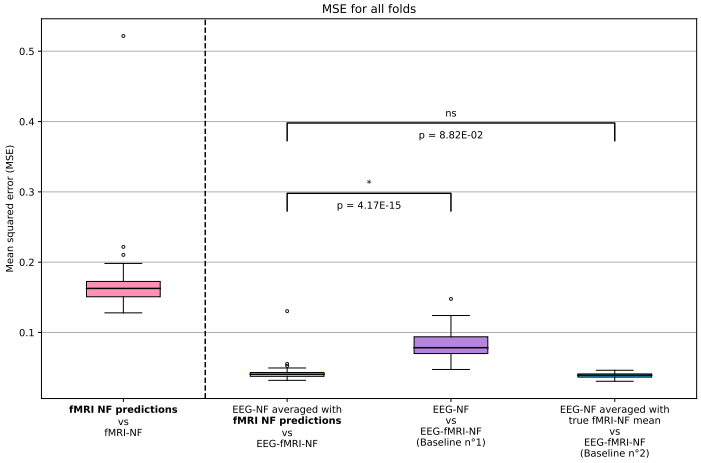
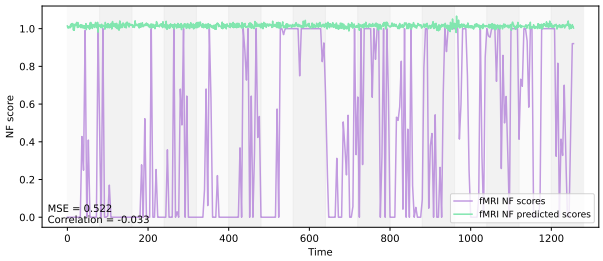


Figure 14: **Results for all test subjects across all folds using the mean squared error (MSE) metric for the LSTM with raw signal samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

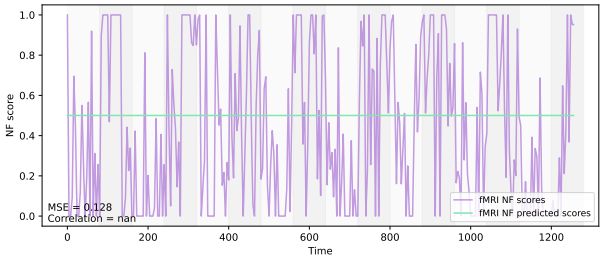
To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 15 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp206 run 3, which represents the highest error, and sub-xp204 run 3, which represents the lowest error.

For the highest MSE example (a), the true fMRI NF scores exhibit an irregular pattern, similarly to the previous section. Very surprisingly, the model's predictions are almost flat, with only small variations, and a noticeable offset from the mean of the true scores. This result has a mean MSE of 0.522, marking it as a clear outlier.

For the lowest MSE example (b) with a value of 0.128, although the true fMRI NF scores approximately follow the expected rest/task trend at certain points, they remain quite spiky. Even more surprisingly, the model's predictions are also flat, but they are more accurately centered around the mean of the true scores. This alignment with the mean suggests some basic understanding of the overall trend. However, the fact that the average MSE over all folds with this approach is very similar to the extracted features approach, despite the very different-looking results, raises questions. Indeed, it is disturbing that these flat predictions, which "play it safe", have the same average error as previous predictions that attempted high and low predicted values. To tell the story, it was this observation that gave us the idea of looking at the results with baseline n°2 presented in section 4.1.



(a) Prediction for sub-xp206 run 3



(b) Prediction for sub-xp204 run 3

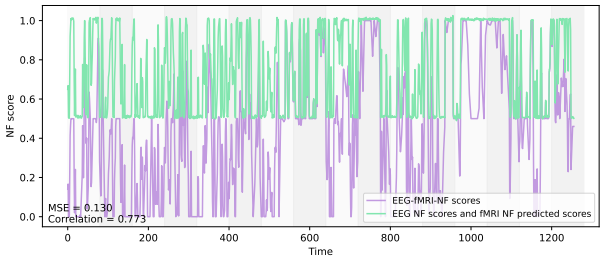
Figure 15: **Examples of predictions made using an LSTM model with raw signal samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp206 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp204 run 3, representing the lowest error across all folds.

As described in section 3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 16 presents the predictions for the same subjects used in Figure 15, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

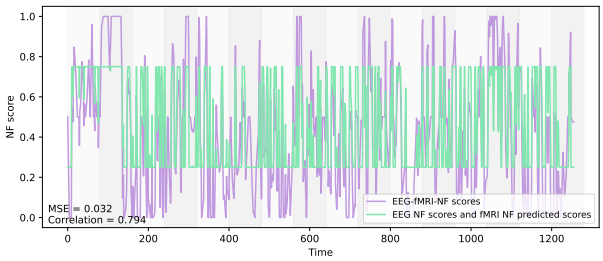
For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores also do not exhibit a clear rest/task trend. We can see very clearly that the final result here represents the EEG NF scores averaged with the flat fMRI

prediction. This outcome shows the trend of the EEG NF scores along with the significant offset from the fMRI NF prediction. This leads to poor alignment with the true scores, resulting in the section's highest mean MSE of 0.130.

For the lowest MSE example (b), the true bi-modal EEG-fMRI NF scores follow the rest/task trend slightly better. In this case, it is visible again that the final result is the EEG NF scores averaged with a flat fMRI NF prediction. However, here, the flat fMRI prediction does not exhibit a significant offset, resulting in an outcome that is better-centered around the true scores. Although the overall result initially appears to lack the desired accuracy and amplitude, it has a mean MSE of 0.032, which is very similar to the previous section's lowest MSE of 0.035.



(a) Prediction for sub-xp206 run 3



(b) Prediction for sub-xp204 run 3

Figure 16: **Examples of final results made using an LSTM model with raw signal samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 17 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are identical to those from the previous section, as the same values of EEG NF scores are used. In summary, the vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 stands out as a clear outlier.

