Direct Learning of Neuronal Firing Representations for Long-Term Motor Intent Predictions

Long Meng and Xiaogang Hu, *Senior Member, IEEE*

***Abstract*—Accurate hand movement prediction plays a pivotal role in advancing robotic control technologies. Neuronal firing signals, as the driving representation of motor intentions, offer a physiologically meaningful approach to decode motor commands. These representations are typically extracted using blind source separation techniques. However, the high computational intensity of these methods limits practical applications. Therefore, we directly learned neuronal firing representations from surface electromyogram (sEMG) signals via an efficient deep forest (DF) framework. Specifically, we first obtained populational neuronal firing rate signals as the ground truth. The DF model was trained to map sEMG signals directly to populational neuronal firing rate. To enable robust and continuous finger force predictions, we evaluated the DF framework on data obtained across multiple sessions, with an average session interval of 6.58 days. Our results revealed that the DF framework accurately maps sEMG amplitudes to neuronal firing representations, achieving comparable accuracy to source-separation-based method with significantly reduced computational time. The developed DF model also outperformed neural network models and other decision-tree-based ensemble methods. Furthermore, despite utilizing the same input features, the DF framework significantly outperformed the sEMG-amplitude approach, showcasing its capacity to capture complex neural drive information for more accurate finger force predictions. Moreover, the robustness test against noise interference revealed that the DF framework maintained stable performance under different noise levels. These findings highlight the potential of DF framework as an efficient solution for real-time robotic control applications.**

***Impact Statement*—Accurate interpretation of motor intention from sEMG is essential for the control of advanced prosthetic and robotic hands, but current signal separation pipelines remain too slow for routine use. Because motor commands are conveyed by collective firings of motoneurons, we instead adopt direct learning of neuronal firing representations using a lightweight deep forest model. This approach preserves the neural detail of blind source separation while cutting computation time by about 80%. Despite the efficiency gain, the model achieved decoding accuracy comparable to current blind source separation methods tested on data in week-long intervals under varying noise levels. This direct learning technique supports accurate control of prosthetic or exoskeletal hands for long-term use. The demonstrated real-time efficiency and robustness also lay the foundation for future applications in dexterous robotic control.**

***Index Terms*—Deep forest, deep learning, finger force prediction, neural decoding, surface electromyogram.**

I. Introduction

HUMAN hands are among the most dexterous body parts, capable of performing intricate and precise movements essential for daily functioning [1]. In recent years, significant progress in robotics has led to the development of advanced prosthetic and exoskeletal hands, enabling the separate actuation of each finger and joint [2], [3]. Alongside mechatronic device advancements, the development of accurate and efficient neural decoding approaches is essential to facilitate intuitive robotic control, promoting their potential applications in human–machine interactions [4], [5], remote surgery [6], [7], and assisting individuals with physical disabilities in regaining functional independence [8], [9].

Hand movements are generally controlled by neural commands from the brain that travel through spinal pathways to motoneurons [10]. These neurons activate the innervated muscles, producing the desired movements. Surface electromyogram (sEMG), a noninvasive method for capturing neuromuscular activity, is widely used for decoding motor intentions [11], [12]. The introduction of flexible high-density sEMG (HD-sEMG) electrode arrays has further improved spatial resolution of sEMG signals, enabling accurate capture of detailed muscle activation patterns [13].

One approach to recognizing hand movements has been the classification of a limited number of predefined gestures [14], [15], [16]. However, the practical usage of this approach is limited by its inability to handle smooth transitions between gestures or adapt to new gestures. Alternatively, regression-based approaches, such as proportional direct control, are promising for mapping macroscopic or microscopic sEMG features to the desired motor output in a continuous way. The macroscopic features, such as sEMG amplitude, are commonly derived from global sEMG signals and have been extensively used in myoelectric control applications [17], [18]. However, macroscopic

Received 5 May 2025; revised 2 July 2025; accepted 6 August 2025. This work was supported in part by the National Science Foundation under Grant CBET-2246162, Grant IIS-2330862, and Grant IIS-2319139 and in part by the Department of Defense under Grant W81XWH2110185. This article was recommended for publication by Associate Editor Yucheng Dong upon evaluation of the reviewers' comments. *(Corresponding author: Xiaogang Hu.)*

Long Meng is with the Department of Mechanical Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: lmm7405@psu.edu).

Xiaogang Hu is with the Department of Mechanical Engineering, Pennsylvania State University, University Park, PA 16802 USA, and also with the Departments of Kinesiology, and Physical Medicine & Rehabilitation, the Huck Institutes of the Life Sciences, and the Center for Neural Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: xxh120@psu.edu).

Digital Object Identifier 10.1109/TAI.2025.3597271

sEMG features often suffer from an inaccurate representation of the underlying neural drive due to various factors, including superimposition of action potentials, crosstalk from neighboring compartments, and motion artifacts [15], [19]. Such inaccurate representations result in prediction errors, thereby limiting their practical applications. Recently, research has shown the possibility to extract microscopic features [20]. For example, motoneuron discharge trains are extracted from HD-sEMG signals via blind source separation (BSS) techniques for detailed and accurate analysis of neuromuscular activity [20], [21], [22]. Then, the population neural firing rate is calculated as a representation of the neural drive signal of finger muscles, thereby addressing limitations of macroscopic features and improving motor intention decoding accuracy [23], [24]. Because BSS-based decomposition typically requires substantial computational time, an effective strategy is to compute the separation matrix using BSS-based decomposition during the training phase, and then apply it to sEMG decomposition in the testing phase [25]. Although promising, the computational demand of such technique is high. Accordingly, there is an urgent need for an approach to directly learn neuronal firing representations from sEMG signals.

Convolutional neural network (CNN)-based and long short-term memory (LSTM)-based deep neural network models have demonstrated strong capabilities in modeling complex nonlinear input–output relations, suitable for neural decoding tasks [15]. For example, A CNN-LSTM-based framework has been used to estimate joint trajectories from sEMG signals, enabling position control in cable-driven exoskeletons [26]. Other deep learning architectures, including temporal convolutional networks and CNN with squeeze-and-excitation modules (CNN-SE) have also been used for real-time motor intent decoding for prosthetic control [27]. Similarly, CNN-, CNN-LSTM-, and CNN-gated recurrent unit (GRU)-based models have been implemented for elbow force estimation [28], demonstrating superior performance over traditional regression methods. However, several factors need to be considered before their implementation. The performance of most deep learning frameworks is sensitive to hyperparameter selection [29]. Improper parameter configurations can result in poor convergence, overfitting, or underfitting, thereby degrading prediction performance. Additionally, deeper models with higher complexity can capture more intricate input–output relations but require larger training datasets to mitigate overfitting risks [30]. This hinders their application in data-constrained settings, whereas the DF model offers a lightweight, hyperparameter-insensitive alternative that performs well even with small datasets and limited computational resources.

Another critical challenge hindering widespread adoption lies in the model adaptability to long-term usage scenarios [15]. In the training phase, models may overly rely on random, time-sensitive patterns that do not persist over time. For example, noise with a specific power level affecting a subset of electrodes can dominate the learning process [13]. While these patterns may enhance short-term performance, they fail to capture the stable feature representations of sEMG signals, degrading long-term model performance.

As an alternative deep learning framework, deep forest (DF) [31] is a decision tree-based architecture that processes inputs layer by layer through ensembles of decision trees, forming a hierarchical structure [13]. In each layer, multiple forest models (e.g., random forests) produce probabilistic outputs, which are fused with the original features for subsequent layers. This adaptive cascade approach enables deep feature extraction without compromising performance on limited training data. DF can also reduce overfitting by stopping layer expansion once model performance converges. Additionally, a key advantage of deep forest is its robustness to hyperparameter selection [31], making it less sensitive to model tuning compared with traditional neural-network-based deep learning frameworks, and potentially beneficial for cross-session generalization, where data distributions vary across days.

