

Classification of Electromyographic Hand Gesture Signals Using Modified Fuzzy C-Means Clustering and Two-Step Machine Learning Approach

Guangyu Jia^{ID}, Hak-Keung Lam^{ID}, *Fellow, IEEE*, Shichao Ma, Zhaohui Yang^{ID}, Yujia Xu^{ID}, and Bo Xiao^{ID}

Abstract—Understanding and classifying electromyogram (EMG) signals is of significance for dexterous prosthetic hand control, sign languages, grasp recognition, human-machine interaction, etc.. The existing research of EMG-based hand gesture classification faces the challenges of unsatisfied classification accuracy, insufficient generalization ability, lack of training data and weak robustness. To address these problems, this paper combines unsupervised and supervised learning methods to classify an EMG dataset consisting of 10 classes of hand gestures. To lessen the difficulty of classification, clustering methods including subtractive clustering and fuzzy c-means (FCM) clustering algorithms are employed first to obtain the initial partition of the inputs. In particular, modified FCM algorithm is proposed to accustom the conventional FCM to the multi-class classification problem. Based on the grouping information obtained from clustering, a type of two-step supervised learning approach is proposed. Specifically, a top-classifier and three sub-classifiers integrated with windowing method and majority voting are employed to accomplish the two-step classification. The results demonstrate that the proposed method achieves 100% test accuracy and the strongest robustness compared to the conventional machine learning approaches, which shows the potential for industrial and healthcare applications, such as movement intention detection, grasp recognition and dexterous prostheses control.

Index Terms—EMG signals, clustering, hand gesture classification, machine learning, deep learning, fuzzy c-means (FCM).

Manuscript received February 14, 2020; revised April 3, 2020; accepted April 6, 2020. Date of publication April 9, 2020; date of current version June 5, 2020. This work was supported in part by King's College London and the China Scholarship Council and has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service through the EPSRC Tier-2 capital under Grant EP/P020259/1. (Corresponding author: Hak-Keung Lam.)

Guangyu Jia, Hak-Keung Lam, Shichao Ma, and Yujia Xu are with the Centre for Robotics Research, Department of Engineering, King's College London, London WC2R 2LS, U.K. (e-mail: guangyu.jia@kcl.ac.uk; hak-keung.lam@kcl.ac.uk; hzmashichao@163.com; yujia.xu@kcl.ac.uk).

Zhaohui Yang is with the Centre for Telecommunications Research, Department of Engineering, King's College London, London WC2R 2LS, U.K. (e-mail: yang.zhaohui@kcl.ac.uk).

Bo Xiao is with The Hamlyn Centre for Robotic Surgery, Imperial College London, London SW7 2AZ, U.K. (e-mail: b.xiao@imperial.ac.uk).

Digital Object Identifier 10.1109/TNSRE.2020.2986884

I. INTRODUCTION

THE electromyogram (EMG) signals from human muscle are widely used to identify neuromuscular diseases. Apart from medical and diagnostic applications, the recognition of patterns of EMG signals generated by different movements has great values and potential in Biomechanics, Kinesiology, Mechanical bionics, etc.. One of the most successful and significant applications is the EMG controlled prosthesis, which is beneficial for amputees and patients who have difficulties to make specific movements. Understanding and classifying EMG signals are also helpful for recognising gestures, sign languages, human-machine interaction [1], etc..

Recently, the machine learning based approaches have been employed in classification of EMG signals [2]–[6]. To overcome the disadvantages of human interpretation of physiological signals such as dependency on expert knowledge and inevitable human errors, the computer supported signal analysis including signal processing, feature extraction, classification, etc. were developed and various on-line and off-line systems have been established [7]–[10]. Rami N. Khushaba et al. collected a series of EMG signals having different number of finger movements and channels and proposed a variety of EMG signal processing techniques [11]–[14]. This paper adopts the dataset provided by Rami N. Khushaba et al. in [15] which includes 10 single and combine finger movements recorded through 2 electrodes. K. Anam et al. used the recurrent neural network on this dataset and achieved 96.3% validation accuracy. Rami N. Khushaba et al. employed k-nearest neighbors (KNN) algorithm, support vector machine and Bayesian fusion methods on this dataset which achieved 92.0% offline accuracy. Also, with this dataset, K. Anam et al. proposed system employing spectral regression discriminant analysis for dimensionality reduction, extreme learning machine for classification and the majority vote for the classification smoothness, which was able to recognize the individual and combined fingers movements with up to 98.0% classification accuracy [16].

Having reviewed the success of the reported results, the existed study on EMG-based hand gesture classification has some limitations and challenges. At first, EMG signals are prone to contaminations caused by real-world factors such as electrode shift and muscle fatigue. Small changes of the EMG traces may result in great impacts on classification results

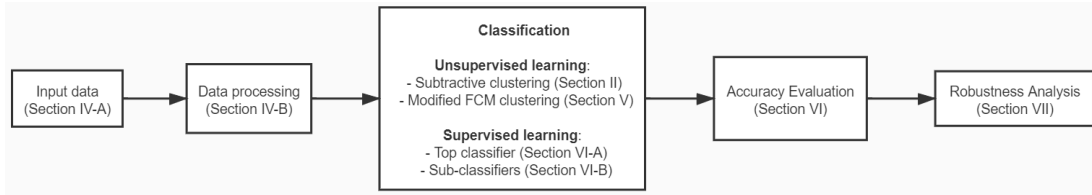


Fig. 1. Diagram of the paper's structure.

