



Deep Learning-Based Approaches for Decoding Motor Intent From Peripheral Nerve Signals

Diu K. Luu^{1†}, Anh T. Nguyen^{1,2†}, Ming Jiang³, Jian Xu¹, Markus W. Drealan¹, Jonathan Cheng^{4,5}, Edward W. Keefer⁵, Qi Zhao^{3*} and Zhi Yang^{1,2*}

¹ Department of Biomedical Engineering, University of Minnesota, Minneapolis, MN, United States, ² Fasikl Incorporated, Minneapolis, MN, United States, ³ Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, United States, ⁴ Department of Plastic Surgery, University of Texas Southwestern Medical Center, Dallas, TX, United States, ⁵ Nerves Incorporated, Dallas, TX, United States

OPEN ACCESS

Edited by:

Silvia Muceli,
Chalmers University of Technology,
Sweden

Reviewed by:

Daniilo Pani,
University of Cagliari, Italy
Sigrid Dupan,
University of Edinburgh,
United Kingdom

*Correspondence:

Qi Zhao
qzhao@umn.edu
Zhi Yang
yang5029@umn.edu

[†]These authors share first authorship

Specialty section:

This article was submitted to
Neuroprosthetics,
a section of the journal
Frontiers in Neuroscience

Received: 15 February 2021

Accepted: 20 May 2021

Published: 23 June 2021

Citation:

Luu DK, Nguyen AT, Jiang M, Xu J, Drealan MW, Cheng J, Keefer EW, Zhao Q and Yang Z (2021) Deep Learning-Based Approaches for Decoding Motor Intent From Peripheral Nerve Signals. *Front. Neurosci.* 15:667907. doi: 10.3389/fnins.2021.667907

Previous literature shows that deep learning is an effective tool to decode the motor intent from neural signals obtained from different parts of the nervous system. However, deep neural networks are often computationally complex and not feasible to work in real-time. Here we investigate different approaches' advantages and disadvantages to enhance the deep learning-based motor decoding paradigm's efficiency and inform its future implementation in real-time. Our data are recorded from the amputee's residual peripheral nerves. While the primary analysis is offline, the nerve data is cut using a sliding window to create a "pseudo-online" dataset that resembles the conditions in a real-time paradigm. First, a comprehensive collection of feature extraction techniques is applied to reduce the input data dimensionality, which later helps substantially lower the motor decoder's complexity, making it feasible for translation to a real-time paradigm. Next, we investigate two different strategies for deploying deep learning models: a one-step (1S) approach when big input data are available and a two-step (2S) when input data are limited. This research predicts five individual finger movements and four combinations of the fingers. The 1S approach using a recurrent neural network (RNN) to concurrently predict all fingers' trajectories generally gives better prediction results than all the machine learning algorithms that do the same task. This result reaffirms that deep learning is more advantageous than classic machine learning methods for handling a large dataset. However, when training on a smaller input data set in the 2S approach, which includes a classification stage to identify active fingers before predicting their trajectories, machine learning techniques offer a simpler implementation while ensuring comparably good decoding outcomes to the deep learning ones. In the classification step, either machine learning or deep learning models achieve the accuracy and F1 score of 0.99. Thanks to the classification step, in the regression step, both types of models result in a comparable mean squared error (MSE) and variance accounted for (VAF) scores as those of the 1S approach. Our study outlines the trade-offs to inform the future implementation of real-time, low-latency, and high accuracy deep learning-based motor decoder for clinical applications.

Keywords: convolutional neural network, deep learning, feature extraction, motor decoding, neuroprosthesis, neural decoder, peripheral nerve interface, recurrent neural network

1. INTRODUCTION

Upper-limb amputation affects the quality of life and well-being of millions of people in the United States, with hundreds of thousand new cases annually (Ziegler-Graham, 2008). Neuroprosthetic systems promise the ultimate solution by developing human-machine interfaces (HMI) that could allow amputees to control robotic limbs using their thoughts (Harris, 2011; Johannes, 2011; Schultz, 2011; Cordella, 2016). It is achieved by decoding the subject's motor intent with neural data acquired from different parts of the nervous system. Proven approaches include surface electromyogram (EMG) (Sebelius, 2005; Fougner, 2012; Jiang, 2012; Amsuss, 2013; Zuleta, 2019; George, 2020a,b), electroencephalogram (EEG) (Hu, 2015; Zeng, 2015; Sakhavi, 2018; Kwon, 2019), cortical recordings (Mollazadeh, 2011; Hochberg, 2012; Irwin, 2017), and peripheral nerve recordings (Micera, 2011; Davis, 2016; Vu, 2017, 2020; Wendelken, 2017; Zhang, 2017; Nguyen and Xu, 2020).

However, implementing an effective HMI for neuroprostheses remains a challenging task. The decoder should be able to predict the subject's motor intents accurately and satisfy certain criteria to make it practical and useful in daily lives (Vujaklija, 2017; Krasoulis, 2019). Some criteria include *dexterity*, i.e., controlling multiple degrees-of-freedom (DOF) such as individual fingers; *intuitiveness*, i.e., reflecting the true motor intent in mind, and *real-time*, i.e., having minimal latency from thoughts to movements.

In recent years, deep learning techniques have emerged as strong candidates to overcome this challenge thanks to their ability to process and analyze biological big data (Mahmud, 2018). Our previous work (Nguyen and Xu, 2020) shows that neural decoders based on the convolutional neural network (CNN) and recurrent neural network (RNN) architecture outperform other "classic" machine learning counterparts in decoding motor intents from peripheral nerve data obtained with an implantable bioelectric neural interface. The deep learning-based motor decoders can regress the intended motion of 15 degrees-of-freedom (DOF) simultaneously, including flexion/extension and abduction/adduction of individual fingers state-of-the-art performance metrics, thus complying with the dexterity and intuitiveness criteria.

Here we build upon the foundation of Nguyen and Xu (2020) by exploring different strategies to optimize the motor decoding paradigm's efficiency. The aim is to lower the neural decoder's computational complexity while retaining high accuracy predictions to make it feasible to translate the motor decoding paradigm to real-time operation suitable for clinical applications, especially when deploying in a portable platform. This paper does not aim to solve the real-time problem but to study different approaches, highlighting their advantages and disadvantages, which will inform future implementation of the motor decoding paradigm in real-time.

First, we utilize feature extraction to reduce data dimensionality. By examining the data spectrogram, we learn that most of the signals' power concentrates in the frequency band 25–600 Hz (Nguyen and Xu, 2020). Many feature extraction techniques (Zardoshti-Kermani, 1995; Phinyomark, 2009, 2012;

Rafiee, 2011) have been developed to handle signals with similar characteristics. Feature extraction aims not only to amplify the crucial information, lessen the noise but also to substantially reduce the data dimensionality before feeding them to deep learning models. This could simultaneously enhance the prediction accuracy and lower the deep learning models' complexity. Here we focus on a comprehensive list of 14 features that consistently appear in the field of neuroprosthesis.

Second, we explore two different strategies for deploying deep learning models: the two-step (2S) and the one-step (1S) approaches. The 2S approach consists of a classification stage to identify the active fingers and a regression stage to predict the trajectories of digits in motion. The 1S approach only has one regression stage to predict the trajectories of all fingers concurrently. In practice, the 2S approach should be marginally more efficient because not all models are inferred at a given moment. The models in the 2S approach are only trained on the subset where a particular finger is active, while all models in the 1S approach are trained on the full dataset. Here we focus on exploring the trade-offs between two approaches to inform future decisions of implementing the deep learning-based motor decoder in real-world applications.