In this study, we pioneered the direct learning of neural firing representations for long-term continuous finger force predictions across multiple days, using an efficient deep forest model. Specifically, to evaluate model performance over time, we collected HD-sEMG signals in three separate sessions, while subjects performed single- and multifinger force tasks. The DF decoding models were built using data from one session and tested on the two remaining sessions. In the training phase, a two-stage BSS-based decomposition approach [32] was applied to extract populational neuronal firing signals. These representations were used as outputs for deep forest models, with sEMG amplitude [root mean square (rms)] features serving as inputs. Then, cross-session neural firing representations were learned from rms features using the trained deep forest models. Lastly, the decoded representations were mapped to finger forces using linear regressions. Our results demonstrated that the DF model could achieve better prediction accuracy than conventional neural-network-based models and comparable prediction accuracy to the BSS-based decoding approach, and that DF can achieve a significantly reduced computational time, underscoring the feasibility of directly learning neural firing representations in an efficient way. The contributions of our study are as follows.

1) First application of the deep forest model to directly learn neuronal firing representations from sEMG signals.
2) Consistently high computational efficiency that addresses the computational limitations of BSS-based neural decoding approaches, while achieving comparable decoding performance and maintaining robustness under varying noise levels.
3) Validation of the learned neuronal firing representations through continuous finger force predictions across multiple days, demonstrating their physiological relevance and practical utility. The DF-based framework further outperformed amplitude-based methods, deep neural network architectures, and other ensemble models.

II. Materials

A. Subject Information

Eight subjects participated in this experiment, including five males and three females. All participants were neurologically intact, with ages ranging from 21 to 35. Before participation,

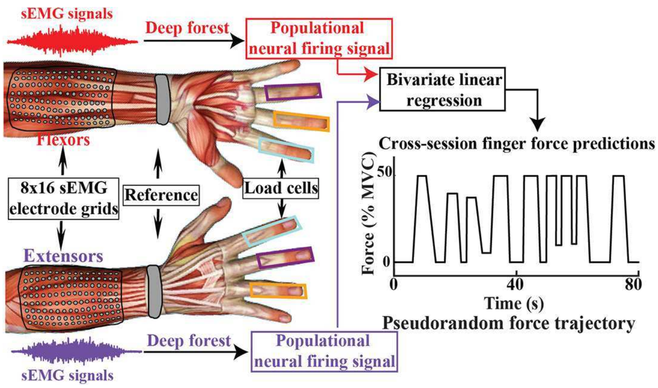

Fig. 1. Overview of the research framework and experimental setup.

subjects signed a consent form acknowledging that they had fully understood the experimental procedures, purposes, their rights, and potential risks. This experiment was reviewed and approved by the Institutional Review Board of the Pennsylvania State University.

B. Data Acquisition

1) Force Measurement: As shown in Fig. 1, three fingers (index, middle, and ring fingers) were attached to separate miniature load cells (SM-200N, Interface) for individual force measurements, sampled at a rate of 1000 Hz. Before each session, we measured the maximum voluntary contraction (MVC) of each finger flexion. Then, subjects performed two types of tasks (single- and multifinger tasks) in a random order. In the single-finger tasks, subjects flexed only one finger following a pseudorandom force pattern (Fig. 1) under the instruction to avoid coactivation of other fingers. Three repetitions were performed for each involved finger, resulting in a total of nine (three trials/finger $\times$ three fingers) single-finger trials. In the multifinger tasks, subjects concurrently flexed the three fingers, following the same pseudorandom force pattern, with three multifinger trials performed in total. Therefore, a total of 12 trials were performed for each session. Each trial lasted 80 s, and each subject participated in three sessions with an average intersession interval of 6.58 days, enabling the assessment of long-term decoding performance. The pseudorandom force trajectory, as illustrated in Fig. 1, consisted of trapezoidal forces ranging from 0% to 50% MVC, ramped with varying durations and amplitudes to introduce unpredictability in force patterns. The 50% MVC was chosen as the upper limit because it aligns with the force levels required for most daily activities while minimizing the risk of fatigue.

2) sEMG Data Acquisition and Preprocessing: For sEMG data acquisition, we first cleaned the skin over the flexor digitorum superficialis (FDS) and extensor digitorum communis (EDC) of the forearm to reduce skin-electrode impedance. Then, two 8 $\times$ 16 (128-channel) electrode arrays were placed on the FDS and EDC, respectively, by palpating the skin around these muscles. Each electrode array featured electrodes with a diameter of 3 mm and an interelectrode distance of 10 mm. During the experiment, sEMG signals were sampled at a rate of 2048 Hz, amplified with a gain of 1000 and a pass band of 10–900 Hz via EMG-USB2+ system (OT Bioelettronica, Torino, Italy). The collected sEMG data were preprocessed using a sEMG-specific interference removal approach [33] to eliminate motion artifacts.

III. Methods

A. Source Separation-Based Neural Decoding Approach

To derive training labels and comparing the performance of DF and BSS-based decoders, we applied and further customized a previously developed a source-separation-based neural decoding approach [32], in order to extract populational neuron firing signals as finger force-driven neural firing representations. This neural decoding approach included two key steps: 1) initial motor unit (MU) extraction; and 2) MU pool refinement. In our study, we extracted neuronal firing representations separately for sEMG collected from FDS and EDC. Afterward, we evaluated their contributions and combined their effects on finger force predictions.

1) Initial MU Extraction: Based on our preliminary experiment, we divided each 80-s trial from the training dataset of finger l ($l \in \{\text{index}, \ \text{middle}, \ \text{ring}\}$) into four 20-s segments for the initial MU extraction. Specifically, we employed a BBS approach, the fast independent component analysis (FastICA) algorithm [34], for the 128-channel 20-s sEMG decomposition. FastICA was selected as the baseline decomposition method because it has been widely validated in sEMG decomposition studies for its high decomposition accuracy and fast computational convergence, and serves as one of the most commonly used BSS methods for extracting motoneuron discharge information from high-density sEMG [23], [35], [36]. To increase the observations, we first conducted channel extensions by duplicating the original 128 channels by a factor of 9. The nine sets of duplicated data were incrementally delayed by one to nine samples. To remove the correlation between observations, we whitened the extended signals. Then, we applied the FastICA algorithm to obtain the MU source signals and corresponding separation vectors. The parameter settings were consistent with a previous study [25]. For example, we employed the contrast function $G(x) = (1/3)x^3$ to accelerate convergence. The number of decomposed MUs was set to 200. K-means++ was applied for binary clustering of discharging events. After the sEMG decomposition, the MU quality was evaluated using the silhouette (SIL) value. MUs with SIL lower than 0.5 were removed from further analysis. In addition, duplicated MUs were identified when more than 80% of their spike trains were synchronized within a $\pm$2.5 ms time window [25], and the MU with the higher SIL was retained. Then, we concatenated the separation vectors from all 20-s sEMG segmentation to obtain the raw MU pool and separation matrix for finger l ($B_{1,l}$).

2) MU Pool Refinement: To obtain the MU pool specific to the finger l, we refined the raw MU pool by quantifying the correlation [coefficient of determination (R^2)] between the firing rate of each MU and finger forces. Specifically, we employed the initial separation matrix to decompose all the single-finger trials and calculated corresponding spike trains. The spike trains were then segmented using a sliding window of 0.5 s and a sliding step of 0.125 s. The spikes within each segment were summed and concatenated to form a time course of firing rates. The time series of firing rate was smoothed by a Kalman filter to address sporadic, large-amplitude, and isolated fluctuations. Based on previous studies [23], [24], the parameters of observation matrix, observation covariance, system matrix and system covariance were set to 1, 0.5, 1, and 0.1, respectively. For each MU, we calculated the R^2 of its smoothed firing rate with the force of activated fingers. If the average R^2 for finger l was the highest, we retained this MU. The separation vectors of all retained MUs formed the refined separation matrix ($B_{2,l}$).

3) Neural Firing Signal Extraction: To derive the neuronal firing representation as our learning target, we applied $B_{2,l}$ to decompose trials corresponding to finger l. Similarly, we extracted the time courses of firing rates for all retained MUs. These firing rate time courses were then averaged and smoothed using a Kalman filter, resulting in the neuronal firing representations used for subsequent analyses.