[17], [18]. Also, due to the limitation of data acquisition, the number of available data tends to be insufficient for deep learning's training [15]. To compensate the weak robustness of the established models and the insufficient training data, more data are employed for training in some literatures [17], [18], which further complicates the classification. In addition, the trained models tend to be subject-specific and are not easy to be generalized to other datasets, since different subjects have different data characteristics such as contractions' strength, skin thickness and muscular mass [18].

This paper attempts to solve the relevant challenges including model's weak generalization capability, lack of training data, weak robustness and unsatisfied classification performance with the dataset [15] mentioned above by Rami N. Khushaba. The dataset is used for the reason that it includes 10 single and combined figure movements, two electrodes was used when data collection which has less intrusion and lower computational cost, besides, the dataset has been used in the above literature which can be regarded as the benchmark dataset for comparison. Unsupervised and supervised learning methods are used to handle the classification of EMG signals. The main classification procedures are shown in Fig. 1. Data preprocessing is conducted first to de-noise and prepare data for classifiers. Then unsupervised learning approaches including two clustering methods are adopted to obtain the initial partition of the dataset. Based on the proximal information provided by clustering methods, supervised learning is then employed to implement the classification. The experimental results demonstrate that the proposed method achieves 100% test accuracy as well as the strongest robustness compared to conventional machine learning methods. The contributions of this paper are summarised as follows:

- Modifying the algorithm of fuzzy c-means (FCM) clustering to accustom to the problem of multi-class classification.
- Two-step classification method is designed to lessen the difficulty of model's training compared to the one-step classification.
- The generalization ability of the proposed model is improved by introducing majority voting at two classification stages.
- The proposed model obtains 100% test accuracy and the strongest robustness compared to five widely used machine learning methods, which shows the feasibility of practical application.
- Auxiliary data preprocessing methods including feature extraction, normalization as well as windowing method are used for different classifiers.
- Comprehensive comparisons with conventional machine learning methods including feed-forward neural network

(FNN), CNN, decision tree, random forest, KNN are conducted.

The structure of the rest of this paper is as follows. Section II introduces the preliminaries used in this paper. The overall picture of the experiment is illustrated in Section III. Data information and processing methods are elaborated in Section IV. The unsupervised learning algorithm is introduced in Section V. Section VI describes the supervised learning approaches, the classification procedures and the obtained results. The robustness analysis is presented in Section VII. Section VIII draws the conclusion.

II. PRELIMINARIES

In this section, the preliminaries about the algorithms used in this paper are presented.

A. Subtractive Clustering

To determine the appropriate number of clusters and the initial value of the cluster centers, subtractive clustering algorithm, which is an extension of the mountain clustering method proposed in [19], is adopted in this paper. Assumed that the dataset X consists of n samples awaited clustering, i.e., $X = (x_1, x_2, \dots, x_n)$. With subtractive clustering, it is assumed that each sample x_i is a potential cluster center and possesses the density of surrounding samples as follows [20]:

$$D_i = \sum_{z=1}^n e^{-\alpha \|x_i - x_z\|^2}, \quad \alpha = \frac{4}{r_a^2}, \quad (1)$$

where r_a is a positive constant. Obviously, if x_i has lots of neighbouring samples, it will have a high density value. Calculate the density of all samples and the data point with the highest density D_{c_1} is selected to be the first cluster center x_1^* . To avoid that the neighbouring data points of cluster center become a new cluster center, all samples' density is updated by

$$D_i = D_i - D_{c_1} e^{-\beta \|x_i - x_1^*\|^2}, \quad \beta = \frac{4}{r_b^2}, \quad (2)$$

where r_b is a positive constant r_b is generally greater than r_a . Repeat the above procedures of obtaining new cluster center and modifying densities until

$$D_{c_k} < \epsilon D_{c_1},$$

where ϵ is a specific fraction [20].

B. Other Machine Learning Models

To compare the classification performance of the proposed model, other conventional machine learning models in the literature including FNN [21], random forest [22], decision tree [23] and KNN [24], [25] are also employed in this paper.

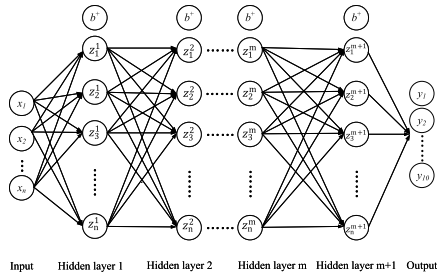


Fig. 2. The structure of FNN.