Now, regarding the MSE between predicted and true fMRI NF scores, we observe a surprising cluster between approximately 0.12 and 0.23, with a single outlier at 0.52 corresponding to the run shown in the example figure above. As seen in the examples, these LSTM models using raw signal-based samples tend to result in flat predictions that are often centered around the mean of the true fMRI NF scores. This suggests that the model may struggle to predict the variations in the true fMRI NF scores. Instead, it appears to default to safer, less variable predictions, which could be a consequence of the model's difficulty in extracting meaningful information from the raw signal-based EEG inputs. One possible explanation is that the LSTM architecture identified through the genetic search, especially the use of only 4 units per layer, might not be sufficient to handle the variability present in raw EEG signals. This limitation could lead to overly generalized predictions. The next question, then, is why the genetic search resulted in such an architecture. As we saw earlier, the overall performance in terms of MSE is almost the same as in the previous section, indicating that this "safe" approach is as effective as the previous models, which had more variation in their predictions. So, during the genetic search, this architecture which produces those flat predictions was likely selected as a parent due to its good mean MSE, and then was "safe" enough so that any other architecture attempting more variations could not outperform it.

4.4 CNN model with extracted features samples as inputs

In this section, we present the results of our method applied to the 1D CNN type with extracted features-based input data. After running the genetic algorithm, the architecture hyperparameters found are as follows: three convolutional layers, with the first having 32 filters (subsequent layers doubling that number from the preceding one), all with a kernel size of 3. The dense layer contains 64 neurons. Kernel regularizers (applied to the convolutional and dense layers) were set to 0.001. Moreover, a spatial dropout rate of 0.8 was employed for the convolutional layers, while the dropout rate for the dense layer was set to 0.2. An illustration of this network is provided in Figure 18.

The use of three convolutional layers, which is not the maximum allowed by the genetic algorithm, along with only

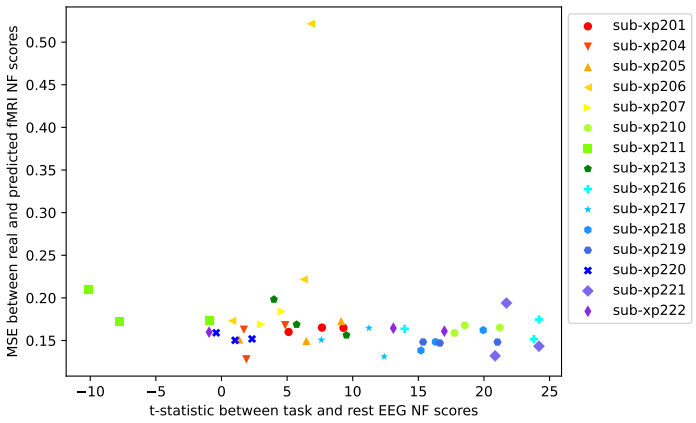


Figure 17: **Analysis for the LSTM approach with raw signal samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.145.

32 filters in the first layer and only 64 neurons in the dense layer, allows us to consider that this model is relatively small. The regularization applied is fairly low, suggesting that the model is either small enough not to require significant regularization or that the dropouts are sufficient to mitigate overfitting. And indeed, the spatial dropout rate applied to the convolutional layers is very high, which likely forces the network to learn more robust patterns. However, the dropout applied to the dense layer is surprisingly low, further indicating that the model might not be as prone to overfitting as larger models might be. This reflects a similar tendency observed in the LSTM approach, where the use of extracted features simplifies the network's task. Consequently, the genetic search selects smaller models, which appear well-suited to this kind of input data.

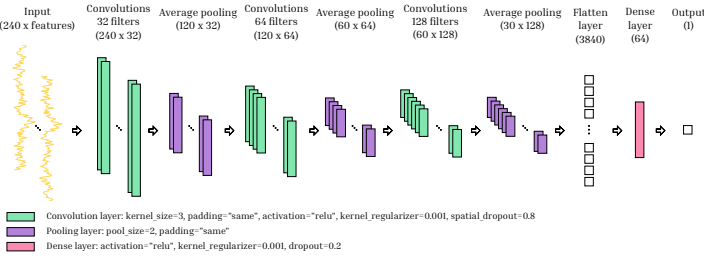


Figure 18: **Architecture of the CNN found through the genetic search, using extracted features samples as input.**

Figure 19 presents the training and validation loss curves for the 15 CNN models trained on extracted features-based samples, as described in the preceding sections.

Both loss curves demonstrate good convergence overall, indicating that the model is learning effectively from the data without significant overfitting. The close alignment of the training and validation losses suggests that the model maintains good generalization ability, adapting well to unseen data. Similarly to the LSTM approach with extracted features samples, the validation loss is occasionally slightly higher than the training loss, which could indicate minor overfitting or some variability in the data, with some validation subsets possibly containing more challenging samples. In summary, the CNN model with extracted features samples exhibits a solid training process with good convergence between training and validation loss, making it the cleanest of all sections so far.

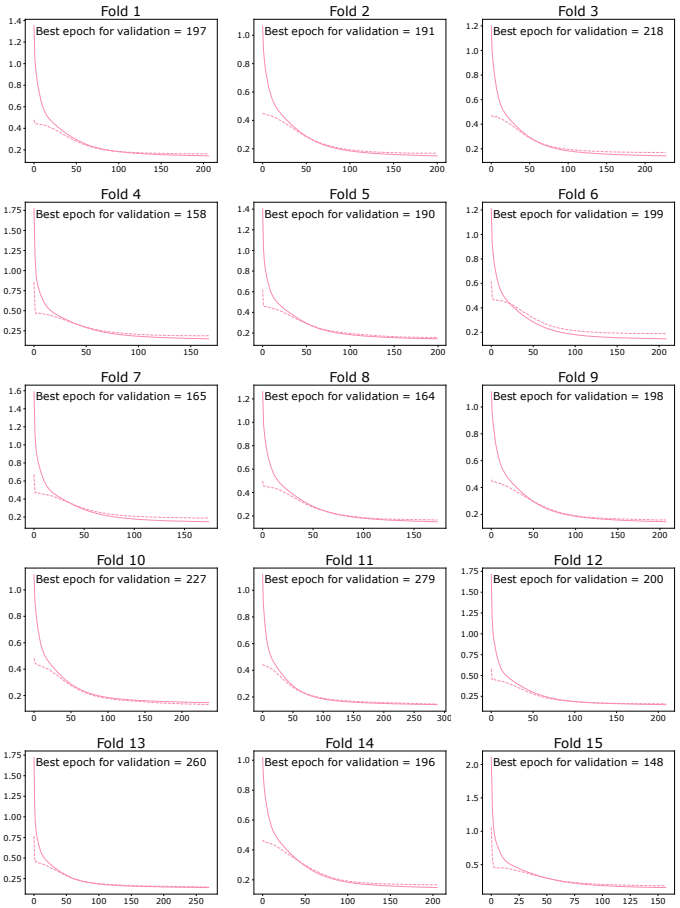


Figure 19: **Learning curves for the 15 folds of the CNN approach with extracted features samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Same as previous sections, Figure 20 presents four key

comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1586(\pm 0.0212)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0397(\pm 0.0053)$. Thirdly, for baseline n°1: the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (purple boxplot) is $0.0829(\pm 0.0198)$. The MSE values for predictions and final results are the lowest so far. Like the LSTM approaches, the predictions averaged with EEG NF scores significantly ($p = 3.40e{-}20 < 0.05$ using a paired t-test) outperform baseline n°1. Fourthly, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0384(\pm 0.0037)$. Here, as the p-value computed using a paired t-test is $p = 0.204 > 0.05$, we cannot reject the null hypothesis of identical average scores, meaning that our predictions averaged with EEG NF scores have a similar accuracy as the mean of true fMRI scores averaged with EEG NF scores.

To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 21 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp211 run 3, which represents the highest error, and sub-xp221 run 3, which represents the lowest error.

For the highest MSE example (a), the true fMRI NF scores are remarkably regular and clean, representing an ideal signal pattern. Despite this, the model's predictions are nearly flat, with occasional spikes that often move in the opposing direction during task blocks, leading to a mean MSE of 0.203. It is surprising that such perfect true scores result in the highest error across all folds. This discrepancy suggests that factors beyond the quality of the true fMRI NF scores might be influencing the model's performance, such as the quality of the EEG signals used as input.

For the lowest MSE example (b) with a value of 0.120, the true fMRI NF scores align roughly with the expected rest/task trend but remain quite spiky. The model's predictions follow the true scores somewhat closely, displaying a similar spikiness, and appear to be approximately centered around the mean of the true scores. This suggests that the model was able to capture information from the EEG inputs, even though it struggled to fully match the amplitude of the true scores.

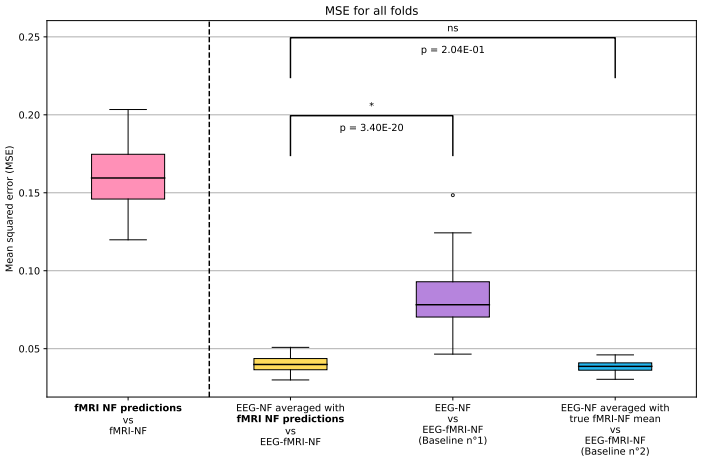
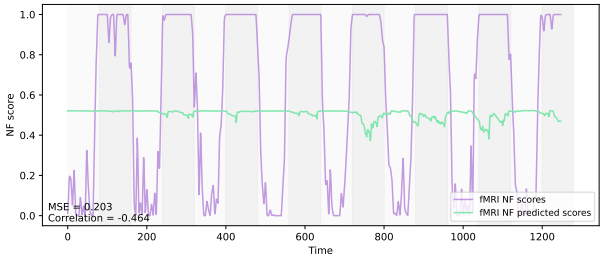


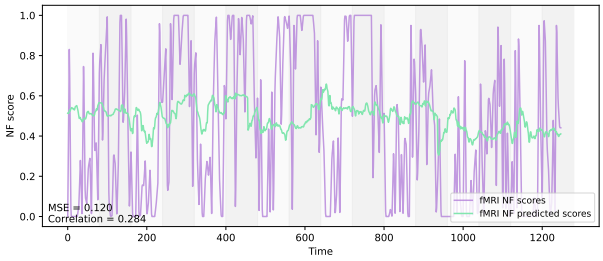
Figure 20: **Results for all test subjects across all folds using the mean squared error (MSE) metric for the CNN with extracted features samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

As described in section 3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 22 presents the predictions for the same subjects used in Figure 21, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a) with a value of 0.051, the true bi-modal EEG-fMRI NF scores exhibit a problematic pattern, with many plateaus at 0.5 during the task blocks. This indicates that when the true fMRI NF scores were at 1 (as seen in the preceding figure), the EEG NF scores were at 0. This discrepancy suggests potential issues with the quality of the EEG NF scores or possibly the EEG signals themselves. As the model's predictions in this case were almost flat, the final result show clearly the EEG NF scores with slight variations introduced by the fMRI NF

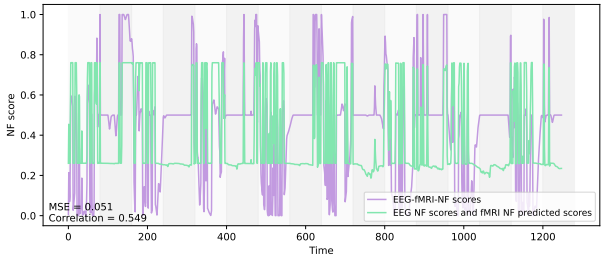


(a) Prediction for sub-xp211 run 3

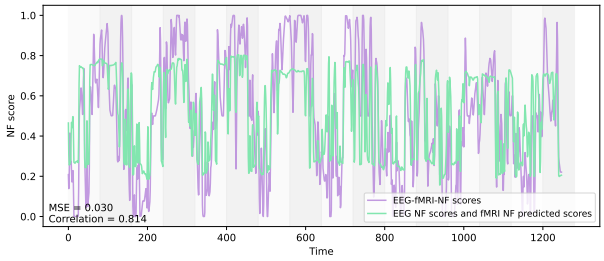


(b) Prediction for sub-xp221 run 3

Figure 21: **Examples of predictions made using a CNN model with extracted features samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp211 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp221 run 3, representing the lowest error across all folds.



(a) Prediction for sub-xp211 run 3



(b) Prediction for sub-xp221 run 3

Figure 22: **Examples of final results made using a CNN model with extracted features samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

predictions. The fact that it is a mean leads to reduced amplitude in this case.