The rest of this paper is organized as follows: Section "Data Description" introduces the human participant of this research, the process of collecting input neural signals from the residual peripheral nerves of the participant, and establishing the ground-truth for the motor decoding paradigm using deep learning models. Section "Data Preprocessing" elaborates on how to cut raw input neural data into trials and extract their main features in the temporal domain before feeding to deep learning decoding models. Section "Proposed Deep Learning Models and Decoding Strategies" discusses the two approaches to efficiently translate motor intent from the residual peripheral nerves of the participant into motor control of the prosthesis as well as the architecture and the hyper-parameters of the deep learning models used in each approach. Section "Experimental Setup" is about the three machine learning models used as the baseline and how input neural data are allocated to the training and validation set. Section "Metrics and Results" presents the metrics to measure the performance of all models used in the motor intent decoding process and discusses the main results of both proposed approaches. Section "Discussion" discusses the role of feature extraction in reducing the deep learning motor decoders' complexity for real-time applications, how to further apply it in future works, and the advantages of machine learning and deep learning motor decoders in different scenarios where input dataset's size varies. Finally, section "Conclusion" summarizes the main contributions of this paper.

2. DATA DESCRIPTION

2.1. Human Participant

The human experiment is a part of the clinical trial DEXterous Hand Control Through Fascicular Targeting (DEFT), which is sponsored by the DARPA Biological Technologies Office as part of the Hand Proprioception and Touch Interfaces (HAPTIX)

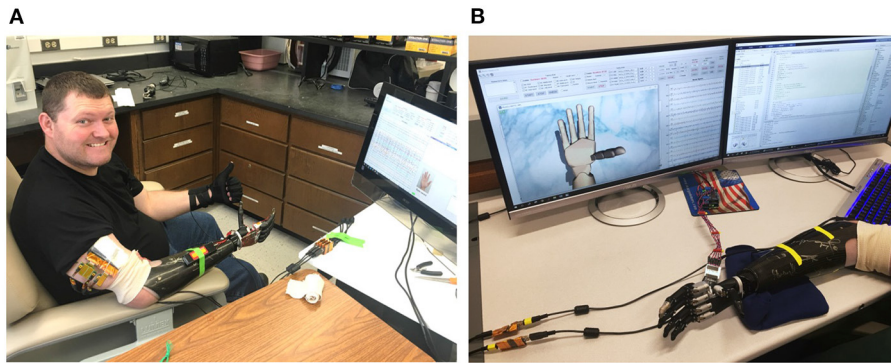


FIGURE 1 | Photo of the (A) amputee and (B) the data collection software during a training session. The patient performs various hand movements repeatedly during the training session. Nerve data and ground-truth movements are collected by a computer and displayed in real-time on the monitor for comparison.

program, identifier No. NCT02994160¹. The human experiment protocols are reviewed and approved by the Institutional Review Board (IRB) at the University of Minnesota (UMN) and the University of Texas Southwestern Medical Center (UTSW). The amputee voluntarily participates in our study and is informed of the methods, aims, benefits, and potential risks of the experiments prior to signing the Informed Consent. Patient safety and data privacy are overseen by the Data and Safety Monitoring Committee (DSMC) at UTSW. The implantation, initial testing, and post-operative care are performed at UTSW by Dr. Cheng and Dr. Keefer, while motor decoding experiments are performed at UMN by Dr. Yang's lab. The clinical team travels with the patient in each experiment session. The patient also completes the Publicity Agreements where he agrees to be publicly identified, including showing his face.

The participant is a transradial male amputee who has lost his hand for over 5 years (**Figure 1**). Among seven levels of upper-limb amputation, transradial is the most common type that accounts for about 57% of upper-limb loss in the U.S. (Schultz, 2011; Cordella, 2016). Like most amputees, the subject still has phantom limb movements; however, such phantom feelings fade away over time. By successfully decoding neural signals from the residual nerves of an amputee who has lost his limb for a long time, we would offer a chance to regain upper-limb motor control for those who are sharing the same conditions.

The patient undergoes an implant surgery where four longitudinal intrafascicular electrode (LIFE) arrays are inserted into the residual median and ulnar nerves using the microsurgical fascicular targeting (FAST) technique (**Figure 2**). The electrode array' design, characteristics, and surgical procedures are reported in Cheng (2017) and Overstreet (2019). The patient has the electrode arrays implanted for 12 months, during which the conditions of the implantation site is regularly monitored for signs of degradation.

The patient participates in several neural stimulations, neural recording, and motor decoding experiment sessions. He initially has weak phantom limb movements due to reduced motor

control signals in the residual nerves throughout the years. However, the patient reports that the more experiment sessions he takes part in, the stronger his phantom control and sensation of the lost hand become. This suggests that training may help re-establish the connection between the motor cortex and the residual nerves, resulting in better motor control signals.

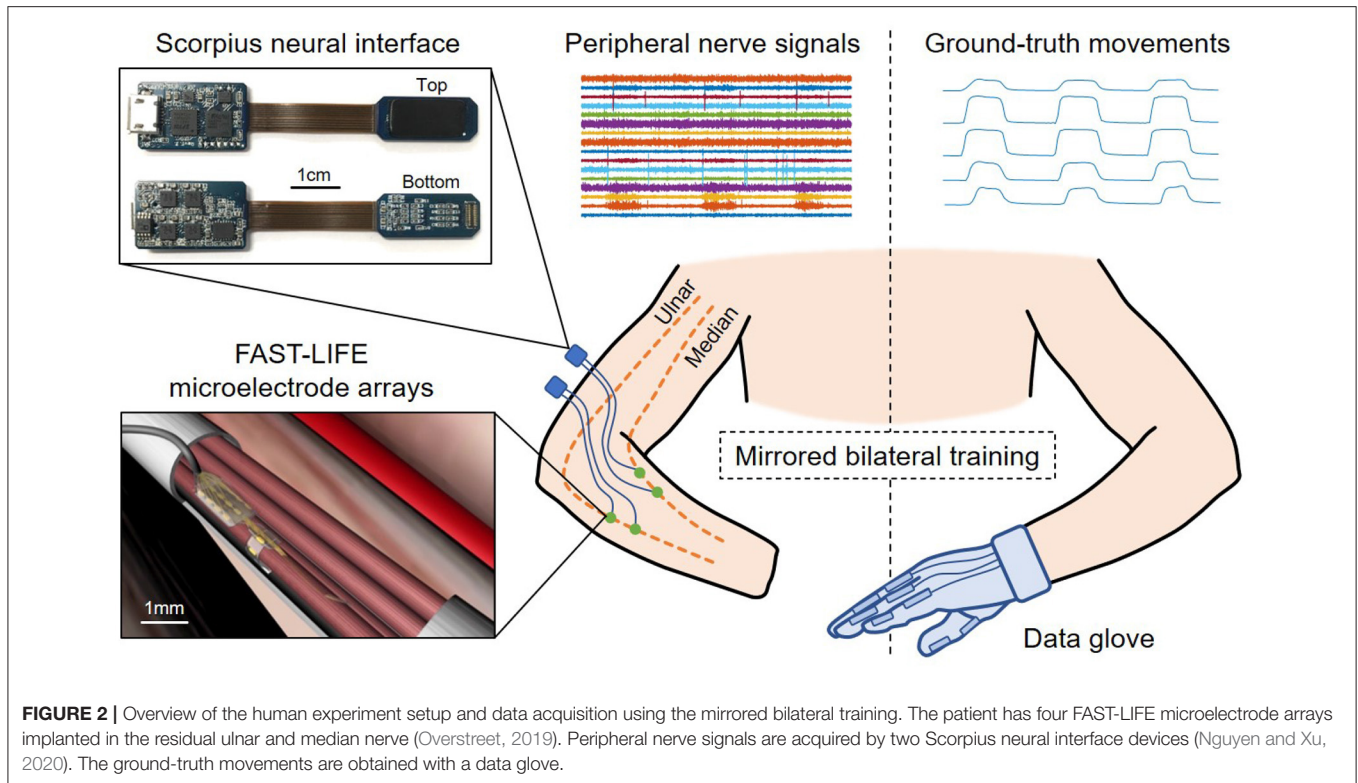
2.2. Nerve Data Acquisition

Nerve signals are acquired using the Scorpius neural interface (**Figure 2**)—a miniaturized, high-performance neural recording system developed by Yang's lab at UMN. The system employs the Neuronix chip family, which consists of fully-integrated neural recorders designed based on the frequency shaping (FS) architecture (Xu, 2014, 2020; Yang, 2016, 2018, 2020). The specifications of the Scorpius system are reported in Nguyen and Xu (2020). The system allows acquiring nerve signals with high-fidelity while suppressing artifacts and interference. Here two Scorpius devices are used to acquire signals from 16 channels across four microelectrode arrays at a sampling rate of 40 kHz (7.68 Mbps, 480 kbps per channel). The data are further downsampled to 5 kHz before applying a bandpass filter in 25–600 Hz bandwidth to capture most of the signals' power. This results in a pre-processed data stream of 1.28 Mbps (80 kbps per channel).