Particularly, when training FNN, the optimization algorithms used in this paper including Levenberg-Marquardt (LM) method, conjugate gradient backpropagation with Powell-Beale restarts and adaptive moment estimation (Adam), in which LM algorithm achieved the best performance in most cases of the experiments in this paper. The update rule of LM algorithm is given below [26].

$$W_{k+1} = W_k - (J_k^T J_k + \lambda I)^{-1} J_k e_k, \quad (3)$$

where k is the index of iterations, W_k represents the weights to be updated, J_k denotes the Jacobian matrix, λ is the combination coefficient, I is the identity matrix and e_k denotes the error vector containing the output errors for each input vector [26]. An example of FNN structure used in this paper is shown in Fig. 2, where there are $m + 1$ hidden layers with n nodes in each layer and the number of classes is N .

III. OVERALL IMPLEMENTATION PROCEDURES

The overall working principle of this paper is illustrated as follows.

In the clustering process, all training inputs are processed by normalization (Section IV-B.1) and feature extraction (Section IV-B.2). The processed data are first clustered by subtractive clustering (Section II-A, V) which determines the number of clusters and the initial values of the cluster centres. The obtained cluster numbers and the values of cluster centres are then used by FCM algorithm which aims to offer the explicit partition of the dataset by means of updating the clusters' centers and the membership degree according to the optimization procedures (Section V). As each data point has been given a targeted class, FCM is modified to be able to provide the same cluster number for all samples in the same class.

In the classification process, the classifiers in Fig. 3 takes training samples as inputs which are processed by normalization (Section IV-B.1) and windowing method (Section IV-B.3) to generate the windowing data. When the samples get into the top classifier, as three clusters and the corresponding classes are determined by the clustering methods, the top classifier will classify the windowing data into three classes (Section V). The outputs of the top classifier indicate which sub-classifier to be used for the further classification, and they will be processed by majority voting which aims to rectify the classification results of windowing samples. Next, the windowing samples will be classified by the corresponding sub-classifier, and the classification results of sub-classifiers

will be processed by majority voting. The final predicted outputs are then modified to be the corresponding classes, i.e., the predicted results 1, 2, 3, 4 by the first sub-classifier indicate the corresponding original classes 1, 5, 7, 9, similarly for the results obtained by other sub-classifiers.

At the test stage, each sample processed by normalization and windowing method will go through the top classifier first and obtain a cluster (group) number $i, i \in \{1, 2, 3\}$, then the i -th sub-classifier will be used accordingly to provide the final classification result of this data point.

The overall structure of the implementation procedures is demonstrated in Fig. 3. The following sections elaborate the details of the experiment.

IV. DATA INFORMATION AND PROCESSING

A. Data Information

The data used in this paper is the public dataset in [15] by Rami N. Khushaba. 8 subjects aged between 20 and 35 years old without neurological or muscle disorders were recruited to execute the required order of the finger movements. Ten classes of fingers movements shown in Fig. 4 are collected: Thumb (T), Index (I), Middle (M), Ring (R), Little (L) and the pinching of combined Thumb-index (TI), Thumb-Middle (TM), Thumb-Ring (TR), Thumb-Little (TL), and finally the hand close (HC). The data information is summarized in Table I.

1) *Advantages of the Dataset*: This dataset [15] is selected for the reason that it detected single and combined finger movements with just two electrodes, which leads to less intrusion, lower computational load and would make the prosthesis design more dexterous and flexible. Plenty of other related datasets either used more than two EMG electrodes for data collection [27]–[30] or did not consider combined finger movements [31], [32].

2) *Sampling Method*: The sampling method used in this paper is the stratified sampling which is implemented by dividing the dataset into subgroups and selecting the samples from each group. As there are 10 classes of hand gestures, to make the inputs contain the balanced amount of samples from each class, we randomly select training, validation and test samples from each class and combine them to form the final datasets. Thus the classifier is fed by the data consisting of the equal number of samples from each class.

B. Data Processing

As shown in Fig. 3, the input data will go through the data processing procedures including normalization, which is introduced in Section IV-B.1, feature extraction in Section IV-B.2 and windowing in Section IV-B.3.

1) *Normalization*: To de-noise and eliminate the large difference of signal amplitude between different subjects, the input signals of each subject were rescaled to the range $[0, 1]$ through

$$y_N = \frac{y - y_{\min}}{y_{\max} - y_{\min}}, \quad (4)$$

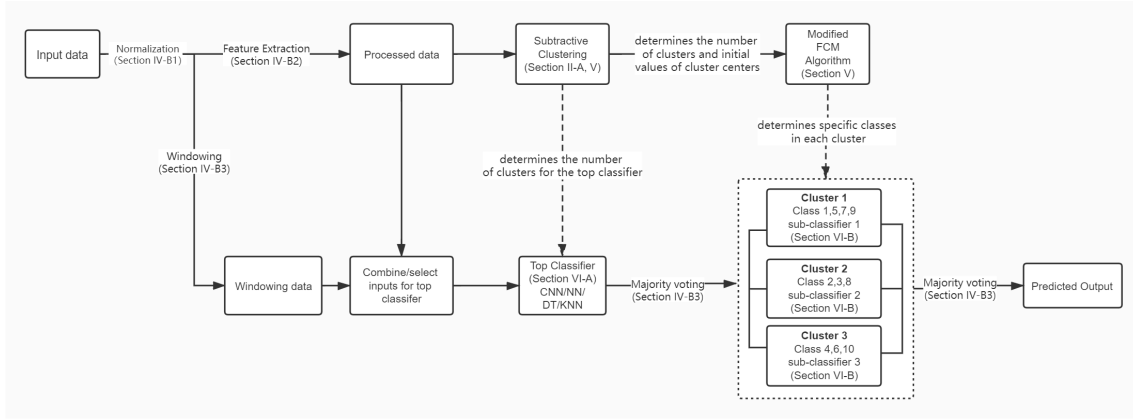


Fig. 3. Diagram of the overall structure of the experiment. The solid lines indicate the process of the implementation and the dashed lines represent the auxiliary information provided for training.