For the lowest MSE example (b), the true bi-modal EEG-fMRI NF scores align more closely with the expected rest/task trend, though they are not entirely ideal, especially towards the end of the run. The final result here seems to track the true scores well. The mean MSE of 0.030 reflects this closer alignment, though the model still struggles to fully match the true scores' amplitude. These examples further highlight the influence of the quality of EEG NF scores, likely reflecting the quality of EEG signals, in achieving good prediction performance.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 23 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are the same as the previous sections. In summary, a vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 appears to be a clear outlier.

Now, regarding the MSE between predicted and true fMRI NF scores, we observe a general trend where a higher t-statistic is associated with a lower MSE. For instance, the highest MSE example run we observed earlier, from sub-xp211, corresponds to the lowest t-statistic value (around -10). While the lowest MSE example run (from sub-xp221) does not have the absolute highest t-statistic value, it still ranks among the highest (around 21). However, there are a few counterexamples: for instance, the run from sub-xp204 in the bottom left corner shows a very low MSE despite a t-statistic close to 0. Conversely, the runs from sub-xp221 and sub-xp216 in the top right corner have t-statistics close to 25 but very high MSEs. This indicates that good predictive performance can sometimes occur even when the neurofeedback response and signal quality (as measured by the t-statistic) are weak. To conclude, the correlation coefficient value of -0.358 suggests that, while there seems

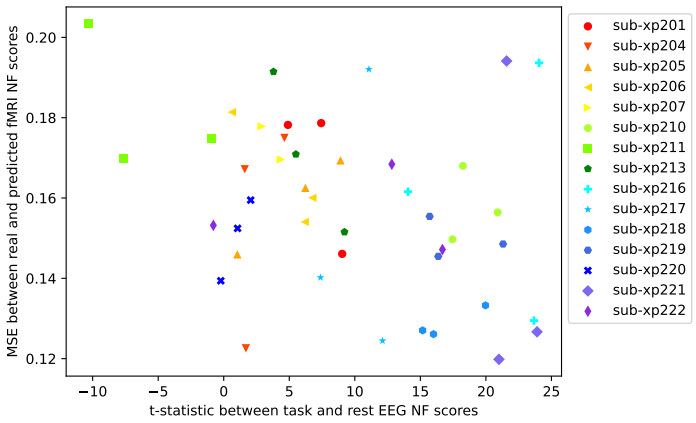


Figure 23: **Analysis for the CNN approach with extracted features samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.358.

to be a link between the t-statistic and MSE, it is not the only factor influencing the model's performance.

4.5 CNN model with raw signal samples as inputs

In this final section, we present the results of our method applied to the 1D CNN type with raw signal-based input data. After running the genetic algorithm, the architecture hyperparameters found include four convolutional layers, starting with 128 filters in the first layer, all using a kernel size of 3. The dense layer contains 64 neurons. Regularization values were set to 0.01. Additionally, a spatial dropout rate of 0.6 was used for the convolutional layers, while the dense layer had a dropout rate of 0.4. An illustration of this network is available in Figure 24.

Similarly to the LSTM approach, the architecture identified through the genetic search for raw signal inputs is larger than the one found for extracted features inputs. The presence of four convolutional layers, with the first one already having 128 filters and subsequent layers doubling that number, makes this a relatively large model. Notably, the dense layer has the same number of neurons as in the previous section, which is modest in this context. This suggests that the model may have "won" the genetic search by striking a balance: it has substantial capacity for extracting meaningful patterns in the convolutional layers, while the relatively small dense layer helps prevent overfitting. Regularization also plays a critical role here, with average values for kernel regularizers and dropout rates indicating that the network required it to perform well. The slight increase in error, 0.166, compared to the extracted features data model (0.159), supports the interpretation given for the LSTM approach that this model is more complex because it faces greater challenges in extracting meaningful patterns directly from the raw signal inputs.

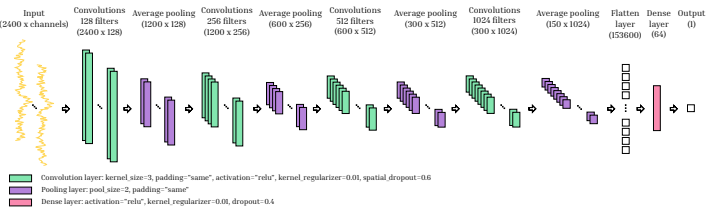


Figure 24: **Architecture of the CNN found through the genetic search, using raw signal samples as input.**

Figure 25 displays the loss curves for the 15 CNN models trained on raw signal-based samples, as described in the preceding sections.

The near-complete overlap of the training and validation loss curves throughout the training process signifies that the model's learning was highly consistent across both datasets. This close alignment is an indicator that the model is generalizing well to unseen data, as there is little to no sign of overfitting. However, the flatness of both curves towards the end of training suggests that the model reached a plateau where further training did not lead to significant improvements. This stability could either indicate that the model quickly learned the patterns in the data, or that there is limited information in the data, making additional epochs unlikely to yield further performance gains. The overall short number of epochs, ranging between 38 and 89 before early stopping, suggests that the model quickly reached its optimal performance. In summary, the CNN model with raw signal inputs demonstrates an efficient training process, characterized by rapid convergence and stable overlapping loss curves.

Same as preceding sections, Figure 26 presents four key comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1692(\pm 0.0299)$. Sec-