2.3. Ground-Truth Collection

The mirrored bilateral training paradigm (Sebelius, 2005; Jiang, 2012) is used to establish the ground-truth labels needed for supervised learning (**Figure 2**). The patient performs various hand gestures with the able hand while simultaneously imagining doing the same movement with the phantom/injured hand. During the ground-truth collection, the virtual hand and/or motorized prosthesis hand can follow the glove's movement to provide additional visual cues for the amputee. The ground truth is recorded while the patient poses his arms in different positions, including holding arms overhead, spreading to two sides, reaching the front, and resting along the body. The gestures include bending the thumb, index, middle, ring, little finger, index pinch, tripod pinch, grasp/fist, and resting. We record

¹<https://clinicaltrials.gov/ct2/show/NCT02994160>



the training data in multiple sessions. During each session, the amputee performs one hand gesture 100 times, each time 4 s altering between resting and flexing. Peripheral nerve signals are acquired from the injured hand with the Scorpius system, while ground-truth movements are captured with a data glove (VMG, 30, Virtual Motion Labs, TX) from the able hand. The glove can acquire up to 15 DOF; however, we only focus on the main 10 DOF (MCP and PIP) corresponding to the flexion/extension of five fingers.

3. DATA PREPROCESSING

3.1. Cutting Raw Neural Data

Raw neural data are cut using a sliding window to resemble online motor decoding (Figure 3A). Here the window's length is set to 4 s with an incremental step of 100 ms. At any instant of time, the decoder can only observe the past neural data. The pseudo-online dataset contains overlapping windows from a total of 50.7 min worth of neural recordings. Each of these 4 s neural data segments serves as an input trial of the motor decoding process later.

3.2. Feature Extraction

Previous studies have shown that feature extraction is an effective gateway to achieve optimal classification performance with signals in the low-frequency band by highlighting critical hidden information while rejecting unwanted noise and interference. Here we select 14 of the most simple and robust features that are

frequently used in previous motor decoding studies (Zardoshti-Kermani, 1995; Phinyomark, 2009, 2012; Rafiee, 2011). They are chosen such that there is no linear relationship between any pair of features. All features can be computed in the temporal domain with relatively simple arithmetic, thus aiding the implementation in future portable systems.

Table 1 summarizes the descriptions and formula of the features. Figure 3B illustrates the process to compute feature data. x_i is the 4 s neural data segments, which are further divided into windows of 100 ms with N is the window length. Two consecutive windows are 80% overlapped, which is equivalent to a 20 ms time step. This results in a data stream of 224 features over 16 channels with a data rate of 179.2 kbps (11.2 kbps per channel), which is more than 40 times lower than the raw data rate. Figure 4 presents an example of the feature data in one trial that shows a clear correlation between the changes of the 14 extracted features and the finger's movement. The amplitude of each feature is normalized by a fixed value before feeding to the deep learning models.

4. PROPOSED DEEP LEARNING MODELS AND DECODING STRATEGIES

4.1. Two-Step (2S) Strategy

Each finger exists in a binary state: active or inactive, depending on the patient's intent to move it or not. There are 32 different combinations of five fingers corresponding to 32 hand gestures. Only a few gestures are frequently used in daily living activities, such as bending a finger ("10000", "01000", ..., "00001"), index