TABLE I
DATA INFORMATION

Sampling rate	Sampling time	Number of electrodes	Sample's size	Number of subjects	Number of samples
4000 Hz	5 seconds	2	20000×2	8	480 (training: 320, test: 160)



Fig. 4. Ten classes of finger movements. The sub-figures from left to right correspond to the single finger movements (Thumb (T), Index (I), Middle (M), Ring (R) and Little (L) in this first line) and combined finger movements (ThumbIndex (TI), ThumbMiddle (TM), ThumbRing (TR), ThumbLittle (TL) and hand close (HC) in this second line), respectively.

where y is the signal to be normalized, y_N is the normalized signal, y_{\min} and y_{\max} denote the minimum and maximum elements in y , respectively.

2) Feature Extraction: Feature extraction is used to extract important information from the raw data and reduce the redundancy. After trying more than 30 time-domain and frequency-domain features as well as their combinations [33]–[35], the two features which achieve the best performance were selected, i.e., in time domain, the sum of absolute value (SAV) of each channel:

$$\text{SAV} = \sum_{i=1}^L |e_i|, \quad (5)$$

where e_i represents the EMG signal in a segment i , L denotes length of the whole signal and $|\cdot|$ represents the absolute value of the signal. The second feature is obtained in frequency domain, that is, the standard deviation (STD) of each channel:

$$\text{STD} = \sqrt{\frac{1}{L-1} \sum_{i=1}^L (|F(e_i)| - \mu)^2},$$

$$\mu = \frac{1}{L} \sum_{i=1}^L |F(e_i)|, \quad (6)$$

where $F(\cdot)$ denotes the Fourier transform of the signal.

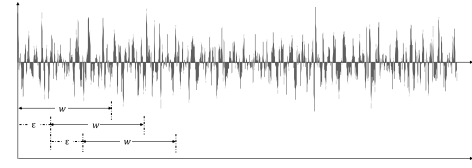


Fig. 5. Windowing scheme with window size w and increment ϵ .

3) Windowing Scheme & Majority Voting: As shown in Fig. 5, windowing scheme is implemented by sliding a window of the size w with increment ϵ over the original sample to cut the sample before and after the window while not modifying the contents within the window [36], which has proved to be effective for deep learning training [37]. Through this operation, subsets of original samples can be obtained. Windowing method is adopted for the following reasons. First, the complete sample is too large to process and contains redundancy. Secondly, in the scenarios that data are insufficient for training (we just have 480 EMG samples in implementation), this method provides the large number of input data and can force the CNN to learn features from each windowing data rather than the complete sample, which was shown efficient and provided satisfactory performance for deep learning training [37]. Thirdly, the instantaneous sample of the EMG signal has incomplete information of the overall muscle activity. We found that the integration of windowing method and majority voting can improve the accuracy by means of increasing the training samples and providing comprehensive decision making procedure.

Denote the number of newly generated samples by n_1 , original data length by n_2 , window size by n_3 and the window increment by n_4 , the number of generated samples can be computed as:

$$n_1 = (n_2 - n_3)/n_4 + 1.$$

After training, the new generated samples will be given predicted labels. Majority voting is used at this stage to gather the predicted information of the newly generated samples and derive the final results of all samples. For example, as the length of sliding window in this paper is set to 500 with increment 10, each sample can generate 1951 new samples. After training, majority voting is used to obtain the predicted label for each sample from 1951 corresponding results. If 1500 newly generated samples are classified into class 1, 100 samples are in class 7 and the rest 351 samples are in class 3, after majority voting, the label of the original sample should be class 1.

As there are two classification stages in implementation, voting has been conducted twice at two stages. The first voting is conducted for the top classifier to rectify the predicted results. Then, voting is implemented again after the sub-classifiers to further improve the classification performance.

V. UNSUPERVISED LEARNING APPROACH

Unsupervised learning approach is adopted beforehand to alleviate the difficulty and complication of classification. As mentioned above, we use clustering method including subtractive clustering and FCM to analyze the commonality of the data and group the data into specific number of groups for further classification. Subtractive clustering is commonly used when not knowing how many clusters there should be for a specific dataset and what the initial values of the cluster centers are. Therefore, to find out the number of clusters as well as the cluster centers for the EMG dataset, subtractive clustering is employed first. Then we use FCM to update the cluster centers and provide the degree that each sample belongs to a cluster, where the degree is described by a membership grade. A general FCM is based on minimizing the objective function as follows:

$$\min J_m(U, C) = \sum_{i=1}^S \sum_{j=1}^N \mu_{ij}^m \|x_j - c_i\|^2, \quad (7a)$$