4) Performance Evaluation: Considering that neuronal firing representations were generally linear to the target finger forces [32], we evaluated the extraction performance of the neuronal firing representation via the reconstruction accuracy of the target finger forces. Specifically, we employed a bivariate linear regression analysis to combine the neuronal firing representations obtained from FDS ($\boldsymbol{D}_{f,l}$) and EDC ($\boldsymbol{D}_{f,l}$) for the force prediction of finger l as

$$\text{Force}_l = a_l \boldsymbol{D}_{f,l} + b_l \boldsymbol{D}_{e,l} + c_l \tag{1}$$

where Force_l represents the predicted force of finger l; a_l and b_l represent the coefficients of $\boldsymbol{D}_{f,l}$ and $\boldsymbol{D}_{e,l}$, respectively. c_l represents the intercept.

The predicted forces were compared with the recorded ground truth values and evaluated using two widely-used metrics, the coefficient of determination (R^2) and root mean square error (RMSE), which were presented in the form of mean $\pm$ standard error. The definitions of R^2 and RMSE were as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

where n is the total number of observations; y_i is the measured force for the ith observation; $\hat{y}_i$ is the predicted force for the ith observation; $\bar{y}$ is the average value of measured force, calculated as $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$.

B. Deep Forest Framework

For the training procedure of the DF model, we first segmented the sEMG signals using a sliding window of 0.5 s with a step size of 0.125 s. For each segment, we extracted the rms values from all channels as the input feature vector, as rms has been widely used in muscle force prediction due to its robustness and computational efficiency [37]. The target outputs were the neural firing signals derived from the decomposition.

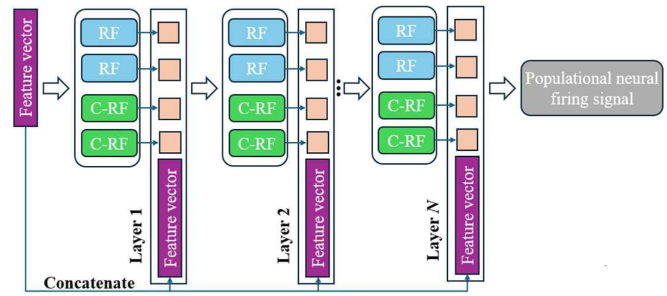

Fig. 2. Deep forest framework. RF and C-RF denote random forest and completely random forest, respectively.

The DF framework (Fig. 2) is a multilayer cascade of forest ensemble modules, each consisting of one random forest (RF) and one completely random forest (C-RF). Each forest produces class distributions at its leaf nodes, which are averaged across trees and concatenated to form the module output. These outputs are then concatenated with the original input features to form an augmented feature vector, which is passed to the next layer. This feature augmentation strategy allows the model to progressively enrich the representation and capture higher-order interactions.

The model was trained in a greedy, layer-wise manner. For each layer, the RF and C-RF were trained using the current input feature vector, while parameters from previous layers were kept fixed to stabilize the learning process. New layers were added sequentially, and training was terminated once the reduction in regression error between two successive layers was below a predefined threshold (10^{-5} in this study), indicating convergence. Each forest contained 100 trees, and two ensemble modules (i.e., two RF + two C-RF) were used per layer [13]. All other hyperparameters followed the default settings introduced in [31].

C. CNN Models

We also compared the DF framework with efficient neural network-based models. CNN models were selected because of their proven ability to learn hierarchical spatial features from biomedical signals and their strong performance in prior sEMG decoding tasks [38], [39], [40]. Specifically, we implemented two CNN frameworks (Fig. 3) for comparison: 1-D CNN and 2-D CNN. For the 128-channel sEMG data, we directly fed the 0.5-s sEMG signal ($\text{X}_{1-\text{D}} \in \mathbb{R}^{N_c \times N_P}$) into the 1-D CNN to extract features, where $N_c = 128$ denotes the number of sEMG channels and $N_P = 0.5\,\text{s} \times 2048\,\text{Hz} = 1024$ denotes the number of data points. To investigate the effects of the spatial information on the prediction of neuronal firing signals, we constructed the 2-D CNN using the rms feature map as the input ($\text{X}_{2-\text{D}} \in \mathbb{R}^{1 \times N_c \times N_R}$), where $N_C \times N_R = 8 \times 16$ denotes the shape of the rms feature map. The rms values were extracted from each segmented 0.5-s sEMG signal. Given the relatively

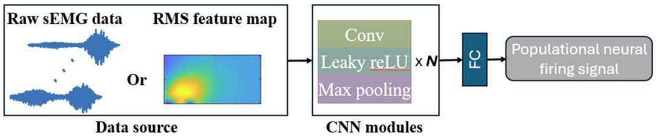

Fig. 3. CNN frameworks. Conv denotes convolutional layer; FC denotes fully connected layer.

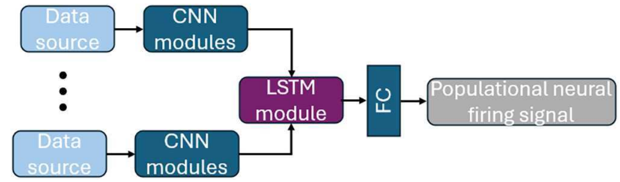

Fig. 4. CNN-LSTM framework.

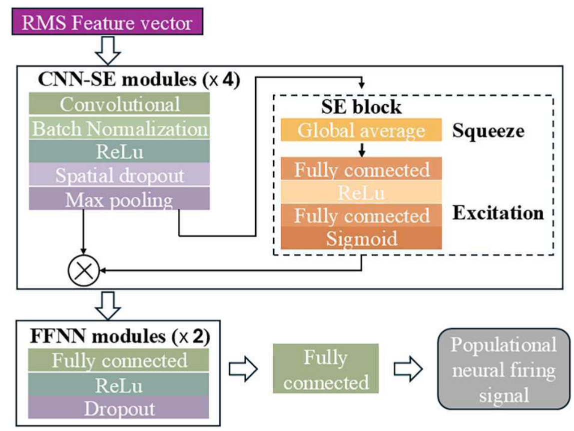

Fig. 5. CNN-SE framework. ⊗ denotes channel-wise multiplication.

small training dataset, we employed lightweight CNN frameworks to prevent overfitting issues. Both CNN frameworks used a modular architecture with convolutional layers, followed by Leaky ReLU activations and max-pooling operations (kernel size = 2, stride = 2). To achieve optimal results, we systematically explored various combinations of layer numbers ($N \in \{1, 2, 3\}$) and kernel sizes ({3,5,7}). The CNN performance on the validation dataset revealed that a single CNN module (i.e., $N = 1$) with a kernel of three achieved the best balance between prediction accuracy and computational efficiency. The model training was conducted using a batch size of 64 and a learning rate of 0.001, optimized with the Adam algorithm [41]. The maximum number of iterations was set to 300 for the training process. The mean squared error (MSE) loss function, augmented with L2 regularization (weight decay of 10^{-4}), ensured stable training and generalization. A fully connected layer with 64 units aggregated extracted features for final predictions. Twenty percent of the training data was reserved as the validation dataset. An early stopping strategy was applied. Namely, the training process was stopped if the validation loss did not decrease for ten consecutive epochs. The model achieving the lowest validation loss was used for subsequent analyses.

D. CNN-LSTM Model

To leverage the proven capability of LSTM networks in capturing long-range temporal dependencies, we extended the optimal CNN architecture by integrating it with LSTM to construct a CNN-LSTM model (Fig. 4). This hybrid architecture was selected due to its effectiveness in modeling sequential sEMG patterns, where CNN extracted local spatial features and LSTM captured their progression over time. Specifically, the input data type and the configuration of the CNN modules (including the number of layers and kernel size) were the same as those of the better CNN model (1-D or 2-D CNN). The input data type and the configuration of the CNN modules (including the number of layers and kernel size) were the same as those of the better CNN model (1-D or 2-D CNN). In this model, the 0.5-s sEMG signal was further divided into five nonoverlapping 0.1-s segments, which were then processed by each CNN module. The LSTM layer, with a single layer of 64 hidden units [15], processed the sequential feature vectors to model temporal dependencies across the five 0.1-s segments. This structure allowed the LSTM to retain memory over longer temporal sequences, capturing the dynamic patterns of the input. The LSTM employed three gating mechanisms—forget gate, input gate, and output gate—to manage information flow effectively and mitigate the vanishing gradient problem, ensuring robust learning of temporal relations. The hidden state output from the LSTM layer was passed to an FC layer with 64 units, mapping the features to a single output neuron for the prediction of neuronal firing signals.