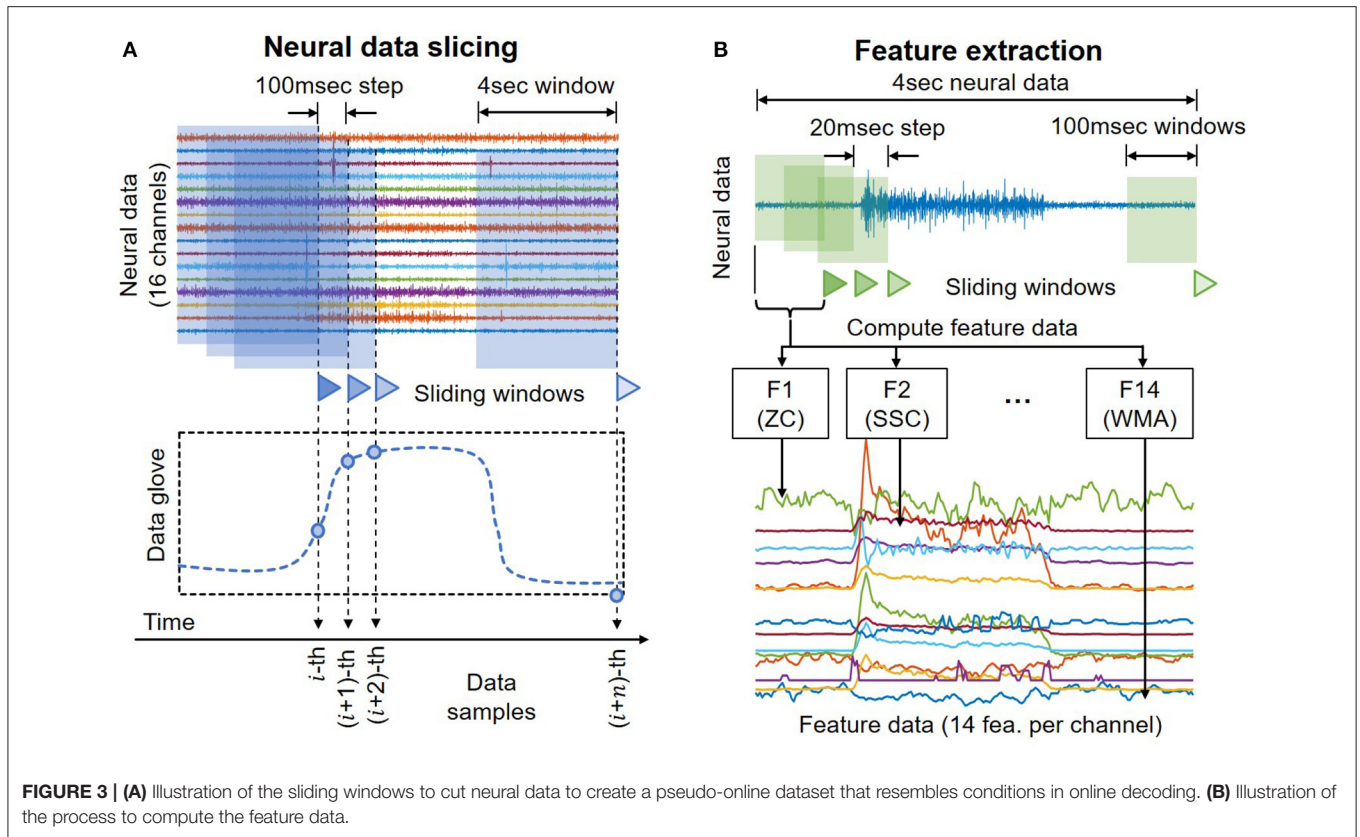
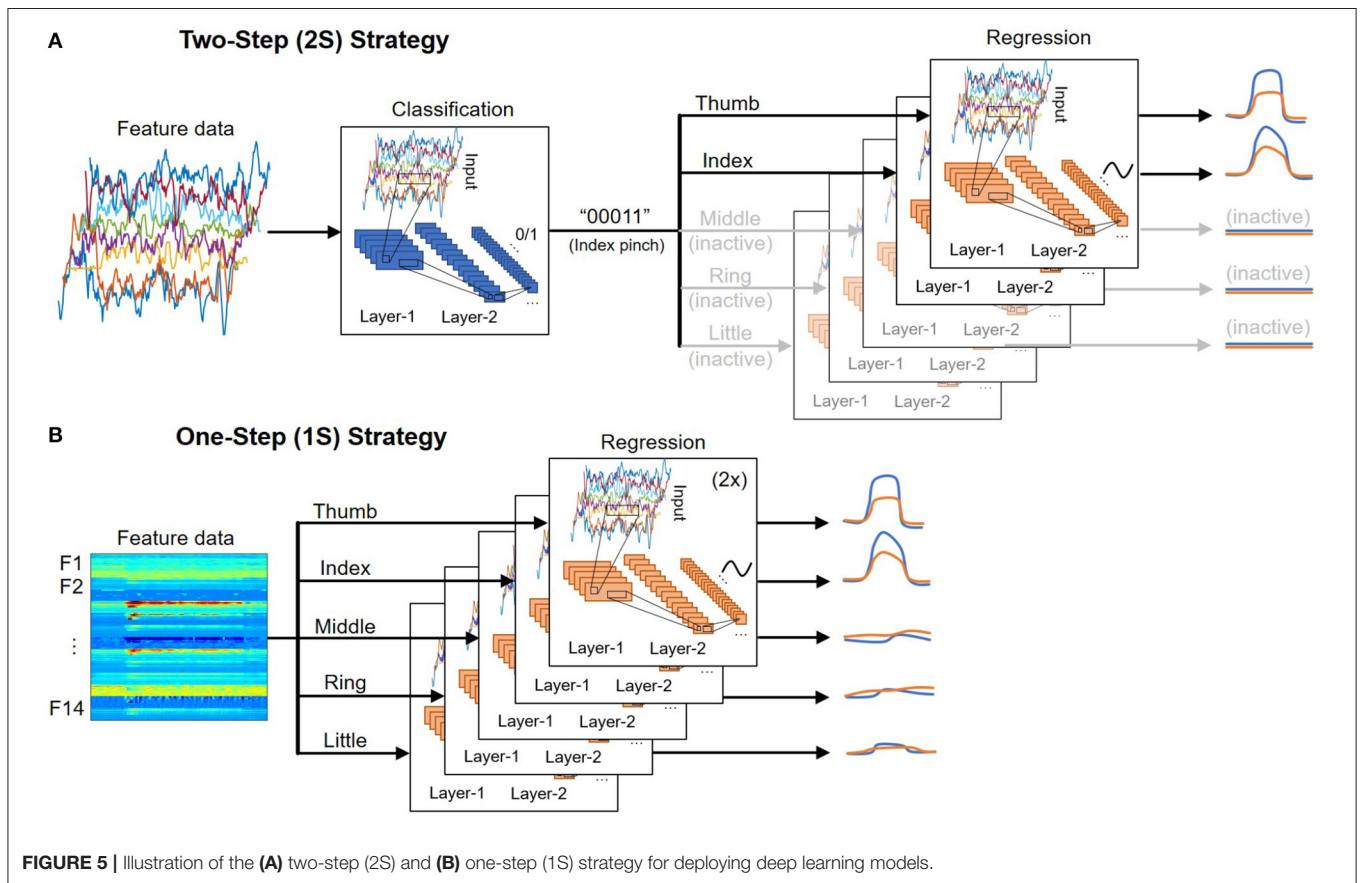
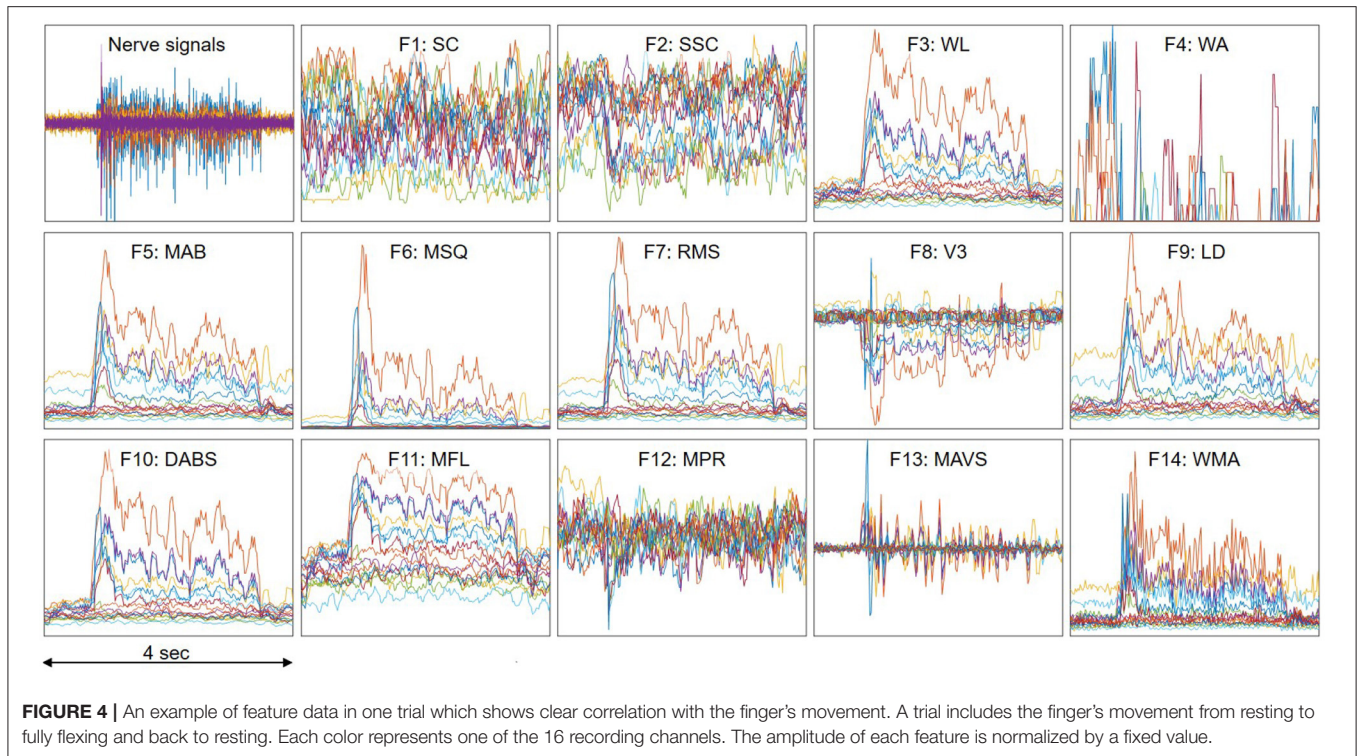


TABLE 1 | List of features, descriptions, and formula.

Feature	Description	Formula	
F1	Zero crossing (ZC)	The number of times the demeaned data change sign.	$\sum_{i=2}^N \text{sgn}(-x_{i-1}x_i)$
F2	Slope sign changes (SSC)	The number of times the differential data change sign.	$\sum_{i=3}^N \text{sgn}[-(x_i - x_{i-1})(x_{i-1} - x_{i-2})]$
F3	Waveform length (WL)	The summation of the absolute values of the differential data.	$\sum_{i=2}^N x_i - x_{i-1} $
F4	Wilson amplitude (WA)	The number of times the change in the signal amplitudes of two consecutive samples exceeds the standard deviation.	$\sum_{i=2}^N \text{sgn}(x_i - x_{i-1} - x_{std})$
F5	Mean absolute (MAB)	The average of the absolute values of the data.	$\frac{1}{N} \sum_{i=1}^N x_i $
F6	Mean square (MSQ)	The average of the square values of the data.	$\frac{1}{N} \sum_{i=1}^N x_i^2$
F7	Root mean square (RMS)	The root of MSQ or v-order 2.	$\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$
F8	V-order 3 (V3)	The cubic root of the average of the cube of the data.	$\sqrt[3]{\frac{1}{N} \sum_{i=1}^N x_i^3}$
F9	Log detector (LD)	The exponential of the average of the log data.	$\exp\left(\frac{1}{N} \sum_{i=1}^N \log x_i \right)$
F10	Difference absolute standard deviation (DABS)	Standard deviation of the absolute of the differential data.	$\sqrt{\frac{1}{N-1} \sum_{i=2}^N (x_i - x_{i-1})^2}$
F11	Maximum fractal length (MFL)	Equivalent to the log of DABS minus an offset that is equal to $1/2\log(N - 1)$	$\log\left[\sqrt{\sum_{i=2}^N (x_i - x_{i-1})^2}\right]$
F12	Myopulse percentage rate (MPR)	The number of times the absolute of the data exceeds the standard deviation.	$\sum_{i=1}^N \text{sgn}(x_i - x_{std})$
F13	Mean absolute value slope (MAVS)	A modified version of MAV that is the difference between the MAV of the first half of a signal window and the second half.	$\frac{\sum_{i=1}^{\lfloor N/2 \rfloor} x_i - \sum_{i=\lfloor N/2 \rfloor + 1}^N x_i }{\lfloor N/2 \rfloor}$
F14	Weighted mean absolute (WMA)	A modified version of the MAB where the first and last 25% of a signal window is given less weight than the middle 50%.	$\frac{1}{N} \sum_{i=1}^N w_i x_i $ where $w_i = 1$ if $i \in [0.25N, 0.75N]$, and $w_i = 0.5$ otherwise