$$\text{s.t. } \sum_{i=1}^S \mu_{ij} = 1, \quad j = 1, 2, \dots, N, \quad (7b)$$

where x_j represents the j -th sample, μ_{ij} denotes the membership degree with respect to the sample x_j in the i -th cluster. m represents the fuzzy overlap degree, c_i denotes the center of the i -th cluster. $U = \{\mu_{ij}\}$, $C = \{c_i\}$, $j = 1, 2, \dots, N$, $i = 1, 2, \dots, S$ in which N is the number of samples and S is the number of clusters. Based on the FCM algorithm, the cluster centers and the degree of membership are updated through

$$c_i = \frac{\sum_{j=1}^N x_j \mu_{ij}^m}{\sum_{j=1}^N \mu_{ij}^m}, \quad \mu_{ij} = \frac{1}{\sum_{p=1}^S \left(\frac{\|x_j - c_i\|}{\|x_j - c_p\|} \right)^{\frac{2}{m-1}}}. \quad (8)$$

However, the problem is that the FCM clustering groups the data only based on the features inherent in the EMG signals and does not take into account the corresponding labels, thus, the data points belong to the same class might be grouped into different clusters through FCM, which is not supposed to

happen. Therefore, we cannot directly use the solution (8) for the 10-class dataset. In order to tackle this problem, i.e., taking samples' labels into consideration, making use of the label information when clustering and making the samples in the same class share the same cluster number, the modified FCM clustering is proposed as follows.

Assume that there are P classes of samples, and each class has Q samples, then we have $N = P \times Q$. As each sample will be assigned to a specific cluster by using FCM method, the problem of using the above objective function is that the samples in the same class might be assigned to different clusters. To avoid this problem, i.e., to make the Q samples which belong to the same class are in the same cluster, J_m is modified as follows.

$$\min J_m(U, C) = \sum_{i=1}^S \sum_{j=1}^P \mu_{ij}^m \sum_{k=1}^Q \|x_{kj} - c_i\|^2, \quad (9a)$$

$$\text{s.t. } \sum_{i=1}^S \mu_{ij} = 1, \quad j = 1, 2, \dots, P. \quad (9b)$$

The Lagrangian function of problem (9) is

$$F(\lambda, U, C) = \sum_{i=1}^S \sum_{j=1}^P \mu_{ij}^m \sum_{k=1}^Q \|x_{kj} - c_i\|^2 + \sum_{j=1}^P \lambda_j \left(\sum_{i=1}^S \mu_{ij} - 1 \right). \quad (10)$$

Then, μ_{ij} and c_i are obtained as follows. First, calculate the derivative of F with respect to μ_{ij}

$$\frac{\partial F(\lambda, U, C)}{\partial \mu_{ij}} = m \cdot \mu_{ij}^{m-1} \sum_{k=1}^Q \|x_{kj} - c_i\|^2 + \lambda_j = 0. \quad (11)$$

Obviously,

$$\mu_{ij} = \frac{1}{\left(\sum_{k=1}^Q \|x_{kj} - c_i\|^2 \right)^{\frac{1}{m-1}}} \cdot \left(\frac{-\lambda_j}{m} \right)^{\frac{1}{m-1}}. \quad (12)$$

Since

$$\sum_{i=1}^S \mu_{ij} = 1, \quad (13)$$

substituting Eq. (12) into Eq. (13), we obtain the equation as follows:

$$\sum_{i=1}^S \frac{1}{\left(\sum_{k=1}^Q \|x_{kj} - c_i\|^2 \right)^{\frac{1}{m-1}}} \cdot \left(\frac{-\lambda_j}{m} \right)^{\frac{1}{m-1}} = 1. \quad (14)$$

Obviously it follows,

$$\left(\frac{-\lambda_j}{m} \right)^{\frac{1}{m-1}} = \left(\sum_{i=1}^S \frac{1}{\left(\sum_{k=1}^Q \|x_{kj} - c_i\|^2 \right)^{\frac{1}{m-1}}} \right)^{-1}. \quad (15)$$

Substituting Eq. (15) to Eq. (12), we have

$$\mu_{ij} = \left(\sum_{p=1}^S \left(\frac{\sum_{k=1}^Q \|x_{kj} - c_i\|^2}{\sum_{k=1}^Q \|x_{kj} - c_p\|^2} \right)^{\frac{1}{m-1}} \right)^{-1}. \quad (16)$$

To obtain the centers of the clusters, we can calculate the derivative of F with respect to c_i :

$$\frac{\partial F(\lambda, U, C)}{c_i} = -2 \sum_{j=1}^P \mu_{ij}^m \sum_{k=1}^Q (x_{kj} - c_i) = 0. \quad (17)$$

Then we have,

$$c_i = \frac{\sum_{j=1}^P \mu_{ij}^m \sum_{k=1}^Q x_{kj}}{Q \cdot \sum_{j=1}^P \mu_{ij}^m}. \quad (18)$$

In essence, the modification of FCM is to consider a class of data points as a whole and calculating the membership degree for each class instead of each sample. After the modification, the samples in the same class share a membership degree μ_{ij} , and the updating rule of cluster centers and membership degree is transformed from (8) to (19) as follows.