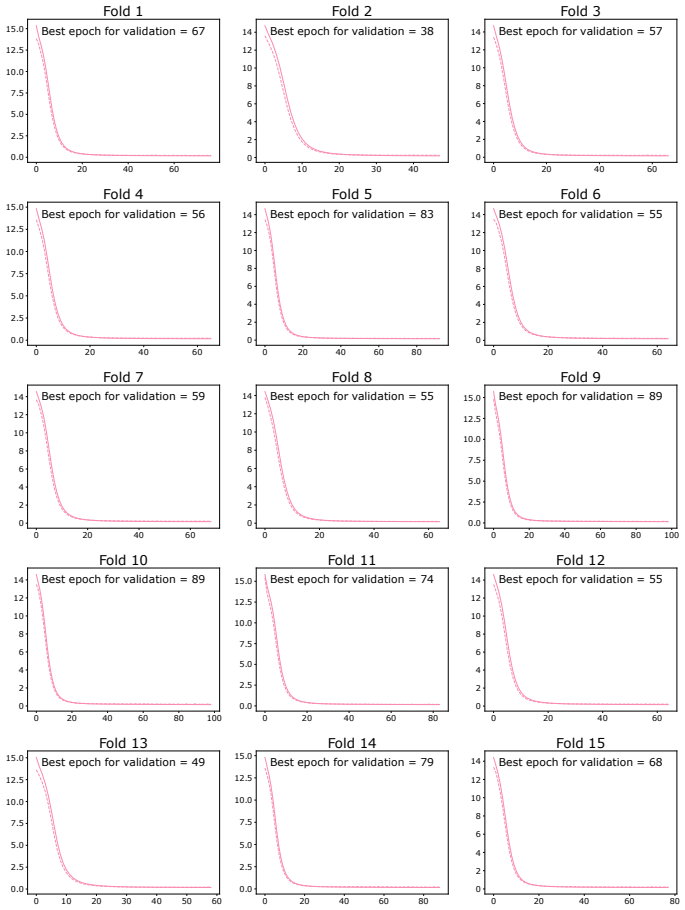


Figure 25: **Learning curves for the 15 folds of the CNN approach with raw signal samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

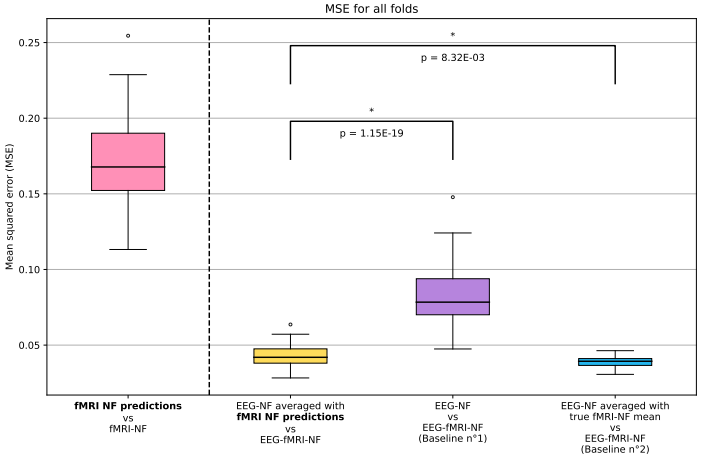


Figure 26: **Results for all test subjects across all folds using the mean squared error (MSE) metric for the CNN with raw signal samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

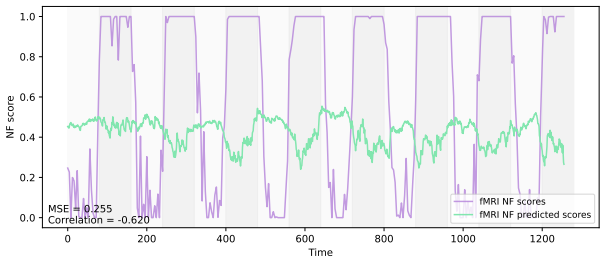
ondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0423(\pm 0.0075)$. Thirdly, for baseline n°1 (purple boxplot): the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores is $0.0831(\pm 0.0196)$. This fourth approach also significantly ($p = 1.15e-19 < 0.05$ using a paired t-test) outperforms baseline n°1. However, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0388(\pm 0.0037)$. Our predictions averaged with EEG NF scores are thus significantly ($p = 8.32e-3 < 0.05$ using a paired t-test) less accurate than the mean of true fMRI scores averaged with EEG NF scores.

To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 27 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp211 run 3, which represents the highest error, and sub-xp210 run 3, which represents the lowest error.

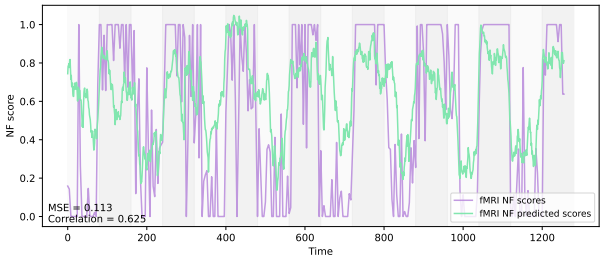
For the highest MSE example (a) with a value of 0.255, we observe that it is the same subject and run as in the previous section. Once again, the true fMRI NF scores are remarkably regular and clean. Despite this, the model's predictions are only somewhat centered around the mean of the true scores and often move in the opposite direction during both task and rest blocks. This outcome further supports the notion raised in the previous section: even when the true scores are nearly ideal, the model struggles, likely due to issues related to the EEG signal inputs rather than the true fMRI NF scores themselves.

For the lowest MSE example (b), the true fMRI NF scores follow the expected rest/task trend with some slight spikiness. The model's predictions appear to be the best among all section examples, aligning well with the true scores and showing good amplitude, resulting in a mean MSE of 0.113.

As described in section 3.3.2, we combined the EEG



(a) Prediction for sub-xp211 run 3



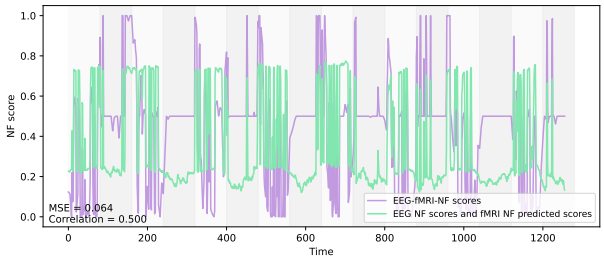
(b) Prediction for sub-xp210 run 3

Figure 27: **Examples of predictions made using a CNN model with raw signal samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp211 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp210 run 3, representing the lowest error across all folds.

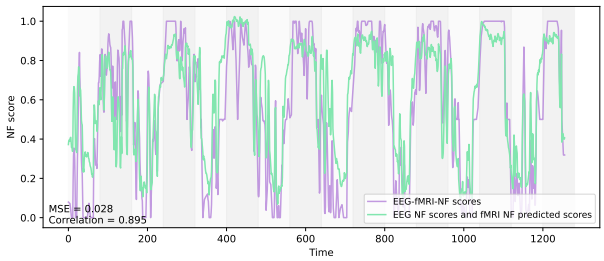
NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 28 presents the predictions for the same subjects used in Figure 27, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores display, as in the previous section, significant flat sections at 0.5 during the task blocks. This flatness suggests again that when the true fMRI NF scores were at 1, the EEG NF scores were at 0, indicating potential issues with the EEG NF scores and even EEG signals. As the model's predictions in this case were somewhat flat, except during the task blocks where they tended to move in the opposite direction of the true scores, the final outcome resembles the EEG NF scores but with reduced amplitude and further misalignment during task blocks. Consequently, this leads to a high mean MSE of 0.064.

