IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 March 2023, accepted 9 April 2023, date of publication 17 April 2023, date of current version 24 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267674

RESEARCH ARTICLE

Artificial Intelligence for sEMG-Based Muscular Movement Recognition for Hand Prosthesis

NAFE MUHTASIM HYE[1], UMMA HANY[1], SUMIT CHAKRAVARTY[2], (Member, IEEE), LUTFA AKTER[3], AND IMTIAZ AHMED[4], (Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Ahsanullah University of Science and Technology, Dhaka 1208, Bangladesh
[2]Department of Electrical and Computer Engineering, Kennesaw State University, Marietta, GA 30060, USA
[3]Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh
[4]Department of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059, USA

Corresponding author: Umma Hany (uhany.eee@aust.edu)

This work was supported by the Start-Up Research Grant of the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA.

ABSTRACT The muscular activities gathered by real-time myoelectric interfaces of surface electromyography (sEMG) can be used to develop myoelectric prosthetic hands for physically disabled people. However, the acquired myoelectric signals must be accurately classified in real time to properly control the operation of the external devices. In this study, we propose methods for detecting and classifying muscular activities using sEMG signals. These methods include outlier removal, data manipulation, data preprocessing, dimensionality reduction, and classification. We use the Ninapro database 1 (DB1) containing sEMG signals from 27 intact subjects while performing 53 hand movements repeatedly. We apply the Principle Component Analysis (PCA), Independent Component Analysis (ICA), and t-distributed Stochastic Neighbor Embedding (t-SNE) feature extraction methods for dimensionality reduction. Five machine learning (ML) algorithms and deep learning artificial neural networks (ANN) are applied for the classification of muscular movements. It is observed that for the recognition of 53 muscular movements of 27 subjects with preprocessed raw data, ANN obtains the highest accuracy of 93.92% for inter-subject and 97.73% for intra-subject movement recognition. Among the ML algorithms, K-Nearest Neighbors (KNN) performs the best with both t-SNE features and the preprocessed raw data in least computational time. With the preprocessed raw data, KNN obtains 93.174% and 97.458% for inter-subject and intra-subject movement classification, respectively while with the t-SNE features, KNN obtains 89.844% accuracy for inter-subject and 95.04% accuracy for intra-subject in reduced computational time.

INDEX TERMS Gesture recognition, computational and artificial intelligence, biomedical signal processing.

I. INTRODUCTION

The myoelectric interfaces of the surface electromyography gather muscular activity information for different movements. The sEMG signals of different muscular activities can be used to control the operation of external prosthetic devices that are used to make the life of physically disable people easy and comfortable. The external devices may be wearable or not wearable depending on the requirements of the disable people. To ensure the quality and proper operation of the external devices, the muscular activities for different movements are required to be classified accurately in real-time to control the operation of the external devices.

There are many works in the literature on muscular movement recognition based on the EMG signals [1]. In [2], the authors consider feature extraction using the mean absolute value (MAV), the variance (VAR), the waveform length (WL), sEMG histogram (HIST), cepstral coefficients (CC),

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Kafiul Islam.