$$c_i = \frac{\sum_{j=1}^P \mu_{ij}^m \sum_{k=1}^Q x_{kj}}{Q \cdot \sum_{j=1}^P \mu_{ij}^m}, \quad (19a)$$

$$\mu_{ij} = \left(\sum_{p=1}^S \left(\frac{\sum_{k=1}^Q \|x_{kj} - c_i\|^2}{\sum_{k=1}^Q \|x_{kj} - c_p\|^2} \right)^{\frac{1}{m-1}} \right)^{-1}. \quad (19b)$$

The algorithm of clustering the original dataset is shown in Algorithm 1. Note that the data fed into the subtractive clustering model and FCM is the data after feature extraction. In Algorithm 1, we first use the subtractive clustering method to get the intuition of appropriate cluster numbers and initialize the cluster centers. The objective of the optimization problem (9) is reached by updating the values of μ_{ij} and c_i .

Algorithm 1 Integration of Subtractive Clustering and Modified FCM

- 1: Using subtractive clustering method (introduced in Section II-A) to obtain the number of clusters N and initial centers C . Assume that the maximum number of iteration is T and the minimum number of improvement is δ .
 - 2: **repeat**
 - 3: Calculate μ_{ij} according to Eq. (19b).
 - 4: Update the cluster centers c_i by Eq. (19a).
 - 5: Calculate the objective function Eq. (9a).
 - 6: **until** the objective value improves by less than the threshold δ or after T iterations.
-

The STOP conditions are reaching the threshold $\delta = 0.0001$ or the maximum number of iterations $T = 50$. If any one of the conditions is satisfied, the algorithm stops.

VI. SUPERVISED LEARNING APPROACHES

The supervised learning approaches are employed to implement the classification based on the cluster results obtained in Section V. As described in Section III and Section V, three clusters are obtained after using subtractive clustering and modified FCM algorithm:

- Cluster 1: class 1, class 5, class 7, class 9
- Cluster 2: class 2, class 3, class 8
- Cluster 3: class 4, class 6, class 10

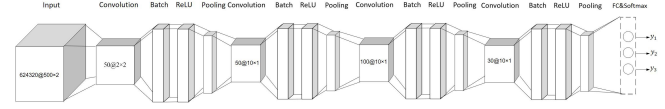


Fig. 6. The structure of the CNN used in this paper.

Thus, we establish a top model to classify the inputs into three classes first, then use three classifiers respectively to classify the data in three clusters. To train the top model, the target of inputs in classes 1, 5, 7, 9 is set to 1, the target of inputs in classes 2, 3 and 8 is set to 2 and that of the rest is set to 3.

A. Top Classifier

Five different models for the top classifier are attempted: CNN, FNN, decision tree, KNN and random forest. The specifications of the classifiers are shown in Table II. The architecture of the top CNN is shown in Fig. 6.

For the whole classification structure, the top classifier's accuracy is significant for the final accuracy. If a data point is wrongly classified in the beginning by the top classifier, it is impossible that the sample is predicted correctly in the end, which is also the main fragility of this two-step classification structure.

In order to find out the most appropriate data processing method, feature extraction, windowing method and the combination of them are adopted in this paper. That is, three data types: (i) data after feature extraction, (ii) data after windowing and (iii) data after the combination of windowing and feature extraction, are respectively fed into the classifiers FNN, CNN, KNN, decision tree and random forest. Each classifier is fed by three types of inputs (i)-(iii) and the data type which achieves the best performance is chosen for this classifier. Table III shows the selected data processing method for each classifier and the corresponding classification accuracy.

The training and test performance of various classifiers with different input types is shown in Table III. It can be seen from the results that FNN, KNN and random forest are fed by the data after feature extraction, CNN's input is the windowing data. Decision tree uses the data processed by feature extraction and windowing, which means doing windowing first and then extract the features from the newly generated windowing samples. Two features used here are illustrated in Section IV-B.2.

The results in Table III shown that CNN with windowing method achieves the best performance, i.e., 100% test accuracy when classifying the data into three clusters. Decision tree and random forest also have high accuracy, 97.50% and 96.25% respectively, FNN and KNN follows with the test accuracy 95.63% and 95.00%. From the results shown in Table III, it can also be concluded that the dimension reduction method is more preferable for the conventional machine learning methods while the deep learning method relies on large amount of training data to learn and abstract out the patterns inherent in the data.

Based on the classification performance of different models, CNN with the windowing method is selected to be the top classifier.

TABLE II
CONFIGURATIONS OF MACHINE LEARNING MODELS

FNN	CNN	KNN	Random Forest	Decision Tree
Structure: basic FNN structure is shown in Fig. 2. Three hidden layers used in this paper are of the size (25, 25, 25); Optimizer: LM*; Epoch: 500; Loss function: MSE**.	Structure: the layers, filter size and the number of features are shown in Fig. 6; Optimizer: Adam; Batchsize: 128; Epoch: 60; Loss function: MSE; Data shuffling: once before training.	k-nearest neighbours: 3; Leaf size: 30; Weight function used in prediction: uniform weights; The distance metric to use for the tree: minkowski; Power parameter for the Minkowski metric: 2.	The number of trees in forest: 500; Criterion: Gini Impurity; The minimum number of samples required to split an internal node and to be at a leaf node: 2, 1, respectively.	The measurement of the quality of a split: Gini impurity; The smallest number of samples at a leaf node: 1; The minimum number of samples required to split an internal node: 2.