E. CNN-SE Model

Motivated by recent work on attention-based CNN for motor intent decoding [27], we tailored the CNN-SE architecture for our task. These mechanisms aimed to adaptively recalibrate channel-wise feature responses by explicitly modeling interdependencies between channels, thereby enhancing the network sensitivity to informative patterns in the sEMG input. To ensure a fair comparison, we preserved the original network structure and parameters to the maximum extent. As shown in Fig. 5, the CNN-SE model consisted of four sequential Conv-SE modules, each comprising a 1-D convolutional layer (with kernel sizes of 20, 5, 3, and 3, respectively), batch normalization, ReLU activation, spatial dropout (dropout rate = 0.1), and max pooling (kernel sizes: 5, 3, and 2 for the last three modules). Each convolutional layer had 64 output channels. A SE block was appended to each Conv-SE module to recalibrate feature responses by adaptively weighting the channel-wise activations through a squeeze (global average pooling) and excitation (two fully connected layers with ReLU and sigmoid activations) operation. After passing through the CNN-SE stack, the output feature vector was processed by two feedforward neural network (FFNN) modules. Each FFNN module consisted of a fully connected layer with 128 units, a ReLU activation function, and dropout. This was followed by a final fully connected layer to predict the populational neural firing signal.

TABLE I
OPTIMAL HYPERPARAMETERS AND SEARCH RANGES FOR TREE-BASED ENSEMBLE MODELS

XGBoost	LightGBM	CatBoost
Max tree depth: 5 (3,5,7)	Max tree depth: 7 (3,5,7)	Max tree depth: 7 (3,5,7)
Learning rate: 0.05 (0.01, 0.05, 0.1)	Learning rate: 0.1 (0.01, 0.05, 0.1)	Learning rate: 0.1 (0.01, 0.05, 0.1)
Number of trees: 500 (100, 300, 500)	Number of trees: 500 (100, 300, 500)	Number of boosting iterations: 500 (100, 300, 500)
Row sampling ratio: 0.6 (0.6, 0.8, 1)	Maximum leaf nodes: 31 (31, 63, 127)	L2 regularization for leaf scores: 123 (0.001, 0.1, 1)
Column sampling ratio by node: 0.6 (0.6, 0.8, 1.0)		Threshold for one-hot encoding: 5 (5, 10, 20)
L2 regularization term on weights: 0.001 (0.001, 0.1, 1)		

Note: Value before the parentheses indicates the selected optimal parameter, while the values inside the parentheses represent the grid search range explored during model tuning

F. Tree-Based Ensemble Models

To provide a fair and comprehensive evaluation, we included three widely used tree-based ensemble methods in our comparison: extreme gradient boosting (XGBoost) [42], light gradient boosting machine (LightGBM) [43], and categorical boosting (CatBoost) [44]. To ensure consistency across methods, the input features and output targets for these models were kept identical to those used in the DF framework.

XGBoost is an optimized implementation of gradient boosting that constructs trees in a level-wise manner and employs second-order derivatives for more accurate loss approximation. Its regularization mechanisms help prevent overfitting and improve generalization. XGBoost has shown excellent performance in a variety of sEMG-based decoding studies [45], [46], including force and joint angle predictions, due to its ability to capture complex nonlinear relations while maintaining computational efficiency.

LightGBM is a gradient boosting framework that introduces two major innovations: gradient-based one-side sampling and exclusive feature bundling. These improvements enable faster training and lower memory usage. Unlike the level-wise tree growth in XGBoost, LightGBM grows leaf-wise trees with depth constraints, which allows for deeper and more specialized tree structures. LightGBM has been implemented in decoding applications for its speed and accuracy, especially on large datasets [47], [48].

CatBoost is a gradient boosting algorithm that is particularly suited for datasets with categorical features. It introduces ordered boosting and symmetric trees to reduce overfitting and improve stability. It has been applied recently in biosignal decoding [47] and gait prediction tasks [49]. While categorical encoding is less relevant for our continuous sEMG features, the robustness and regularization strategies of CatBoost still make it a competitive choice.

To ensure rigorous evaluation, we conducted a grid search for the key hyperparameters of each method using the training dataset and selected the best-performing model based on the RMSE on the validation dataset. The optimal hyperparameter settings and their respective search ranges are summarized in Table I. For each method, the values in parentheses represent the grid search range, while the values preceding them indicate the selected optimal parameters.

G. sEMG-Amplitude-Based Force Predictions

In addition, we also compared the DF and BSS (FastICA) decoding approaches with the commonly used sEMG amplitude approach. This method was included as a representative traditional baseline method because muscle activation levels are generally proportional to sEMG amplitude (rms), and rms-based features have been frequently adopted in intent prediction studies [50], [51], [52]. Specifically, the sEMG data were segmented using the same sliding window strategy as the DF approach. For each segment, we calculated the average rms of sEMG data from FDS ($\boldsymbol{A}_{f,l}$) and EDC ($\boldsymbol{A}_{e,l}$), respectively. Then, the force of finger l was predicted using a bivariate linear regression model

$$\mathrm{Force}_l = a_l \boldsymbol{A}_{f,l} + b_l \boldsymbol{A}_{e,l} + c_l \tag{4}$$

where a_l and b_l represent the coefficients of $\boldsymbol{A}_{f,l}$ and $\boldsymbol{A}_{e,l}$, respectively. c_l represents the intercept.

H. Validation Protocols

In this study, two validation protocols were explored, namely, within-session and cross-session protocols.

1) Within-Session Protocol: All the training, validation, and testing data came from the same recording session. For finger l ($l \in \{\mathrm{index},\ \mathrm{middle},\ \mathrm{ring}\}$), there were three single-finger trials and three multifinger trials. We divided the data into three sets, each of which had a single-finger trial and a multifinger trial. The three sets alternated as the testing dataset, while the remaining two sets were combined and randomly divided into training and validation datasets with an 8:2 ratio. The average results across the testing datasets were then calculated and reported.

2) Cross-Session Protocol: To evaluate long-term model performance, we employed a leave-one-session-out validation protocol. Specifically, data from one session were randomly divided into training and validation datasets with an 8:2 ratio. The remaining two sessions were used as the testing dataset to assess the generalizability of the model across different days. This procedure was repeated three times, with each session taking turns as the data source for training and validation. The reported performance metrics represent the average results obtained from the three testing sessions.

I. Evaluation of Robustness to Background Noise

Given signal variations in everyday settings, it is crucial to ensure the model performance under various external interference, such as different background noise. For experiments conducted in the lab, data quality can be controlled to minimize interference. To simulate unpredictable noise that may arise during actual usage, we tested the model performance against various levels of background noise added to the signals. Specifically, Gaussian noise was introduced to the testing dataset at different signal-to-noise ratio (SNR) levels, i.e., 10, 12.5, 15, 17.5, and 20 dB. For each defined SNR level, noise was added individually to all sEMG channels, ensuring that the signal in each channel adhered to the specified SNR constraints.

J. Statistical Analysis

In this study, repeated-measures analysis of variance (RM ANOVA) and paired t-tests were carried out when the compared groups satisfied the requirements for parametric analysis: 1) normality (assessed via the Shapiro–Wilk test); and 2) sphericity (evaluated via Mauchly's test for three or more groups). If requirements were not satisfied, we employed the Friedman test and Wilcoxon signed-rank test for nonparametric analysis. For multiple comparisons, the Holm–Bonferroni correction was applied, and only the adjusted *p*-values were reported. The significance level was set to 0.05.