short-time Fourier transform, marginal discrete wavelet transform (mDWT) and classification using linear discriminant analysis (LDA), k-nearest neighbors (KNN), support vector machine (SVM), multi-layer perceptron (MLP), SVM with a radial basis function (RBF) kernel on the Ninapro DB1 and achieve a classification accuracy of 76% by several of the above-mentioned methods. In [3] and [4], the authors use Ninapro DB1 of 27 subjects and classify the movements using RMS features and a convolutional neural network (CNN) classifier. The authors in [3] report inter-session gesture recognition accuracy of 94% for 6 movements while the authors in [4] report 70.5% intra-session gesture recognition accuracy for 53 movements. The authors in [5] propose a system that depends on the extraction of multi-channel EMG activation trajectories underlying hand movements, and classifying the extracted trajectories using a metric based on multi-dimensional dynamic time warping (MD-DTW). The authors in [5] apply their proposed system on Ninapro database 2 (DB2) for 40 different hand movements of 40 subjects and obtain a classification accuracy of 90%. However, the main limitation of the proposed work is that a classification of the movement cannot be obtained until the end of the trajectory is detected. This restricts the usability of the proposed system to applications that can be controlled using discrete commands. The authors in [6] use the raw sEMG of Ninapro DB1 and obtain 75.45% intra-session accuracy in recognition of 52 movements of 27 subjects using Long short-term memory (LSTM) classifier. The authors in [7] apply transfer learning (TL) and deep learning algorithms to learn features from the big data collected from multiple users. Three different deep learning networks with raw EMG, spectrograms and continuous wavelet transform (CWT) as input are trained and tested on three datasets - two datasets from Evaluation database comprised of 19 and 17 intact subjects and one dataset of 10 participants from Ninapro database 5 (DB5). The maximum achieved offline accuracy on test data from Evaluation database is 98.31% for 7 gestures of 17 participants using CWT-based ConvNet and 68.98% using Ninapro database of 18 gestures of 10 participants using raw EMG-based ConvNet. The authors in [8] use high density EMG signal to decode the motion intention based on the transient EMG signals and achieve decoding performance of 94.21% with CapgMyo and below 70% using Ninapro database 4 (DB4) for 8 finger movements using SVM. In [9], the authors use RMS features and apply temporal convolutional network (TCN) classifier for gesture recognition and report 89.76% intra-session accuracy in classifying 52 movements of 27 subjects using Ninapro DB1. Transfer learning based several approaches are proposed in [10], [11], [12], and [13] and applied on Ninapro database 2, 3, 5 and 6, respectively. The reported accuracy in [10] is 49.76% for self-decoding and 52.52% for subject transfer using CNN. The reported average accuracy in [11] is 67.98% with Gaussian kernel function SVM classifier (GKF-SVM) using Mean value of square root (MSR) features and 70.40% using transfer learning. The authors in [12] propose Transfer learning based multi-scale kernel convolutional neural network (TL-MKCNN) and achieve 97.22% within-session, 74.48% cross-subject, and 90.30% cross-day accuracy that are higher than the MKCNN model by 4.31%, 11.58%, and 5.51%, respectively. The authors in [13] 86.3% accuracy using their proposed Few-shot Learning for Hand Gesture (FS-HGR) framework. The authors in [14] use tensor-based multilinear singular value decomposition (MLSVD) for hand gesture recognition with multiple channels for training and single channel for recognition. The authors in [14] apply the proposed methods on Ninapro, CapgMyo (DB-a, DB-b and DB-c), and CSL-HDEG databases for intra-session, inter-session and inter-subject evaluations and obtained highest accuracy of 75.2%, 75.4%, 68.3% and 67.7%, respectively for inter-subject classification. The authors in [15] develop a real-time system for prosthetic hands control and use a database with EMG data from local volunteers and Ninapro 2 and 3 databases. They implement a MLP classifier on a platform for rapid prototyping (Raspberry Pi 3 Model B+) and generate responses in real-time (11ms) with an average accuracy of 96.30% for 11 hand and wrist gestures/movements.

Though there have been extensive research works on sEMG-based movement recognition, still there is scope for improvement considering big database of a higher number of users and movements. For real-time hand prosthesis, the muscular movements must be accurately detected and classified in real-time with low computational complexity and time.

In this paper, we propose the methods of sEMG signals based movement detection and classification for hand prosthesis using artificial intelligence techniques. To control the operation of the external prosthetic devices properly, the muscular activities due to different movements are required to be detected and classified accurately. Therefore, the sEMG signals for different muscular activities are collected, preprocessed, and classified using different machine learning and deep learning methods. We collect the sEMG data from the publicly available Ninapro DB1 [16]. Ninapro DB1 includes 10 repetitions of 53 movements (including rest position) of 27 intact subjects. We apply dimensionality reduction techniques on the preprocessed dataset to reduce the computational complexity and processing time of the machine learning methods. We apply different Linear feature extraction techniques as PCA, ICA and Non-linear feature extraction technique as t-SNE to reduce the dimensionality of the dataset. Among those, t-SNE performs the best in terms of accuracy. t-SNE is a non-linear dimensionality reduction technique suitable for visualizing high-dimensional data in a low-dimensional space of two or three dimensions. After applying the feature extraction, we apply five different machine learning methods and compare the movement classification accuracy. As we can observe that for 53 movements, the KNN classifier shows the best accuracy