* LM: Levenberg-Marquardt method.

** MSE: mean squared error.

TABLE III
TRAINING AND TEST PERFORMANCE OF THE TOP CLASSIFIERS

Classifier	Input type	Training Accuracy	Test Accuracy
FNN	Feature extraction	100.00%	95.63%
KNN	Feature extraction	98.75%	95.00%
Decision Tree	Windowing & Feature extraction	100.00%	97.50%
Random Forest	Feature extraction	99.69%	96.25%
CNN	Windowing	100.00%	100.00%

B. Sub-Classifiers

As CNN with windowing method achieves satisfied performance at the first classification stage, to train three sub-classifiers, we also adopt windowing method to generate windowing samples. Since the windowing inputs are used by sub-classifiers, majority voting is conducted again when collecting the results. More specifically, majority voting is used twice in the process of two-step classification. The first voting is conducted to rectify the predicted results of the top classifier. It is implemented again after the sub-classifiers to further improve the classification performance.

The three sub-classifiers are modeled by CNNs which have the same specification as the top classifier's. Because there are four classes in Cluster 1, the structure of the first sub-classifier has four nodes in the output layer, the second and the third sub-classifier have the same structure as the top classifier shown in Fig. 6.

After training, the training and test accuracy of three sub-classifiers are all 100%. Till now, the proposed two-step approach achieves 100% test accuracy (the test accuracy is 94.98% before the majority voting), meaning every original sample in the test dataset can be correctly classified.

VII. ROBUSTNESS

Robustness analysis aims to evaluate the model's tolerance of input perturbations. We add different levels of Gaussian noise to the test data and feed the contaminated data into the classifiers. Assume that the inputs are represented by a matrix a whose dimensions are $s_1 \times s_2$. After adding additive form of noise, the inputs becomes

$$\tilde{a} = a + m \cdot \text{randn}(s_1, s_2),$$

where \tilde{a} denotes the contaminated inputs, m represents the noise levels which are positive values, $\text{randn}()$ represents the function of Gaussian noise and \cdot denotes the element-wise

TABLE IV
ROBUSTNESS RESULT: TEST ACCURACY OF CLASSIFIERS* WHEN GIVING DIFFERENT LEVELS OF NOISE

Model	Proposed method	CNN	FNN**	KNN	RF	DT
Noise Level						
1e-5	100.00%	94.06%	75.62%	71.88%	75.62%	81.88%
2e-5	95.63%	92.50%	60.63%	41.25%	73.13%	78.13%
1e-4	42.50%	58.96%	24.37%	32.50%	53.75%	56.25%

* RF and DT respectively denote random forest and decision tree.

** Optimization algorithm in training NN: cgb.

multiplication, the size of $\text{randn}(s_1, s_2)$ is the same as the size of \tilde{a} . Given that the raw inputs' order of magnitude is 10^{-5} , the noise levels are accordingly set to $m = 1 \times 10^{-5}$, 2×10^{-5} , 1×10^{-4} .

To make comprehensive comparison with other conventional machine learning approaches, we also employ decision tree, random forest, KNN, FNN and CNN for one-step training. Different from the proposed method, the comparative models are fed by regular data, i.e., 10 classes inputs and targets as well as the corresponding contaminations for robustness analysis. The results are shown in Table IV. The proposed model achieves the 100% test accuracy when giving the level of 1×10^{-5} additive noise, and has 95.63% test accuracy when the noise level is 2×10^{-5} .

From Table IV, it can be concluded that the proposed model provides the best performance among the classifiers, CNN, FNN, decision tree, random forest and KNN follow successively, which further demonstrates that the proposed method has better generalization ability to accept wider range of data and its suitability to real-life application when there are inevitable data perturbations caused by electrode shifts, muscle fatigue, etc..

VIII. CONCLUSION

In this paper, the unsupervised learning approaches and supervised learning methods are employed to classify EMG signals consisting of 10 classes of hand gestures. The clustering algorithms including subtractive clustering and the modified FCM are first employed to group the classes which have similarity and are difficult to be classified. Based on the obtained clusters, the classification procedure is divided into two steps. At the first stage, a top classifier is used to classify the data into three clusters; at the second stage, sub-classifiers are trained accordingly to classify the data

in each cluster. The proposed method achieves 100% test accuracy and the strongest robustness compared to five widely used classifiers. The main contributions of this paper include the modification of conventional FCM algorithm to make it accustomed to multi-class classification, the design of two-step classification which reaches 100% classification accuracy and strong robustness, and the comprehensive comparison with five widely used machine learning models.