In contrast, the lowest MSE example (b) shows true bi-modal EEG-fMRI NF scores that follow the expected rest/task trend quite well. The model's predictions in this case were well-aligned with the true scores, resulting in a final output that exhibits good amplitude and accurately captures the overall trend. This example has a mean MSE of 0.028, the lowest across all sections. The close match between the final result and the true bi-modal scores in this instance may be due to the model's ability to generalize well when the input signals are of higher quality.



(a) Prediction for sub-xp211 run 3



(b) Prediction for sub-xp210 run 3

Figure 28: **Examples of final results made using a CNN model with raw signal samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 29 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are the same as the previous sections. In summary, a vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 appears to be a clear outlier.

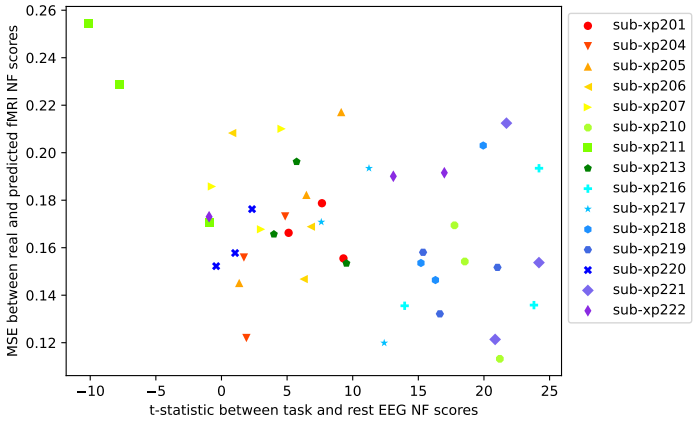


Figure 29: **Analysis for the CNN approach with raw signal samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.365.

Regarding the MSE between predicted and true fMRI NF scores, we observe the same trend as the previous section, where a higher t-statistic is generally associated with a lower MSE. For example, the highest MSE run we observed earlier, from sub-xp211, corresponds to the lowest t-statistic value (around -10). While the lowest MSE run (from sub-xp210) does not have the absolute highest t-statistic value, it still ranks among the highest (around 21). As with the previous section, there are a few counterexamples. For instance, the run from sub-xp204 in the bottom left corner shows a very low MSE despite having a t-statistic close to 0. Conversely, the runs from sub-xp221, sub-xp218, and sub-xp216 in the top right corner have t-statistics between 20 and 25 but quite high MSEs. It is interesting to note that the correlation coefficient (-0.365) and the outliers are similar to those in the previous section, despite differences in the architecture and the formatting of the input data. This consistency suggests that, at least for the CNN approaches, some additional factors influencing the results may be linked to inherent information within the raw EEG data and true fMRI NF scores. We will discuss it a bit further in the next section.

5. Discussion

The use of a genetic algorithm was primarily motivated by the complexity of our application, mainly due to the nature of our input and output data. Despite thorough preprocessing, EEG input data remains noisy, influenced both externally by MRI recording conditions and internally by the brain's electrical activity. Additionally, as previously mentioned, although EEG and fMRI measure the same physical phenomenon, the relationship between these two modalities remain indirect and not well understood. This lack of prior expertise made it difficult to select an appropriate model architecture. Moreover, on a more conceptual level, we wanted to avoid proposing an architecture based on arbitrary choices without concrete justification. As a result, the genetic algorithm method allowed us to automatically move closer to an optimal architecture.

The main limitation of this approach is the computation time. Each individual in the genetic population requires a model to be trained, which is time-consuming. A larger population size and a greater number of generations would likely result in an architecture closer to the theoretical global optimum. Moreover, the long computation time also influenced the initial choices we made. For instance, we limited the number of convolutional layers in the 1D CNN to 4 because a more complex model would take too long to train, and we aimed to limit the genetic search to about one week per case. As a result, more ambitious architectures were not explored using this method. However, we can take some reassurance from the fact that we had previously tested larger model architectures outside of the genetic algorithm framework, which did not yield better performance. This allowed us to empirically choose the initial pre-selected values for the genetic search.

Initially, we expected LSTMs to outperform 1D CNNs in predicting fMRI NF scores due to their reputed strength in handling sequential data. However, our experiments revealed that 1D CNNs and LSTMs were similar in terms of performance, with 1D CNN outperforming LSTM only within the extracted features approach. One possible explanation for why LSTMs did not outperform 1D CNNs could be the limited quantity and representativity of our dataset, which included only 15 subjects with 3 runs each. LSTMs typically excel with datasets that exhibit clear trends, such as those found in weather forecasting or sales predictions. However, the EEG data, particularly in simultaneous EEG-fMRI acquisitions, may have been too challenging for the LSTM to handle effectively. Additionally, complex architectures like LSTMs may struggle to generalize well to unseen data when working with a dataset that is not sufficiently representative. This could explain why a simpler architecture, like the 1D CNN, slightly outperformed LSTMs in this particular application.

On the other hand, we anticipated that the extracted features samples approach would yield better results than the raw signal samples approach, and our experiments confirmed this for the CNN configurations. Extracting bandpowers in the alpha and beta ranges from the raw signals appears to assist the models during the feature extraction phase of the network done in the convolutional layers. In

contrast, using raw signals directly adds complexity for the model, as it essentially requires an additional step of extracting meaningful features. The reasoning behind trying raw signals was to simplify the pipeline for real-life applications, where the model could be used directly on raw signals from the subject to predict fMRI NF scores. Additionally, it served to test the neural networks' core strength in autonomously extracting interesting features. However, based on our results, incorporating an extra processing step to compute bandpowers before using the model in a neurofeedback protocol is a viable option, as it is not a costly operation at all.