pinching (“00011”), or grasp/fist (“11111”). Therefore, classifying the hand gesture before regressing the fingers’ trajectories would significantly reduce the possible outcome and lead to more

accurate predictions. The movement of inactive fingers could also be set to zero, which lessens the false positives when a finger “wiggles” while it is not supposed to.



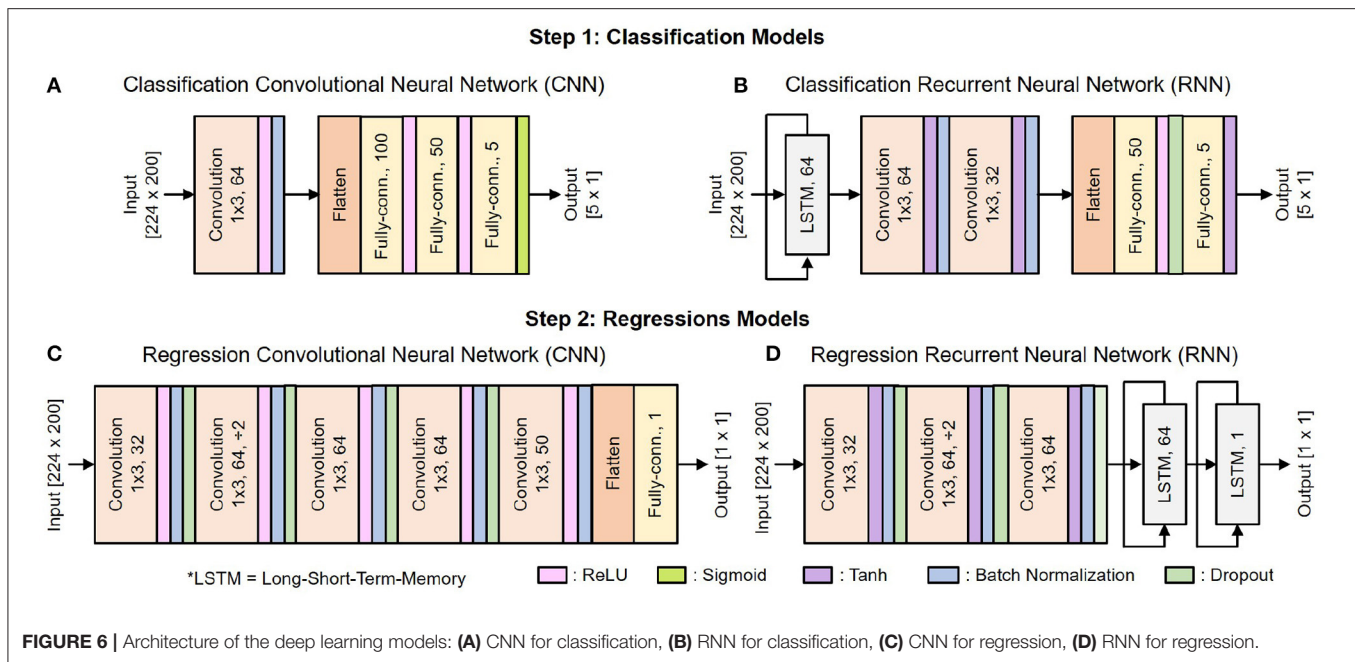


Figure 5A shows an illustration of the 2S strategy. In the first step, the classification output is a $[1 \times 5]$ vector encoding the state of five fingers. While the dataset used in this study only includes nine possible outcomes, the system can be easily expanded in the future to cover more hand gestures by appending the dataset and fine-tuning the models. In contrast, many past studies focus on classifying a specific motion, which requires modifying the architecture and re-training the models to account for additional gestures.

In the second step, the trajectory of each DOF is regressed by a deep learning model. Ten separate models regress the trajectory of 10 DOF (two per finger). The models associated with inactive fingers are disabled, and the prediction outputs are set to zero. As a result, the dataset used to train each model is only a subset of the full dataset where the corresponding DOF is active. While all models use the same architecture, they are independently optimized using different sets of training parameters such as learning rate, minibatch size, number of epochs, etc., to achieve the best performance. An advantage of this approach is that if one DOF fails or has poor performance, it would not affect the performance of others.

4.2. One-Step (1S) Strategy

Figure 5B shows an illustration of the 1S strategy. It is the most straightforward approach where the trajectories of each DOF are directly regressed regardless of the fingers' state. As a result, the full dataset, which includes data when the DOF is active (positive samples) and idle (negative samples), must be used to train each DOF. Because the number of negative samples often exceeds the number of positive samples from 5:1 to 10:1, additional steps such as data augmentation and/or weight balancing need to be done during training. This also leads to more false-positives where an idle DOF still has small movements that could affect the overall accuracy.

Although the 1S procedure is more straightforward than the 2S, its time-latency and efficiency are not necessarily better than the 2S. While our hands are at rest most of the time, the 1S approach has to continuously predict all fingers' trajectories regardless of their activeness status. The 2S approach goes through a classification step to identify the active DOF before predicting the trajectories of those DOF, which helps disperse deep learning models depending on the hand gestures, direct resources to the active DOF, and results in lower overall latency and computation in most implementations where there is only one processing unit (GPU or CPU) in the second step. It is shown in **Table 3** that a simple RF can help achieve comparable classification outcomes of accuracy above 0.99 to those from the two deep learning techniques. The regression steps of both approaches utilize the same deep learning model.

4.3. Deep Learning Models

Figure 6 shows the architecture of the deep learning classification and regression models. They include standard building blocks such as convolutional, long-short term memory (LSTM), fully-connected, and dropout layers of different combinations, order, and set of parameter values. The architecture is optimized by gradually adding layers and tuning their parameters while tracking the decoder's efficacy using 5-fold cross-validation. As the performance converges, additional layers would tend to result in over-fitting.

There are 10 copies of the regression model for 10 DOF, each of which is trained separately. We use Adam optimizer with the default parameters $\beta_1 = 0.99$, $\beta_2 = 0.999$, and a weight decay regularization $L_2 = 10^{-5}$. The mini-batch size is set to 38, with each training epoch consists of 10 mini-batches. The learning rate is initialized to 0.005 and reduced by a factor of 10 when the training loss stopped improving for two consecutive epochs.

TABLE 2 | Comparison between this work and Nguyen and Xu (2020).

	No. of layers			No. of parameters
	Conv.	LSTM	Fully conn.	
Nguyen and Xu (2020)	21	2	3	25,927,050
This work (classification)	1	0	3	1,465,749
This work (regression)	3	2	0	767,200

As shown in **Figure 6**, the input of both 1S and 2S networks is a matrix [224 x 200] where 224 is the total number of features, and 200 is the time vector. During the data preprocessing step, raw neural data are cut into several 4 s segments. Each of these 4 s segments goes through the feature extraction step, during which it is further cut into 100 ms segments with an incremental step of 20 ms as illustrated in **Figure 3B**. There are 14 features to be extracted from each channel as mentioned in **Table 1**. Hence, a 4 s segment of raw neural data from 16 channels results in a matrix [224 × 200] where 224 is the total number of extracted features from 16 channels [16 × 14 = 224] and 200 is the total number of time steps cut from 4 s segments [4 s/20 ms = 200].