REFERENCES

- [1] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: Detection, processing, classification and applications," *Biol. Procedures Online*, vol. 8, no. 1, pp. 11–35, Dec. 2006.
- [2] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Comput. Methods Programs Biomed.*, vol. 161, pp. 1–13, Jul. 2018.
- [3] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, and A. T. Chronopoulos, "Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions," *Neurocomputing*, vol. 276, pp. 2–22, Feb. 2018.
- [4] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, Apr. 2018.
- [5] A. Dwivedi, Y. Kwon, A. J. McDaid, and M. Liarokapis, "A learning scheme for EMG based decoding of dexterous, in-hand manipulation motions," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 10, pp. 2205–2215, Oct. 2019.
- [6] K.-T. Kim, C. Guan, and S.-W. Lee, "A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 94–103, Jan. 2020.
- [7] U. Cote-Allard et al., "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.
- [8] U. Cote Allard et al., "A convolutional neural network for robotic arm guidance using sEMG based frequency-features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2464–2470.
- [9] K.-H. Park and S.-W. Lee, "Movement intention decoding based on deep learning for multiuser myoelectric interfaces," in *Proc. 4th Int. Winter Conf. Brain-Comput. Interface (BCI)*, Feb. 2016, pp. 1–2.
- [10] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers Neurosci.*, vol. 11, p. 379, Jul. 2017.
- [11] R. N. Khushaba, A. Al-Timemy, S. Kodagoda, and K. Nazarpour, "Combined influence of forearm orientation and muscular contraction on EMG pattern recognition," *Expert Syst. Appl.*, vol. 61, pp. 154–161, Nov. 2016.
- [12] A. H. Al-Timemy, R. N. Khushaba, G. Bugmann, and J. Escudero, "Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 6, pp. 650–661, Jun. 2016.
- [13] R. N. Khushaba, S. Kodagoda, D. Liu, and G. Dissanayake, "Muscle computer interfaces for driver distraction reduction," *Comput. Methods Programs Biomed.*, vol. 110, no. 2, pp. 137–149, May 2013.
- [14] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control," in *Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2012, pp. 1534–1539.
- [15] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10731–10738, Sep. 2012.
- [16] K. Anam, R. N. Khushaba, and A. Al-Jumaily, "Two-channel surface electromyography for individual and combined finger movements," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 4961–4964.
- [17] A. Waris, I. K. Niazi, M. Jamil, K. Englehart, W. Jensen, and E. N. Kamavuako, "Multiday evaluation of techniques for EMG-based classification of hand motions," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 4, pp. 1526–1534, Jul. 2019.
- [18] S. Benatti, F. Montagna, V. Kartsch, A. Rahimi, D. Rossi, and L. Benini, "Online learning and classification of EMG-based gestures on a parallel ultra-low power platform using hyperdimensional computing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 3, pp. 516–528, Jun. 2019.
- [19] R. R. Yager and D. P. Filev, "Generation of fuzzy rules by mountain clustering," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 209–219, 1994.
- [20] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, 1994.
- [21] G. Jia, Z. Yang, H.-K. Lam, J. Shi, and M. Shikh-Bahaei, "Channel assignment in uplink wireless communication using machine learning approach," 2020, *arXiv:2001.03952*. [Online]. Available: <http://arxiv.org/abs/2001.03952>
- [22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [23] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.
- [24] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, p. 175, Aug. 1992.
- [25] S. S. Mullick, S. Datta, and S. Das, "Adaptive learning-based k -nearest neighbor classifiers with resilience to class imbalance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5713–5725, Nov. 2018.
- [26] H. Yu and B. M. Wilamowski, "Levenberg–Marquardt training," in *Industrial Electronics Handbook* (Intelligent Systems), vol. 5. Boca Raton, FL, USA: CRC Press, 2011, pp. 12–1–12–16.
- [27] F. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Towards the control of individual fingers of a prosthetic hand using surface EMG signals," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 6145–6148.
- [28] F. V. G. Tenore, A. Ramos, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 5, pp. 1427–1434, May 2009.
- [29] R. J. Smith, F. Tenore, D. Huberdeau, R. Etienne-Cummings, and N. V. Thakor, "Continuous decoding of finger position from surface EMG signals for the control of powered prostheses," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2008, pp. 197–200.
- [30] A. Andrews, E. Morin, and L. McLean, "Optimal electrode configurations for finger movement classification using EMG," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 2987–2990.
- [31] G. Tsenov, A. H. Zeghib, F. Palis, N. Shoylev, and V. Mladenov, "Neural networks for online classification of hand and finger movements using surface EMG signals," in *Proc. 8th Seminar Neural Netw. Appl. Electr. Eng.*, 2006, pp. 167–171.
- [32] D. Peleg, E. Braiman, E. Yom-Tov, and G. F. Inbar, "Classification of finger activation for use in a robotic prosthesis arm," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 4, pp. 290–293, Dec. 2002.
- [33] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7420–7431, Jun. 2012.
- [34] M. Asghari Oskoei and H. Hu, "Myoelectric control systems—A survey," *Biomed. Signal Process. Control*, vol. 2, no. 4, pp. 275–294, Oct. 2007.
- [35] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Crit. Rev. Biomed. Eng.*, vol. 45, nos. 1–6, pp. 383–410, 2017.
- [36] S. Moein, *Medical Diagnosis Using Artificial Neural Networks*. Hershey, PA, USA: IGI Global, 2014.
- [37] R. Tibor Schirmer et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," 2017, *arXiv:1703.05051*. [Online]. Available: <http://arxiv.org/abs/1703.05051>