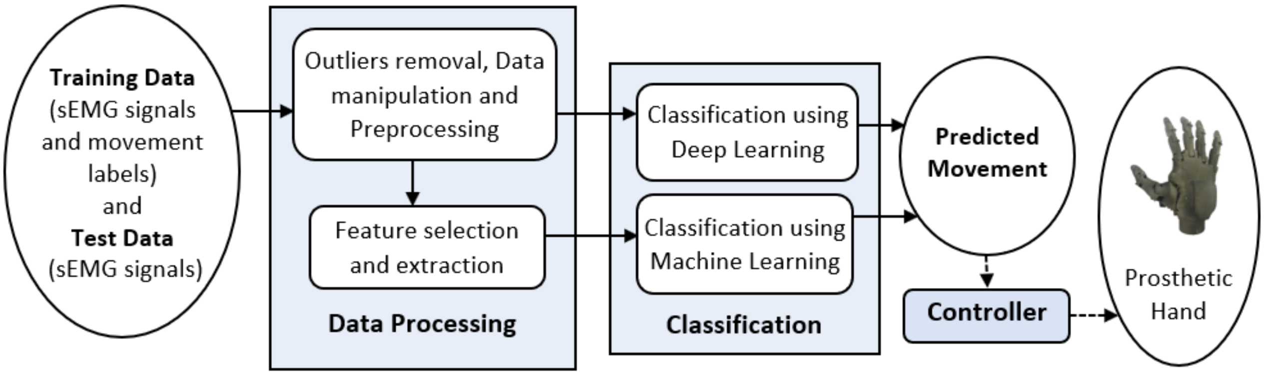

FIGURE 1. System block diagram.

of 93.174% for inter-subject and 97.458% for intra-subject classification with the preprocessed raw data in the least processing time among the ML methods. By extracting the 3-dimensional t-SNE features, the KNN obtains 89.844% inter-subject accuracy and 95.04% intra-subject accuracy with reduced processing time and computational complexity. We also apply deep learning ANN for classification which shows the best accuracy of 93.92% for inter-subject and 97.73% for intra-subject movement recognition.

II. SYSTEM MODEL

We propose sEMG based movement detection and classification using artificial intelligence to control the operation of the prosthetic hand. The system block diagram of the proposed system is shown in Figure 1. Here, the system uses the sEMG signals collected for different muscular activities of the hand using real-time myoelectric interfaces or electrodes. The sEMG signals of both training and test phases are preprocessed and classified to detect the intended hand movements using different machine learning and deep learning methods. The feature extraction method is applied before machine learning-based classification to reduce the dimensionality of the sEMG dataset. It also reduces the computational complexity and processing time of the classifier. Deep learning-based classifiers are applied to classify the movements using preprocessed raw data. The predicted movements may control the operation of the prosthetic hand.

III. DATA SOURCE AND PARTICIPANTS

We use the Ninapro DB1 available in the official Ninapro repository [16]. The database contains muscular activity data acquired using OttoBock sEMG electrodes. The datasets were collected in [17] using 10 OttoBock MyoBock 13E200-50 electrodes (www.ottobock.com), providing an amplified, bandpass-filtered, and Root-Mean-Square (RMS) rectified version of the raw sEMG signals. Among the 10 electrodes, eight are equally spaced around the radio humeral joint of the forearm at position 1, and two are placed on the finger extensor digitorum and finger flexor digitorum at position 2

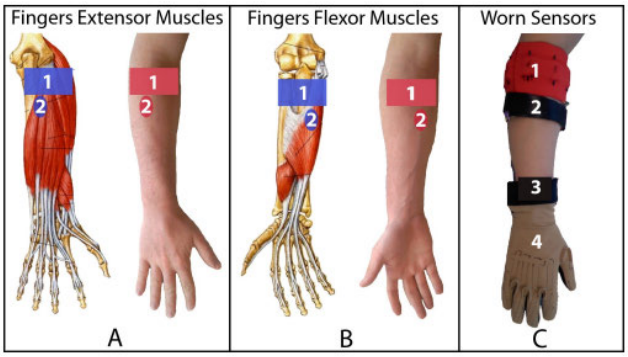

FIGURE 2. Placement of the sEMG electrodes; Position 1: Eight equally spaced electrodes around the radio humeral joint of the forearm; Position 2: Two additional electrodes placed on finger extensor and flexor muscles; Position 3 and 4: Cyberglove sensors. [17].

as shown in Fig. 2. DB1 includes labeled data of 10 repetitions of 52 hand movements and 1 rest position of 27 intact subjects. During the experiment, the subjects were asked to repeat the 52 movements of Exercise A, B, and C as shown in Fig. 3 with the right hand for 5 seconds followed by 3 seconds of rest. The three exercises include the following movements.

1) Exercise A: 12 basic movements of the fingers (flexions and extensions),
2) Exercise B: 8 isometric and isotonic hand configurations and 9 basic movements of the wrist, total 17 movements, and
3) Exercise C: It is done with 23 grasping and functional movements.