Table 2 shows a rough comparison between the deep learning models used in this and our previous work. Note that for regression, the number of learnable parameters is a total of 10 models for 10 different DOF. The addition of feature extraction, thus dimensional reduction, allows significantly lowering the deep learning models' size and complexity. This is essential for translating the proposed decoding paradigm into a real-time implementation for portable systems.

5. EXPERIMENTAL SETUP

In this research, we investigate the performance of two main deep learning architectures: the CNN and RNN for both classification and regression tasks. Besides, the deep learning models are benchmarked against “classic” supervised machine learning techniques as the baseline. They include support vector machine (SVM), random forest (RF), and multi-layer perceptron (MLP).

For baseline techniques, the input of the classification task is the average of 224 features across 200 time-steps, while the input of the regression task is the 30 most important PCA components. The SVM models use the radial basis function (RBF) for classification and polynomial kernel of degree three for regression, with parameter $C = 1$. The RF models use 5 and 10 trees for classification and regression, respectively, with a max depth of three. The MLP model for classification is created by replacing the convolutional layer of the CNN model with a fully-connected layer of 200 units. The MLP model for regression has four layers with 300, 300, 300, and 50 units, respectively.

The 5-fold cross-validation is used to compare the performance of the classification task. For the regression task, the dataset is randomly split with 80% for training and 20% for validation. The split is done such that no data windows from the training set overlap with any data windows from the validation set.

The data processing is done in MATLAB (MathWorks, MA, USA). The deep learning networks are implemented in Python using the PyTorch² library. The deep learning models are trained and evaluated on a desktop computer with an Intel Core i7-8086K CPU and an NVIDIA TITAN Xp GPU.

6. METRICS AND RESULTS

6.1. Metrics

This subsection introduces the metrics to measure the performance of five models, including the two discussed deep learning models and three other supervised machine learning techniques as benchmarks in both classification and regression tasks.

The performance of the classification task is evaluated using standard metrics including accuracy and F1 score derived from true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) as follows:

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \quad (1)$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \quad (2)$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (3)$$

$$\text{Accuracy} = (\text{Sensitivity} + \text{Specificity})/2 \quad (4)$$

$$\text{F1 Score} = \frac{2 \cdot \text{Sensitivity} \cdot \text{Precision}}{\text{Sensitivity} + \text{Precision}} \quad (5)$$

We use the definition of accuracy with an equal weight of sensitivity and specificity because the occurrence of class-1 (active finger) is largely outnumbered by the occurrence of class-0 (inactive finger).

The performance of the regression task is quantified by two metrics: mean squared error (MSE) and variance accounted for (VAF). MSE measures the absolute deviation of an estimated from the actual value of a DOF, while VAF reflects the relative deviation from the actual values of several DOF. They are defined as follows:

$$\text{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (6)$$

$$\text{VAF}(y, \hat{y}) = 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (7)$$

where N is the number of samples, y is the ground-truth trajectory, \bar{y} is the average of y , and \hat{y} is the estimated trajectory. The value of y and \hat{y} are normalized in a range [0, 1] in which 0 represents the resting position.

Although the MSE is the most common metric and effectively measures absolute prediction errors, it cannot reflect the relative importance of each DOF to the general movements. For example, if the average magnitude of DOF A is hundreds of times smaller than that of DOF B, a bad estimation of A still can yield lower MSE than a reasonable estimation of B. The VAF score is more robust in such scenarios; thus, it could be used to compare the performance between DOF of different magnitude. The value of

²<https://pytorch.org/>

TABLE 3 | Classification performance.

	Accuracy					F1 score				
	Thumb	Index	Middle	Ring	Little	Thumb	Index	Middle	Ring	Little
SVM	0.999	0.767	0.916	0.895	0.932	0.999	0.808	0.911	0.771	0.677
RF	0.999	0.975	0.996	0.992	0.988	0.999	0.976	0.996	0.980	0.981
MLP	0.999	0.965	0.973	0.966	0.970	0.999	0.965	0.972	0.954	0.945
CNN	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
RNN	0.999	0.948	0.988	0.970	0.994	0.999	0.957	0.987	0.950	0.994

the VAF score ranges from $(-\infty, 1]$, the higher, the better. This research presents the MSE and VAF scores of different neural decoding models from different approaches to choose the best one for future clinical applications in real-time. A neural decoder that results in negative VAF scores is definitely not the best one. In fact, SVM is the only decoding architecture that occasionally shows negative VAF scores. Therefore, ignoring its negative VAF scores would neither affect the comparison nor change the final decision of the best model. For practicality, negative VAF values are ignored when presenting the data.

6.2. Classification Results

Table 3 shows the average 5-fold cross-validation classification results of all the techniques. The predictability of each finger is largely different from one another. The thumb, which produces strong signals only on the median nerve (first eight channels), is easily recognized by all techniques. Overall, CNN offers the best performance with accuracy and an F1 score for all fingers exceeding 99%. RF closely follows with performance ranging from 98 to 99%. While deep learning still outperforms classic techniques, it is worth noting that RF could also be a prominent candidate for real-world implementation because RF can be more efficiently deployed in low-power, portable systems.

6.3. Regression Results

Figure 7 presents the MSE and VAF scores for both strategies. In the 1S approach, the deep learning models, especially RNN, significantly outperform the other methods in both MSE and VAF, as shown in **Figures 7A,C**. To verify the significance of differences between RNN and the other decoders, we conduct paired t-tests with a Bonferroni correction. The results indicate that the performance differences in MSE and VAF are all statistically significant with $p < 0.001$. In the 2S approach, the performance is more consistent across all methods, where classic methods even outperform deep learning counterparts in certain DOF as shown in **Figures 7B,D**. Between the two strategies, the 1S approach generally gives better results; however, the high performance can only be achieved with RNN.

7. DISCUSSION

7.1. Feature Extraction Reduces Decoders' Complexity

Both deep learning architectures investigated in this study, namely CNN and RNN, deliver comparable motor decoding

performance to our previous work (Nguyen and Xu, 2020) while require much lower computational resources to implement. The average VAF score for most DOF is 0.7, with some exceeding 0.9. Such results are achieved with relatively shallow deep learning models with 4–5 layers, a significant reduction from the previous implementation with 26 layers. While an extra step of feature extraction is required, all feature extraction techniques are specifically designed to be efficiently computed with conventional arithmetic processors (e.g., CPU) and/or hardware accelerators (e.g., FPGA, ASIC).

Another direction for future works would be including additional features. Here we apply the most common features in the temporal domain for their simplicity and well-established standing for neuroprosthesis applications. There are other features, such as mean power, median frequency, peak frequency, etc., in the frequency domain that have been explored in previous literature. The purpose of this study is not to exhaustively investigate the effect of all existing features in motor decoding but to prove the effectiveness of the feature extraction method in achieving decent decoding outcomes. The success of this method will open up to future research on simultaneously applying more features in multiple domains for better outcomes.

Furthermore, it is worth noting that the use of feature extraction excludes most of the high-frequency band 600–3,000 Hz, which is shown in our previous work that could contain additional nerve information associated with neural spikes. A future direction would be extracting that information using spike detection and sorting techniques and combine them with the information of the low-frequency band to boost the prediction accuracy. However, the computational complexity must be carefully catered to not hinder the real-time aspect of the overall system.

7.2. Classic Machine Learning v.s. Deep Learning

The classification task can be accomplished with high accuracy using most classic machine learning techniques (e.g., RF) and near-perfect with deep learning approaches (e.g., CNN). Along with other evidence in Nguyen and Xu (2020), this suggests that nerve data captured by our neural interface contain apparent neural patterns that can be clearly recognized to control neuroprostheses. While the current dataset only covers 9/32 different hand gestures, these are still promising results that would support future developments, including expanding the dataset to cover additional gestures.