IV. RESULTS

A. Comparisons With Deep Learning Techniques

For both the within-session and cross-session validations, the 2-D CNN models achieved better results than the 1-D CNN models. Therefore, features from the 2-D CNN were used as input data for the LSTM layer in the CNN-LSTM model. As shown in Fig. 6, the DF model achieved the best performance for both validation protocols. Specifically, under the within-session validation protocol, the DF can achieve the highest R^2 of 0.85 ± 0.015 and the lowest RMSE of $5.47\% \pm 0.35\%$MVC. Statistical analyses revealed that the DF model demonstrated a significantly higher R^2 compared with the other four models (all $p < 0.05$). In addition, the DF model achieved a significantly lower RMSE compared with the 1-D CNN and CNN-LSTM models (both $p < 0.05$). Similarly, the DF model can achieve the best cross-day performance, with a R^2 of 0.75 ± 0.029 and a RMSE of $6.72\% \pm 0.32\%$MVC. Statistical analyses demonstrated that the DF model significantly outperformed the 1-D CNN model and the CNN-SE model in both R^2 and RMSE (both $p < 0.05$).

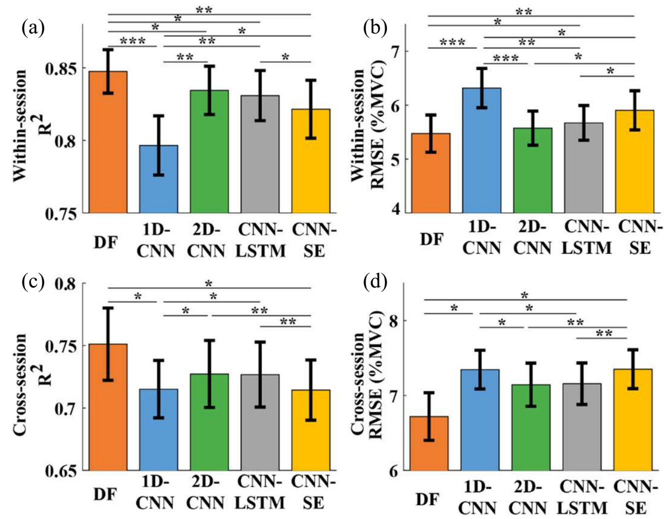

Fig. 6. Comparisons with deep learning techniques. (a) and (b) Present R^2 and RMSE values for within-session finger force predictions, respectively. (c) and (d) Present R^2 and RMSE values for cross-session finger force predictions, respectively. Error bars represent standard errors. *denotes $0.01 < p < 0.05$, **denotes $0.001 < p < 0.01$, ***denotes $p < 0.001$.

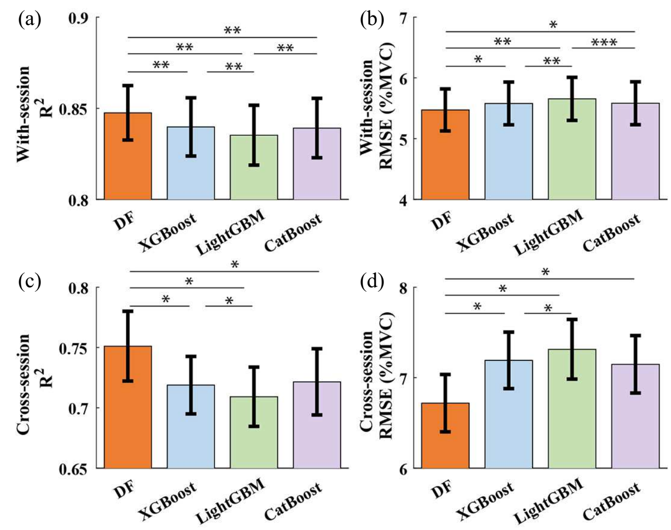

Fig. 7. Comparisons with different tree-based techniques. (a) and (b) Present R^2 and RMSE values for within-session finger force predictions, respectively. (c) and (d) Present R^2 and RMSE values for cross-session finger force predictions, respectively. Error bars represent standard errors. *denotes $0.01 < p < 0.05$, **denotes $0.001 < p < 0.01$, ***denotes $p < 0.001$.

B. Comparisons With Tree-Based Ensemble Methods

As shown in Fig. 7, the DF framework consistently achieved superior performance for both within-session and cross-session evaluations. In the within-session validation protocol, one-way RM ANOVA revealed a significant effect of method on prediction performance in the terms of R^2 [$F(3, 21) = 17.43$, $p < 0.001$, Fig. 7(a)] and RMSE [$F(3, 21) = 18.55$, $p < 0.001$, Fig. 7(b)]. Further pair-wise comparisons indicated that the DF model achieved significantly higher R^2 values compared to XGBoost, LightGBM, and CatBoost (all $p < 0.01$). Similarly, DF yielded a significantly lower RMSE than the three ensemble approaches (all $p < 0.05$). In the cross-session validation protocol, one-way RM ANOVA also revealed a significant effect of method on decoding performance in terms of R^2 [$F(3, 21) = 11.08$, $p < 0.001$, Fig. 7(c)] and RMSE [$F(3, 21) = 12.14$, $p < 0.001$, Fig. 7(d)]. Posthoc pairwise comparisons showed that the DF model significantly outperformed XGBoost, LightGBM, and CatBoost, achieving higher R^2 values and lower RMSE scores (all $p < 0.05$).

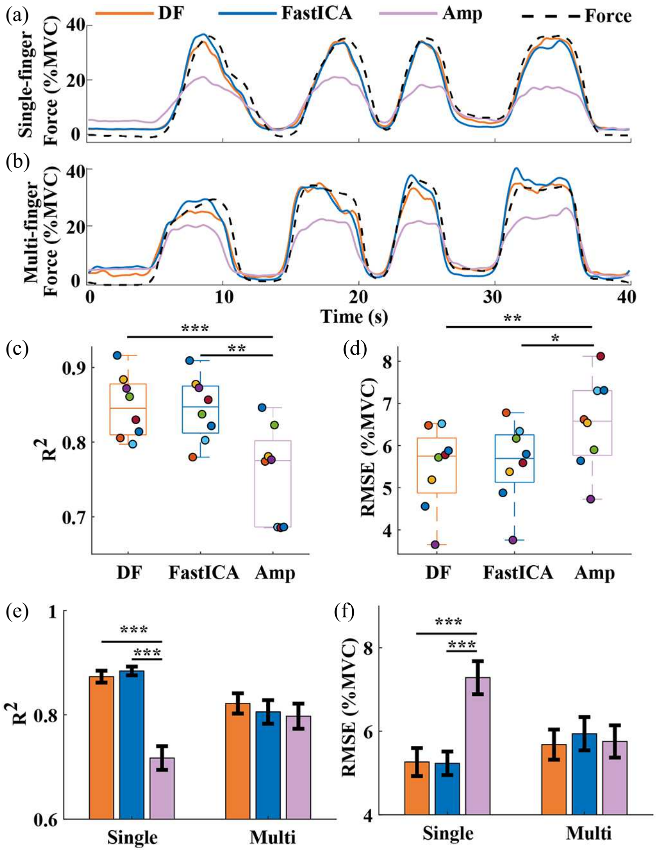

Fig. 8. Within-session finger force predictions using the DF, FastICA, and sEMG-amplitude (amp) approaches. (a) and (b) Present a representative predicted index finger force under the single- and multifinger tasks, respectively. (c) and (d) Show the finger force prediction evaluation in R^2 and RMSE, respectively. Dots of the same color indicate results from the same individual subjects. (e) and (f) Detail the performance of finger force predictions under the single- and multifinger tasks in R^2 and RMSE, respectively. Error bars represent the standard errors. *denotes $0.01 < p < 0.05$, **denotes $0.001 < p < 0.01$, ***denotes $p < 0.001$.