Overall, developing a model that can predict fMRI NF scores from EEG signals for any subject (what we refer to as a global model, as opposed to an individual model) is challenging. The preceding work (Cury et al., 2020b), from which this research stems, involved individualized sparse regression models designed to exploit EEG data alone to predict fMRI NF scores. That work demonstrated that such an endeavor seemed possible. However, it is difficult to directly compare our results with this previous work due to our focus on developing a global model rather than one model per subject. Nevertheless, the earlier study showed that a simple machine learning approach with sparse models could achieve promising results, outperforming what we called baseline n°1 in this work. This motivated us to attempt creating a global model, which would be more practical in a clinical setup.

This exploration was particularly challenging due to the stark differences between the two modalities. Nonetheless, all our models outperformed baseline n°1, where our predictions, when averaged with EEG NF scores, were closer in terms of MSE to the true bi-modal EEG-fMRI NF scores than EEG NF scores alone. This indicates that we successfully enhanced the EEG NF scores. However, the mean correlation between our fMRI NF predictions and the true fMRI NF scores was quite low, and the shapes of some result runs did not perfectly match. It could be interpreted that the EEG NF scores and fMRI NF scores are so distinct that even predicting something vaguely close to the fMRI NF score was enough to surpass baseline n°1. The LSTM with raw signal-based samples presented an interesting case, producing almost flat predictions with a mean MSE similar to those with more amplitude. This made us consider another baseline, referred to as baseline n°2. As we observed, the models did not outperform baseline n°2, indicating that the models provided at best fMRI predictions that had as much error with the true fMRI NF scores as the mean of these true scores. Since these fMRI predictions are derived from an entirely different modality (EEG), this remains a interesting result, but there is still much to understand and improve.

To begin with areas for improvement, let's talk about neural networks. While re-framing the problem could enable the use of fine-tuning to create individualized models, we will focus here on improving global models. The current trend in deep learning leans towards larger architectures, which require a significant amount of data to train. This could still be a viable path if we can acquire more data, either through extensive data collection efforts or by pursuing data sharing and open data initiatives. However, since the simpler 1D CNN models in our study performed slightly better, it is worth considering a contradicting idea in the field: exploring smaller models. These models require less data to train and are less prone to overfitting, making them also an interesting direction for further investigation.

Next, regarding the data itself: there is an understanding that comes after working in the machine learning field for a short while that significant performance improvements often come not from changing the model, but from cleaning, organizing, and understanding the data. With simultaneous EEG-fMRI acquisitions, there is certainly correlated noise (especially when the subject is moving, which is inevitable) between the electrodes, that may have been captured by the model. To our knowledge, there are currently no methods capable of fully correcting these residual noises, highlighting a need for further research in this area.

It would also be valuable to better understand the EEG signal inputs. The t-statistic measure of EEG NF scores, which we used as an approximate indicator of EEG signal quality (as well as neurofeedback response), revealed significant variability between participants. While this variability is advantageous for training a global model with representativity, it also makes the relationship between inputs and scores more challenging for the model to learn. In our experiments, it might have been worth considering the exclusion of sub-xp211, as an outlier like this could potentially confuse the model during training. Although to be precise, we can note that the training losses and results from fold 6, where this subject was in the validation dataset and therefore not used for training or testing, remained consistent with other folds.

Overall, this work provided in-depth exploration of the possibility of predicting fMRI information for any subject using EEG signals acquired from multiple participants. Moving forward, we believe that performance enhancements may still be achieved by developing another modeling approach, but more importantly, by gaining a deeper understanding of the data. Improving noise correction techniques for EEG signals and better characterizing variability between subjects could lead to more robust models.

6. Conclusion

We have presented a method for searching model architecture hyperparameters using a genetic algorithm in the con-

text of predicting fMRI NF scores from EEG signals. This method is flexible and can be easily adapted to different model types. We used our genetic algorithm to converge towards four configurations, using LSTM and CNN types, both with two different data formats: extracted feature-based samples and raw signal-based samples. The CNN with extracted features approach demonstrated slightly superior performance in terms of mean squared error (MSE) compared to the other tested architectures. Our results showed that fMRI NF predictions, when averaged with EEG NF scores, align significantly closer to the true bi-modal EEG-fMRI NF scores than the EEG NF scores alone. This approach can enrich EEG NF scores by incorporating fMRI predictions derived from EEG, offering an improved NF score that leverages multi-modal information. However, our fMRI NF predictions have at best the same MSE with true fMRI NF scores as the mean of these true scores. So, we believe that the models developed using this method are not yet suitable for unimodal EEG neurofeedback applications and still require further improvements.

Acknowledgments

MRI data acquisition was performed at the Neurinfo MRI research facility from the University of Rennes, University Hospital of Rennes, Inria, CNRS and the Rennes Cancer Center. Neurinfo is also supported by the the Brittany Council, Rennes Metropole and GIS IBISA

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript, following all applicable laws and regulations regarding treatment of animals or human subjects. After approval from the Institutional Review Board, the volunteers were included in the OSS IRM (https://clinicaltrials.gov/ct2/show/NCT03440983) study, supported by the University Hospital of Rennes and the University of Rennes.

Conflicts of Interest

We declare we don't have conflicts of interest.

Data availability

The pseudonymized data are available in BIDS format on the OpenNeuro platform at `https://openneuro.org/datasets/ds002338`. It is described in Lioi et al. (2020) as the XP2 protocol.

Code availability

The code developed for this research is available on Gitlab Inria at `https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals`. Additionally, the code has been archived with Software Heritage to ensure long-term preservation at `https://archive.softwareheritage.org/swh:1:dir:a651db8d1934543cac321a1723cdf404348e2156;origin=https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals;visit=swh:1:snp:7004a52e6d9c3e956d761b71bdb3ec6e90a99d8e;anchor=swh:1:rev:f62cd5eff14180de3167bea085892e34a0f3a517`. All implementations were made on an Nvidia RTX A3000 GPU. Since the data used to train the models are open, we share the trained models weights for all configurations and all folds in the same repository as the code.

References

Rodolfo Abreu, Alberto Leal, and Patrícia Figueiredo. Eeg-informed fmri: a review of data analysis methods. *Frontiers in human neuroscience*, 12:29, 2018.

PG Benardos and G-C Vosniakos. Optimizing feedforward artificial neural network architecture. *Engineering applications of artificial intelligence*, 20(3):365–382, 2007.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Melanie Boly, Olivia Gosseries, Marcello Massimini, and Mario Rosanova. Functional neuroimaging techniques. In *The Neurology of Conciousness*, pages 31–47. Elsevier, 2016.

Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7):1636, 2018.

Hyejung Chung and Kyung-shik Shin. Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. *Neural Computing and Applications*, 32(12):7897–7914, 2020.

Giuseppina Ciccarelli, Giovanni Federico, Giulia Mele, Angelica Di Cecca, Miriana Migliaccio, Ciro Rosario Ilardi, Vincenzo Alfano, Marco Salvatore, and Carlo Cavaliere. Simultaneous real-time eeg-fmri neurofeedback: A systematic review. *Frontiers in Human Neuroscience*, 17: 1123014, 2023.

Aslıhan Cura, Haluk Küçük, Erdem Ergen, and İsmail Burak Öksüzoğlu. Driver profiling using long short term memory (lstm) and convolutional neural network (cnn) methods. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6572–6582, 2020.

Claire Cury, Giulia Lioi, Lorraine Perronnet, Anatole Lécuyer, Pierre Maurel, and Christian Barillot. Impact of 1d and 2d visualisation on eeg-fmri neurofeedback training during a motor imagery task. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1018–1021. IEEE, 2020a.

Claire Cury, Pierre Maurel, Rémi Gribonval, and Christian Barillot. A sparse eeg-informed fmri model for hybrid eeg-fmri neurofeedback prediction. *Frontiers in neuroscience*, 13:1451, 2020b.

Jan C de Munck, Sonia I Gonçalves, L Huijboom, Joost PA Kuijer, Petra JW Pouwels, Rob M Heethaar, and FH Lopes da Silva. The hemodynamic response of the alpha rhythm: an eeg/fmri study. *Neuroimage*, 35(3): 1142–1151, 2007.

C Erden. Genetic algorithm-based hyperparameter optimization of deep learning models for pm2. 5 time-series prediction. *International Journal of Environmental Science and Technology*, 20(3):2959–2982, 2023.

Nikolaos Gorgolis, Ioannis Hatzilygeroudis, Zoltan Istenes, and Lazlo-Grad Gyenne. Hyperparameter optimization of lstm network models through genetic algorithm. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–4. IEEE, 2019.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

James V Hansen, James B McDonald, and Ray D Nelson. Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models. *Computational Intelligence*, 15(3): 171–184, 1999.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551, 1989.

Giulia Lioi, Claire Cury, Lorraine Perronnet, Marsel Mano, Elise Bannier, Anatole Lécuyer, and Christian Barillot. Simultaneous eeg-fmri during a neurofeedback task, a brain imaging dataset for multimodal data integration. *Scientific data*, 7(1):173, 2020.

Cesare Magri, Ulrich Schridde, Yusuke Murayama, Stefano Panzeri, and Nikos K Logothetis. The amplitude and timing of the bold signal reflects the relationship between local field potential power at different frequencies. *Journal of Neuroscience*, 32(4):1395–1407, 2012.

Marsel Mano, Anatole Lécuyer, Elise Bannier, Lorraine Perronnet, Saman Noorzadeh, and Christian Barillot. How to build a hybrid neurofeedback platform combining eeg and fmri. *Frontiers in neuroscience*, 11:140, 2017.

Felipe P Marinho, Paulo AC Rocha, Ajalmar RR Neto, and Francisco DV Bezerra. Short-term solar irradiance forecasting using cnn-1d, lstm, and cnn-lstm deep neural networks: A case study with the folsom (usa) dataset. *Journal of Solar Energy Engineering*, 145(4):041002, 2023.

Geoffrey F Miller, Peter M Todd, and Shailesh U Hegde. Designing neural networks using genetic algorithms. In *ICGA*, volume 89, pages 379–384, 1989.

Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences*, 484:302–337, 2019.

Lorraine Perronnet, Anatole Lécuyer, Marsel Mano, Elise Bannier, Fabien Lotte, Maureen Clerc, and Christian Barillot. Unimodal versus bimodal eeg-fmri neurofeedback of a motor imagery task. *Frontiers in Human Neuroscience*, 11:193, 2017.

Lorraine Perronnet, Anatole Lécuyer, Marsel Mano, Mathis Fleury, Giulia Lioi, Claire Cury, Maureen Clerc, Fabien Lotte, and Christian Barillot. Learning 2-in-1: towards integrated eeg-fmri-neurofeedback. *BioRxiv*, page 397729, 2018.

Galina V Portnova, Alina Tetereva, Vladislav Balaev, Mikhail Atanov, Lyudmila Skiteva, Vadim Ushakov, Alexey Ivanitsky, and Olga Martynova. Correlation of bold signal with linear and nonlinear patterns of eeg in resting state eeg-informed fmri. *Frontiers in human neuroscience*, 11:654, 2018.

Tian Renton, Alana Tibbles, and Jane Topolovec-Vranic. Neurofeedback as a form of cognitive rehabilitation therapy following stroke: A systematic review. *PloS one*, 12 (5):e0177290, 2017.

René Scheeringa, Pascal Fries, Karl-Magnus Petersson, Robert Oostenveld, Iris Grothe, David G Norris, Peter Hagoort, and Marcel CM Bastiaansen. Neuronal dynamics underlying high-and low-frequency eeg oscillations contribute independently to the human bold signal. *Neuron*, 69(3):572–583, 2011.

Ranganatha Sitaram, Tomas Ros, Luke Stoeckel, Sven Haller, Frank Scharnowski, Jarrod Lewis-Peacock, Nikolaus Weiskopf, Maria Laura Blefari, Mohit Rana, Ethan Oblak, et al. Closed-loop brain training: the science of neurofeedback. *Nature Reviews Neuroscience*, 18(2): 86–100, 2017.

Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. Automatically designing cnn architectures using the genetic algorithm for image classification. *IEEE transactions on cybernetics*, 50(9):3840–3854, 2020.

Tianlu Wang, Dante Mantini, and Celine R Gillebert. The potential of real-time fmri neurofeedback for stroke rehabilitation: A systematic review. *cortex*, 107:148–165, 2018.

Lingxi Xie and Alan Yuille. Genetic cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1379–1388, 2017.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Vadim Zotev, Raquel Phillips, Han Yuan, Masaya Misaki, and Jerzy Bodurka. Self-regulation of human brain activity using simultaneous real-time fmri and eeg neurofeedback. *NeuroImage*, 85:985–995, 2014.