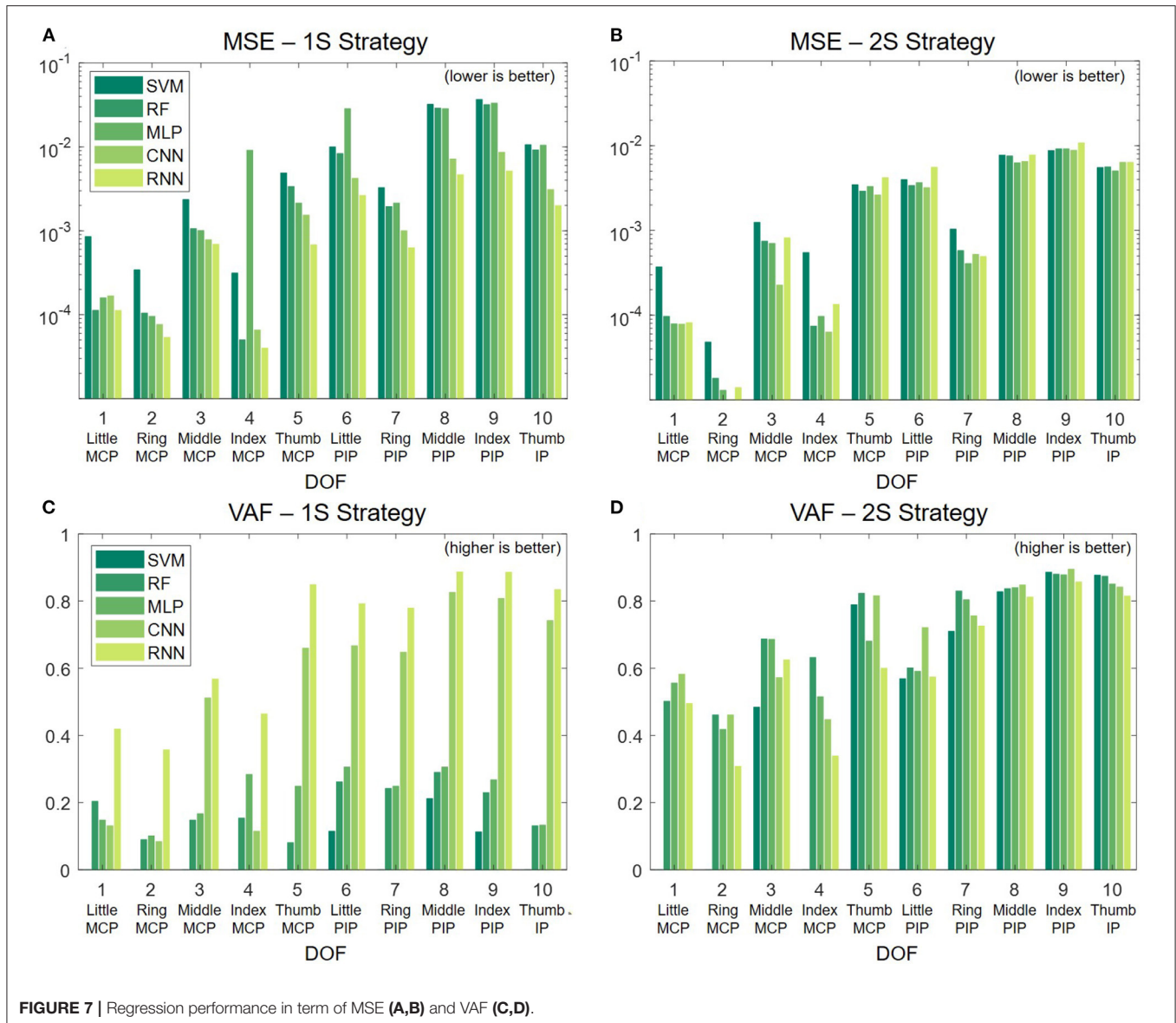


FIGURE 7 | Regression performance in term of MSE (A,B) and VAF (C,D).

The regression results are consistent with the conclusion of many past studies that deep learning techniques only show clear advantages over classic machine learning methods when handling a large dataset. This is evident in the 1S strategy, where each DOF is trained with the full dataset consisting of all possible hand gestures. In contrast, in the 2S strategy, where the dataset is divided into smaller subsets, deep learning techniques lose their leverage. However, as the dataset is expanded in the future, we generally believe that deep learning techniques should emerge as the dominant approach.

7.3. Improving Motor Decoder Effectiveness

This paper focuses on exploring different approaches to build a new decoding paradigm that is simpler and requires less

computational power than that of the previous research for applying in real-time applications. However, for such real-time applications, we also need to test the deep learning decoders' stability in the long run. Future research would apply the same deep learning architectures but train and validate on data recorded on different days. The time gap between the training and validation data may also be varied to investigate the trade-off between the re-training frequency and the decoding outcomes' accuracy.

8. CONCLUSION

This work presents several approaches to optimize the motor decoding paradigm that interprets the motor intent embedded in the peripheral nerve signals for controlling the prosthetic

hand. The use of feature extraction largely reduces the data dimensionality while retaining essential neural information in the low-frequency band. This allows achieving similar decoding performance with deep learning architectures of much lower computational complexity. Two different strategies for deploying deep learning models, namely 2S and 1S, with a classification and a regression stage, are also investigated. The results indicate that CNN and RF can deliver high accuracy classification performance, while RNN gives better regression performance when trained on the full dataset with the 1S approach. The findings layout an important foundation for the next development, which is translating the proposed motor decoding paradigm to real-time applications, which requires not only accuracy but also efficiency.

DATA AVAILABILITY STATEMENT

Further information and requests for resources and reagents should be directed to and will be fulfilled by Zhi Yang (yang5029@umn.edu) and Qi Zhao (qzhao@umn.edu) in accordance with the IRB protocol.

ETHICS STATEMENT

Written informed consent was obtained from the individual(s) for the publication of any

potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

ZY and EK conceptualized the system and experiments. DL, MJ, and QZ designed, trained and benchmarked the deep learning models. AN and JX designed the integrated circuits and carried out hardware integration. EK designed the microelectrode array. JC and EK performed the implantation surgery and post-operative care. DL, AN, JX, and MD conducted experiments, collected nerve data, and performed data analysis. DL and AN prepared the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

The surgery and patients related costs were supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grants HR0011-17-2-0060 and N66001-15-C-4016. The technology development and human experiments were supported in part by internal funding from the University of Minnesota, in part by the NIH under Grant R21-NS111214, in part by NSF CAREER Award No. 1845709, and in part by Fasikl Incorporated.