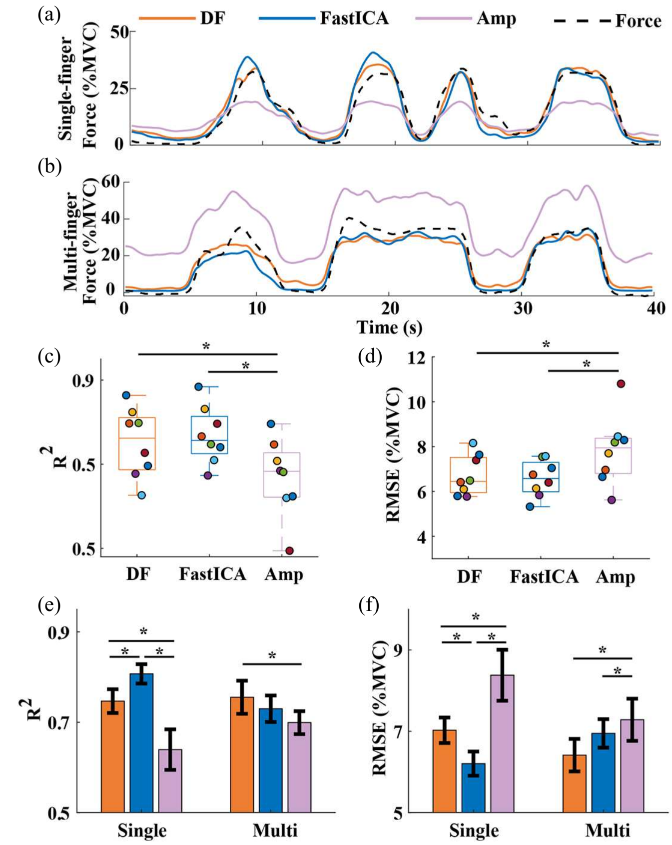

Fig. 9. Cross-session finger force predictions using the DF, FastICA, and sEMG-amplitude (amp) approaches. (a) and (b) Present a representative predicted index finger force under the single- and multifinger tasks, respectively. (c) and (d) Show the finger force prediction evaluation in R^2 and RMSE, respectively. Dots of the same color indicate results from the same individual subjects. (e) and (f) Detail the performance of finger force predictions under the single- and multifinger tasks in R^2 and RMSE, respectively. Error bars represent the standard errors. *denotes $0.01 < p < 0.05$, **denotes $0.001 < p < 0.01$, ***denotes $p < 0.001$.

C. Comparisons With FastICA and sEMG-Amplitude Techniques

Since our learning targets (neuronal firing representations) were sourced from the FastICA-based approach, we conducted a direct comparison with the FastICA method and the sEMG-amplitude approach for finger force predictions.

1) Within-Session Finger Force Predictions: Fig. 8(a) and (b) shows the representative predicted finger forces of the index finger under the single- and multifinger tasks, respectively. In both scenarios, the DF and FastICA approaches can accurately predict the measured finger forces. In contrast, the forces of the sEMG-amplitude approach deviated from the measured finger forces.

As shown in Fig. 8(c) and (d), the DF, FastICA, and sEMG-amplitude achieved a R^2 of 0.85 $\pm$ 0.015, 0.84 $\pm$ 0.015, and 0.76 $\pm$ 0.022, respectively. Correspondingly, the RMSE values were 5.47% $\pm$ 0.35%MVC (DF), 5.59% $\pm$ 0.33%MVC (FastICA), and 6.52% $\pm$ 0.38%MVC (Amp), respectively. Statistical analyses revealed that both the DF and FastICA significantly outperformed the sEMG-amplitude approach in R^2 and RMSE (all $p < 0.05$). No significant differences were detected between the DF and FastICA approaches in R^2 and RMSE. Figs. 8(e) and 9(f) further presented the finger force prediction performances under the single- and multifinger tasks. For the single-finger tasks, DF and FastICA models significantly outperformed the sEMG-amplitude approach in both R^2 and RMSE (all $p < 0.001$). For the multifinger tasks, no significant differences among the three approaches were detected in either R^2 or RMSE.

2) Cross-Session Finger Force Predictions: In the cross-session validation protocol, the DF and FastICA can also accurately predict the measured finger forces under both single- and multifinger tasks, as evidenced by the representative predicted finger forces of the index finger in Fig. 9(a) and (b). In contrast, the predicted force of the sEMG-amplitude approach showed large deviations from the measured finger forces, indicating poor generalizability in the cross-session context.

As shown in Fig. 9(c) and (d), compared with the sEMG-amplitude approach, the DF and FastICA can achieve a higher R^2 and a lower RMSE. Furthermore, statistical analyses demonstrated that the R^2 achieved by the DF (0.75 $\pm$ 0.029)

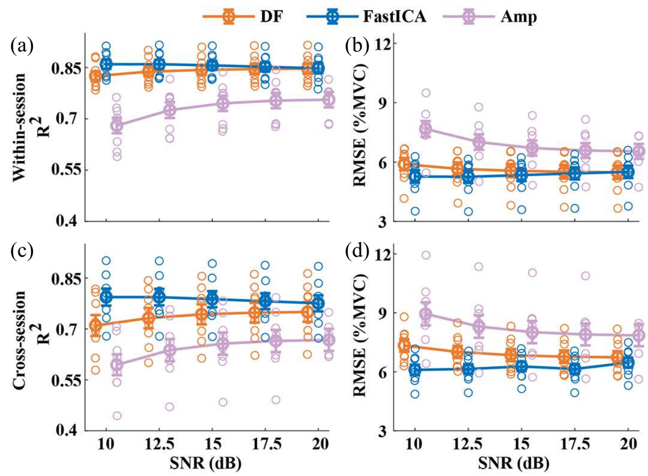

Fig. 10. Evaluation of finger force prediction performance under different noise levels. (a) and (b) Present R^2 and RMSE values for within-session finger force predictions, respectively. (c) and (d) Present R^2 and RMSE values for cross-session finger force predictions, respectively. Hollow circles represent the average result for each subject, with lines connecting the average results of each method across varying noise levels. The positions of each method are slightly offset at different noise levels for better visualization.

and FastICA (0.77 ± 0.024) were significantly higher than that of the sEMG-amplitude approach (0.67 ± 0.033). Similarly, the RMSE achieved by the DF ($6.72\% \pm 0.32\%$MVC) and FastICA ($6.58\% \pm 0.28\%$MVC) were significantly lower than that of the sEMG-amplitude approach ($7.83\% \pm 0.54\%$MVC). In addition, we further analyzed the finger force performance for the single- and multifinger tasks. As shown in Fig. 9(e) and (f), the DF can achieve significantly better results than the sEMG-amplitude approach in both R^2 and RMSE (all $p < 0.05$).

D. Robustness to Noise

Fig. 10(a) and (b) presents the within-session finger force prediction performance for testing data with different SNRs. The DF and FastICA approaches exhibited consistent finger force prediction performance under different noise levels. In contrast, the sEMG-amplitude approach showed declines in prediction performance with decreasing SNR (especially from 15 to 10 dB), as reflected by the decreasing R^2 in Fig. 10(a) and increasing RMSE in Fig. 10(b). Similar to the within-session performance, the DF and FastICA achieved relatively stable cross-session performance under different SNR levels, as shown in Fig. 10(c) and (d). However, the sEMG-amplitude approach demonstrated a notable degradation in cross-session performance with decreasing SNR. Statistical analyses revealed that both DF and FastICA significantly outperformed the sEMG-amplitude approach under each SNR level (all $p < 0.01$).

E. Computational Efficiency Evaluation

Considering the importance of computational efficiency in ensuring real-time performance, we evaluated the processing times of different methods on the testing dataset. Pseudoreal-time testing was conducted by sequentially feeding data segments into the algorithm, simulating the buffer behavior of a real-time data acquisition system. All evaluations were conducted on an AMD Ryzen 7 6800H @ 3.2 GHz, with MATLAB R2023a (The MathWorks Inc., USA) serving as the implementation platform. The within-session and cross-session computational times for each method are summarized in Table II. As expected, the sEMG-amplitude method required the least computational time under both validation protocols. Compared with the FastICA approach, the DF model demonstrated significantly improved computational efficiency, with processing times reduced by approximately 75% for within-session testing and over 80% for cross-session testing. Statistical analyses revealed that the DF model took significantly less computational time than the FastICA model under both validation protocols (both $p < 0.01$).

TABLE II
COMPUTATIONAL TIME OF FORCE PREDICTION (MS)

	Within Session	Cross Session
DF	19.94 ± 1.45	18.54 ± 0.99
FastICA	77.44 ± 9.67	113.04 ± 11.96
Amp	1.66 ± 0.01	1.71 ± 0.02

V. DISCUSSION

In this study, we aimed to directly learn the neuronal firing information from sEMG signals, instead of employing BSS-based techniques, with the goal of enhancing computational efficiency while maintaining decoding accuracy. Specifically, we first obtained the neuronal firing representation (populational firing rate signal) from the FastICA approach as the learning target signals (i.e., training labels). Then, we directly learned the targets from sEMG signals via the DF model. We found that the DF model consistently outperformed other neural network based deep learning models. Compared with the FastICA approach, this new approach could accurately capture the underlying neural drive information encoded in the sEMG signals, as evidenced by the same level of finger force prediction performance but with significantly less computational time. The DF model demonstrated stable cross-session performance and robust performance at various SNR levels, highlighting its potential for long-term utility in different signal quality conditions without the need of model retraining.

Compared with commonly used deep learning techniques, the DF model outperformed them in both within-session and cross-session validation protocols, revealing that the layer-by-layer processing mechanism can effectively accommodate non-differentiable modules (forest-based modules) for learning neuronal firing representations. In addition, unlike gradient-based deep networks that rely heavily on large-scale labeled datasets and complex backpropagation training, the DF model employs ensemble learning structures that eliminate the need for differentiability and reduce dependency on extensive hyperparameter tuning. This design not only simplifies the training process but also enhances robustness in data-limited scenarios, making it suitable for neural decoding applications with limited data and computational resources [53], [54].

The consistently better performance of the DF model over XGBoost, LightGBM, and CatBoost highlights its suitability for decoding neural firing information from sEMG. Unlike conventional boosting methods that rely on shallow ensembles, DF employs a layer-wise cascade structure with feature augmentation, enabling progressive learning and rich feature representations. This hierarchical design is particularly effective in capturing the complex and variable patterns in neural-drive signals. Moreover, the integration of both random forests and completely random forests in each layer enhances model diversity and robustness, contributing to its superior generalization across sessions. While tree-based boosting methods required extensive parameter tuning, DF achieved better results with fewer adjustments, reflecting its stability and adaptability.

Although the inputs of the sEMG-amplitude approach and the DF model were the same, the DF model significantly outperformed the sEMG-amplitude approach under both the within-session and cross-session validation protocols. The inaccurate finger force predictions via the sEMG-amplitude approach could be attributed to its inherent limitations. Considering physiological factors, the muscle compartments of different fingers are spatially close and partially overlap when viewed from the skin surface. Correspondingly, muscle crosstalk occurs due to the overlapping activation of adjacent muscles, making it difficult to isolate signals corresponding to individual finger movements, thus degrading the finger force prediction performance. Additionally, the recorded sEMG signals can be interfered with motion artifacts introduced during muscle activities, compromising their quality and reliability. In long-term (cross-session) scenarios, the performance of the sEMG-amplitude approach further deteriorated. This decline was due to the nonstationary nature of sEMG signals, which could be affected by several factors including variations in electrode placement, changes in skin impedance, and different background noise [15]. These factors introduced inconsistencies in the recorded signals, making it difficult to establish a stable mapping between the sEMG amplitudes and the intended finger forces.

In contrast, the DF approach demonstrated superior performance in both within-session and cross-session scenarios, which is attributed to its reliance on neural firing representations for the interpretation of finger forces. Specifically, the ground-truth neural firing signals came from binary motoneuron discharge events. These binary discharge events were less affected by variations in sEMG signals, resulting in consistent neural firing signals. To obtain neural firing events, two distinct clusters were identified for each source signal during the sEMG decomposition process. The cluster with higher amplitude represented MU discharge events, while the baseline noise cluster was excluded from further analyses. This effective noise removal not only reduced interference but also enhanced robustness under varying noise conditions. In addition, the MU refinement procedure ensured that the ground-truth neural firing signals were specific to individual fingers, thus eliminating the influence of muscle crosstalk and enhancing the accuracy of force predictions.

The comparable performance between the DF and FastICA revealed the effective learning of neuronal firing representations via the DF model. By leveraging its hierarchical structure, the DF model could capture stable and robust neuronal firing representations for the interpretation of finger movements. In addition, unlike FastICA which relied on computationally intensive decomposition of sEMG signals to extract motoneuron discharge events, the DF model directly mapped sEMG amplitude features to neuronal firing representations. This approach ensured that the decoding accuracy remains high and that the computational time remained consistent across all conditions for the DF model, making it highly suitable for real-time applications.

Although we have demonstrated the feasibility of directly learning neuronal firing representations from sEMG signals using the DF model, further validations could be conducted to confirm its broader applicability and robustness. First, we only evaluated the extraction of neuronal firing representations in the context of finger force predictions. In the future, we plan to extend the approach to other hand motor tasks, such as joint kinematic prediction. Second, while this study demonstrated the robustness of the DF model across sessions and noise levels, it did not consider scenarios involving muscle fatigue, which can be validated in future work.

VI. Conclusion

We evaluated the direct learning of neuronal firing representations from sEMG signals using the DF model. The DF model achieved comparable accuracy to FastICA-based approaches in predicting dexterous finger forces while significantly reducing computational time. Moreover, the model demonstrated robust performance in both within-session and cross-session evaluations and remained stable under varying signal noise levels. This underscores its suitability for real-time applications where efficiency and consistency are crucial. The efficient nature of the DF model further enhanced its practicality, providing insights into its processes. These results underline the potential of the direct learning approach as a reliable tool for neural decoding in real-world scenarios.

References

[1] A. R. Sobinov and S. J. Bensmaia, “The neural mechanisms of manual dexterity,” *Nat. Rev. Neurosci.*, vol. 22, no. 12, pp. 741–757, 2021.

[2] K. J. Jyothish and S. Mishra, “A survey on robotic prosthetics: Neuroprosthetics, soft actuators, and control strategies,” *ACM Comput. Surv.*, vol. 56, no. 8, pp. 1–44, Aug. 2024.

[3] S. Funabashi et al., “Tactile transfer learning and object recognition with a multifingered hand using morphology specific convolutional neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7587–7601, Jun. 2024.

[4] C. Zeng, S. Li, Z. Chen, C. Yang, F. Sun, and J. Zhang, “Multifingered robot hand compliant manipulation based on vision-based demonstration and adaptive force control,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 5452–5463, Sep. 2022.

[5] H. Su, W. Qi, Z. Li, Z. Chen, G. Ferrigno, and E. De Momi, “Deep neural network approach in EMG-based force estimation for human–robot interaction,” *IEEE Trans. Artif. Intell.*, vol. 2, no. 5, pp. 404–412, May 2021.

[6] R. Liu, J. Wandeto, F. Nageotte, P. Zanne, M. De Mathelin, and B. Dresp-Langley, “Spatiotemporal modeling of grip forces captures proficiency in manual robot control,” *Bioengineering*, vol. 10, no. 1, 2023, Art. no. 59.

[7] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, “Gesture recognition in robotic surgery: A review,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 2021–2035, Jun. 2021.

[8] C. D. Nistor-Cseppento et al., “The outcomes of robotic rehabilitation assisted devices following spinal cord injury and the prevention of secondary associated complications,” *Medicina (Kaunas)*, vol. 58, no. 10, 2022, Art. no. 1447.

[9] S. Khalid, F. Alnajjar, M. Gochoo, A. Renawi, and S. Shimoda, “Robotic assistive and rehabilitation devices leading to motor recovery in upper limb: A systematic review,” *Disabil. Rehabil. Assist. Technol.*, vol. 18, no. 5, pp. 658–672, Jul. 2023.

[10] K.-S. Hong, N. Aziz, and U. Ghafoor, “Motor-commands decoding using peripheral nerve signals: A review,” *J. Neural Eng.*, vol. 15, no. 3, 2018, Art. no. 031004.

[11] S. Ma et al., “Conditional generative models for simulation of EMG during naturalistic movements,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 9224–9237, May 2025.

[12] Y. Guo et al., “sEMG-based inter-session hand gesture recognition via domain adaptation with locality preserving and maximum margin,” *Int. J. Neural Syst.*, vol. 34, no. 3, pp. 2450010–2450010, 2024.