REFERENCES

- Amsuss, S., Goebel, P. M., Jiang, N., Graimann, B., Paredes, L., and Farina, D. (2013). Self-correcting pattern recognition system of surface EMG signals for upper limb prosthesis control. *IEEE Trans. Biomed. Eng.* 61, 1167–1176.
- Cheng, J., Tillery, S. H., Miguelez, J., Lachapelle, J., and Keefer, E. (2017). Dexterous hand control through fascicular targeting (HAPTIX-DEFT): level 4 evidence. *J. Hand Surgery* 42, S8–S9.
- Cordella, F., Ciancio, A. L., Sacchetti, R., Davalli, A., Cutti, A. G., Guglielmelli, et al. (2016). Literature review on needs of upper limb prosthesis users. *Front. Neurosci.* 10:209.
- Davis, T. S., Wark, H. A. C., Hutchinson, D. T., Warren, D. J., O'Neill, K., Scheinblum, T., et al. (2016). Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves. *J. Neural Eng.* 13:036001.
- Fougner, A., Stavadahl, O., Kyberd, P. J., Losier, Y. G., and Parker, P. A. (2012). Control of upper limb prostheses: terminology and proportional myoelectric control-A review. *IEEE Trans. Neural Syst. Rehabil. Eng.* 20, 663–677.
- George, J. A., Davis, T. S., Brinton, M. R., and Clark, G. A. (2020a). Intuitive neuromyoelectric control of a dexterous bionic arm using a modified kalman filter. *J. Neurosci. Methods* 330:108462.
- George, J. A., Radhakrishnan, S., Brinton, M. R., and Clark, G. A. (2020b). "Inexpensive and portable system for dexterous high-density myoelectric control of multiarticulate prostheses," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Toronto.
- Harris, A., Katyal, K., Para, M., and Thomas, J. (2011). "Revolutionizing prosthetics software technology," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, Anchorage, AK, 2877–2884.
- Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic Arm. *Nature* 485, 372.
- Hu, S., Wang, H., Zhang, J., Kong, W., Cao, Y., and Kozma, R. (2015). Comparison analysis: granger causality and new causality and their applications to motor imagery. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 1429–1444.
- Irwin, Z. T., Schroeder, K. E., Vu, P. P., Bullard, A. J., Tat, D. M., Nu, C. S., et al. (2017). Neural control of finger movement via intracortical brain-machine interface. *J. Neural Eng.* 14, 066004.
- Jiang, N., Vest-Nielsen, J. L., Muceli, S., & Farina, D. (2012). EMG-Based Simultaneous and Proportional Estimation of Wrist/Hand Kinematics in Unilateral Transradial Amputees. *Journal of Neuroengineering and Rehabilitation*, 9(1), 42.
- Johannes, M. S., Bigelow, J. D., Burck, J. M., Harshbarger, S. D., Kozlowski, M. V., and Van Doren, T. (2011). An overview of the developmental process for the modular prosthetic limb. *Johns Hopkins APL Techn. Digest* 30, 207–216.
- Krasoulis, A., Vijayakumar, S., and Nazarpour, K. (2019). Effect of user practice on prosthetic finger control with an intuitive myoelectric decoder. *Front. Neurosci.* 13:891.
- Kwon, O.-Y., Lee, M.-H., Guan, C., and Lee, S.-W. (2019). Subject-independent brain-computer interfaces based on Deep convolutional neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 3839–3852.
- Mahmud, M., Kaiser, M. S., Hussain, A., and Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 2063–2079.
- Micera, S., Rossini, P. M., Rigosa, J., Citi, L., Carpaneto, J., Raspopovic, S., et al. (2011). Decoding of grasping information from neural signals recorded using peripheral intrafascicular interfaces. *J. Neuroeng. Rehabil.* 8, 1–10.
- Mollazadeh, M., Greenwald, E., Thakor, N., Schieber, M., and Cauwenberghs, G. (2011). "Wireless micro-ECoG recording in primates during reach-to-grasp movements," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, San Diego, CA, 237–240.
- Nguyen, A. T., Xu, J., Jiang, M., Luu, D. K., Wu, T., Tam, W.-K., Zhao, W., et al. (2020). A Bioelectric neural interface towards intuitive prosthetic control For amputees. *J. Neural Eng.* 17, 066001.

- Overstreet, C. K., Cheng, J., and Keefer, E. W. (2019). Fascicle specific targeting for selective peripheral nerve stimulation. *J. Neural Eng.* 16, 066040.
- Phinyomark, A., Limsakul, C., and Phukpattaranont, P. (2009). A novel feature extraction for robust EMG pattern recognition. *arXiv:0912.3973*.
- Phinyomark, A., Nuidod, A., Phukpattaranont, P., and Limsakul, C. (2012). Feature extraction and reduction of wavelet transform coefficients for EMG pattern classification. *Elektr. Elektrotech.* 122, 27-32.
- Rafiee, J., Rafiee, M., Yavari, F., and Schoen, M. (2011). Feature extraction of forearm EMG signals for prosthetics. *Exp. Syst. Appl.* 38, 4058-4067.
- Sakhavi, S., Guan, C., and Yan, S. (2018). Learning temporal information for brain-computer interface using convolutional neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 5619-5629.
- Schultz, A. E., and Kuiken, T. A. (2011). Neural interfaces for control of upper limb prostheses: the state of the art and future possibilities. *PM R* 3, 55-67.
- Sebelius, F. C., Rosen, B. N., and Lundborg, G. N. (2005). Refined myoelectric control in below-elbow amputees using artificial neural networks and a data glove. *J. Hand Surg.* 30, 780-789.
- Vu, P. P., Irwin, Z. T., Bullard, A. J., Ambani, S. W., Sando, I. C., Urbanchek, M. G., et al. (2017). Closed-loop continuous hand control via chronic recording of regenerative peripheral nerve interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 515-526.
- Vu, P. P., Vaskov, A. K., Irwin, Z. T., Henning, P. T., Lueders, D. R., Laidlaw, A. T., et al. (2020). A regenerative peripheral nerve interface allows real-time control of an artificial hand in upper limb amputees. *Sci. Transl. Med.* 12:eaay2857.
- Vujaklija, I., Roche, A. D., Hasenoehrl, T., Sturma, A., Amsuess, S., Farina, D., et al. (2017). Translating research on myoelectric control into clinics-Are the performance assessment methods adequate? *Front. Neurobot.* 11:7.
- Wendelken, S., Page, D. M., Davis, T., Wark, H. A., Kluger, D. T., Duncan, C., et al. (2017). Restoration of motor control and proprioceptive and cutaneous sensation in humans with prior upper-limb amputation via multiple Utah slanted electrode arrays (USEAs) implanted in residual peripheral Arm nerves. *J. Neuroeng. Rehabil.* 14, 121.
- Xu, J., Nguyen, A. T., Luu, D. K., Drealan, M., and Yang, Z. (2020). Noise optimization techniques for switched-capacitor based neural interfaces. *IEEE Trans. Biomed. Circ. Syst.* 15, 1024-1035.
- Xu, J., Wu, T., Liu, W., and Yang, Z. (2014). A frequency shaping neural recorder with 3 pF input capacitance and 11 plus 4.5 bits dynamic range. *IEEE Trans. Biomed. Circ. Syst.* 8, 510-527.
- Yang, Z., Nguyen, A. T., and Xu, J. (2020). System and Method for Charge-Balancing Neurostimulator with Neural Recording. U.S. Patent No. 10,716,941.
- Yang, Z., Xu, J., Nguyen, A. T., and Wu, T. (2018). System and Method for Simultaneous Stimulation and Recording Using System-on-Chip (SoC) Architecture. U.S. Patent Application No. 15/876,030.
- Yang, Z., Xu, J., Nguyen, A. T., Wu, T., Zhao, W., and Tam, W.-k. (2016). "Neuronix enables continuous, simultaneous neural recording and electrical microstimulation," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 4451-4454.
- Zardoshti-Kermani, M., Wheeler, B. C., Badie, K., and Hashemi, R. M. (1995). EMG feature evaluation for movement control of upper extremity prostheses. *IEEE Trans. Rehabil. Eng.* 3, 324-333.
- Zeng, H., and Song, A. (2015). Optimizing single-trial EEG classification by stationary matrix logistic regression in brain-computer interface. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 2301-2313.
- Zhang, Y., Nieveen, J., Wendelken, S., Page, D., Davis, T., Bo, A. P. L., et al. (2017). "Individual hand movement detection and classification using peripheral nerve signals," in *International IEEE EMBS Conference on Neural Engineering (NER)*, Shanghai, 448-451.
- Ziegler-Graham, K., MacKenzie, E. J., Ephraim, P. L., Trivison, T. G., and Brookmeyer, R. (2008). Estimating the prevalence of limb loss in the United States: 2005 to 2050. *Arch. Phys. Med. Rehabil.* 89, 422-429.
- Zuleta, J., Ferro, M., Murillo, C., and Franco-Luna, R. (2019). "Identification of EMG activity with machine learning in patients with amputation of upper limbs for the development of mechanical prostheses," in *IOP Conference Series: Materials Science and Engineering*, 012010.

Conflict of Interest: ZY is co-founder of, and holds equity in, Fasikl Inc., a sponsor of this project. This interest has been reviewed and managed by the University of Minnesota in accordance with its Conflict of Interest policy. JC and EK have ownership in Nerves Incorporated, a sponsor of this project.

The remaining authors declare that this study received funding from Fasikl Inc. The funder had the following involvement with the study, including study design, data collection and analysis, and preparation of the manuscript.

Copyright © 2021 Luu, Nguyen, Jiang, Xu, Drealan, Cheng, Keefer, Zhao and Yang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.