[13] X. Jiang, K. Nazarpour, and C. Dai, “Explainable and robust deep forests for EMG-force modeling,” *IEEE J. Biomed. Health Inform.*, vol. 27, no. 6, pp. 2841–2852, Jun. 2023.

[14] D. W. O. Antillon, C. R. Walker, S. Rosset, and I. A. Anderson, “Glove-based hand gesture recognition for diver communication,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9874–9886, Dec. 2022.

[15] L. Meng et al., “User-tailored hand gesture recognition system for wearable prosthesis and armband based on surface electromyogram,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–16, 2022.

[16] Z. Wang et al., “Optimization of inter-subject sEMG-based hand gesture recognition tasks using unsupervised domain adaptation techniques,” *Biomed. Signal Process. Control*, vol. 92, 2024, Art. no. 106086.

[17] Z. Wang, L. Meng, Z. Wang, and W. Chen, “sEMG-based multi-DoF finger force modeling for user-tailored wearable prosthesis and armband applications,” in *Proc. IEEE 20th Int. Conf. Body Sensor Netw. (BSN)*. Piscataway, NJ, USA: IEEE Press, 2024, pp. 1–4.

[18] L. Meng and X. Hu, “Unsupervised neural decoding for concurrent and continuous multi-finger force prediction,” *Comput. Biol. Med.*, vol. 173, 2024, Art. no. 108384.

[19] S. Wang, H. Tang, B. Wang, and J. Mo, “A novel approach to detecting muscle fatigue based on sEMG by using neural architecture search framework,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4932–4943, Aug. 2021.

[20] D. Farina et al., “Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation,” *Nat. Biomed. Eng.*, vol. 1, no. 2, 2017, Art. no. 2.

[21] L. Meng et al., “Evaluation of decomposition parameters for high-density surface electromyogram using fast independent component analysis algorithm,” *Biomed. Signal Process. Control*, vol. 75, 2022, Art. no. 103615.

[22] T. Ma and X. Hu, “Adaptive motor unit decomposition using a cross-validation-based update policy,” *Comput. Biol. Med.*, vol. 193, 2025, Art. no. 110479.

[23] L. Meng and X. Hu, “Robust and lightweight decoder for unsupervised multi-finger force predictions towards the internet of medical things-based applications,” *IEEE Internet Things J.*, vol. 12, no. 9, pp. 12547–12561, Sep. 2024.

[24] L. Meng and X. Hu, “Unsupervised neural decoding to predict dexterous multi-finger flexion and extension forces,” *IEEE J. Biomed. Health Inform.*, vol. 29, no. 3, pp. 1959–1969, Mar. 2025.

[25] Y. Zheng and X. Hu, “Concurrent estimation of finger flexion and extension forces using motoneuron discharge information,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 5, May 2021, Art. no. 5.

[26] P. Sedighi, X. Li, and M. Tavakoli, “EMG-based intention detection using deep learning for shared control in upper-limb assistive exoskeletons,” *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 41–48, Jan. 2024.

[27] J. Zbinden, J. Molin, and M. Ortiz-Catalan, “Deep learning for enhanced prosthetic control: Real-time motor intent decoding for simultaneous control of artificial limbs,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 1177–1186, 2024.

[28] A. Wahid, K. Ullah, S. Irfan Ullah, M. Amin, S. Almutairi, and M. Abohashrh, “sEMG-based upper limb elbow force estimation using CNN, CNN-LSTM, and CNN-GRU models,” *IEEE Access*, vol. 12, pp. 128979–128991, 2024.

[29] M. Abdar et al., “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Inf. Fusion*, vol. 76, pp. 243–297, 2021.

[30] M. H. Rafiei, L. V. Gauthier, H. Adeli, and D. Takabi, “Self-supervised learning for electroencephalography,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 1457–1471, Feb. 2022.

[31] Z.-H. Zhou and J. Feng, “Deep forest,” *Natl. Sci. Rev.*, vol. 6, no. 1, pp. 74–86, 2019.

[32] Y. Zheng and X. Hu, “Concurrent prediction of finger forces based on source separation and classification of neuron discharge information,” *Int. J. Neural Syst.*, vol. 31, no. 6, 2021, Art. no. 6.

[33] Y. Zheng and X. Hu, “Interference removal from electromyography based on independent component analysis,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, 2019, Art. no. 5.

[34] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Netw.*, vol. 13, nos. 4–5, 2000, Art. no. 4–5.

[35] H. Zhao, X. Zhang, M. Chen, and P. Zhou, “Adaptive online decomposition of surface EMG using progressive FastICA peel-off,” *IEEE Trans. Biomed. Eng.*, vol. 71, no. 4, pp. 1257–1268, Apr. 2024.

[36] M. Orlandi et al., “Real-time motor unit tracking from sEMG signals with adaptive ICA on a parallel ultra-low power processor,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 18, no. 4, pp. 771–782, Aug. 2024.

[37] E. A. Clancy and N. Hogan, “Probability density of the surface electromyogram and its relation to amplitude detectors,” *IEEE Trans. Biomed. Eng.*, vol. 46, no. 6, 1999, Art. no. 6.

[38] R. Roy, F. Xu, D. G. Kamper, and X. Hu, “A generic neural network model to estimate populational neural activity for robust neural decoding,” *Comput. Biol. Med.*, vol. 144, 2022, Art. no. 105359.

[39] J. Fan, L. Vargas, D. G. Kamper, and X. Hu, “Robust neural decoding for dexterous control of robotic hand kinematics,” *Comput. Biol. Med.*, vol. 45, Jun. 2023, Art. no. 107139.

[40] L. Meng, L. Vargas, D. G. Kamper, and X. Hu, “Real-time myoelectric-based neural-drive decoding for concurrent and continuous control of robotic finger forces,” *IEEE Trans. Hum.-Mach. Syst.*, vol. 55, no. 2, pp. 256–265, Apr. 2025.

[41] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*,” 2014.

[42] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*. San Francisco, CA, USA: ACM, Aug. 2016, pp. 785–794.

[43] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017.

[44] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: Unbiased boosting with categorical features,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Curran Associates, Inc., 2018.

[45] H.-Y. Yang, Y.-S. Han, and C.-S. Nam, “Study on finger gesture interface using one-channel EMG,” *IEEE Access*, vol. 13, pp. 9606–9614, 2025.

[46] Z. Lu, S. Chen, J. Yang, C. Liu, and H. Zhao, “Prediction of lower limb joint angles from surface electromyography using XGBoost,” *Expert Syst. Appl.*, vol. 264, Mar. 2025, Art. no. 125930.

[47] H. Wang, Q. Tao, and X. Zhang, “Ensemble learning method for the continuous decoding of hand joint angles,” *Sensors*, vol. 24, no. 2, Jan. 2024, Art. no. 660.

[48] W. J. Hans and N. M. Lindsay “Hyperparameter tuning of light gradient boosting machine for electromyography signal classification,” *Eng. Res. Exp.*, vol. 7, no. 2, Jun. 2025, Art. no. 025237.

[49] A. K. Sharma, S.-H. Liu, X. Zhu, and W. Chen, “Predicting gait parameters of leg movement with sEMG and accelerometer using catboost machine learning,” *Electronics*, vol. 13, no. 9, May 2024, Art. no. 1791.

[50] A. Fougner, Ø. Stavdahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, “Control of upper limb prostheses: Terminology and proportional myoelectric control—A review,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 5, pp. 663–677, May 2012.

[51] J. G. Ngeo, T. Tamei, and T. Shibata, “Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model,” *J. Neuroeng. Rehabil.*, vol. 11, no. 1, Dec. 2014, Art. no. 122.

[52] A. Ameri, E. J. Scheme, E. N. Kamavuako, K. B. Englehart, and P. A. Parker, “Real-time, simultaneous myoelectric control using force and position-based training paradigms,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 2, Feb. 2013, Art. no. 2.

[53] X. Chen, P. Wang, Y. Yang, and M. Liu, “Resource-constraint deep forest based intrusion detection method in internet of things for consumer electronic,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 2, pp. 4976–4987, May 2024.

[54] H. Bangui and B. Buhnova, “Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms,” *Comput. Electr. Eng.*, vol. 100, 2022, Art. no. 107901.