Several signal processing steps including synchronization, relabelling, and filtering was performed before making the data available on the repositories [16], [17]. The sEMG signals are sampled to a high sampling frequency of 2 kHz to synchronize the data streams and then relabeled [17]. The Ninapro DB1 contains one matlab file with synchronized variables of each exercise for each subject. A single file includes the variables as shown in Table 1.

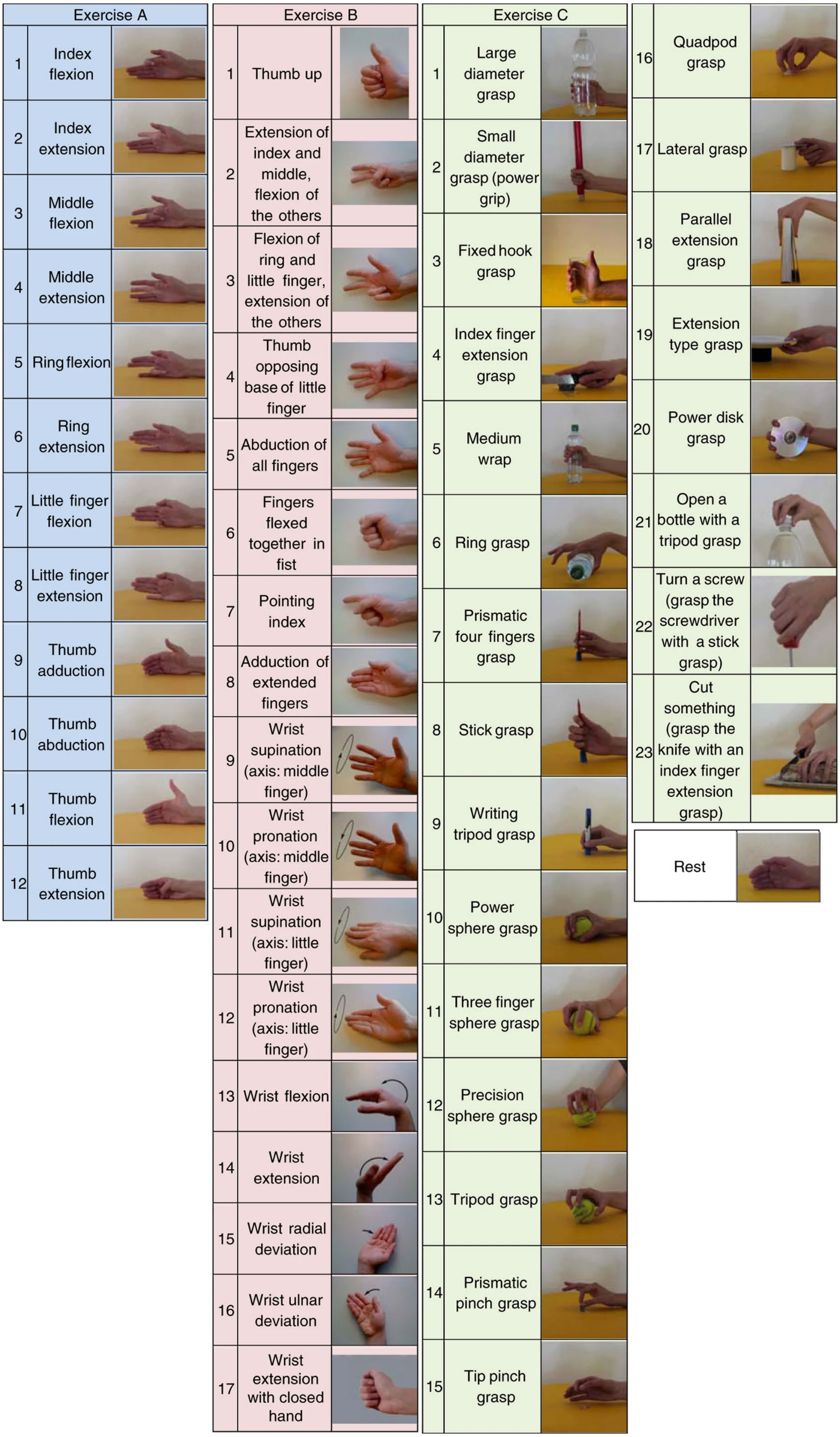

FIGURE 3. Rest position and 52 hand movements of Exercise A: 12 Basic movements of the fingers (flexions and extensions), Exercise B: 17 Isometric, isotonic hand configurations and Basic movements of the wrist and Exercise C: 23 Grasping and functional movements. [17].

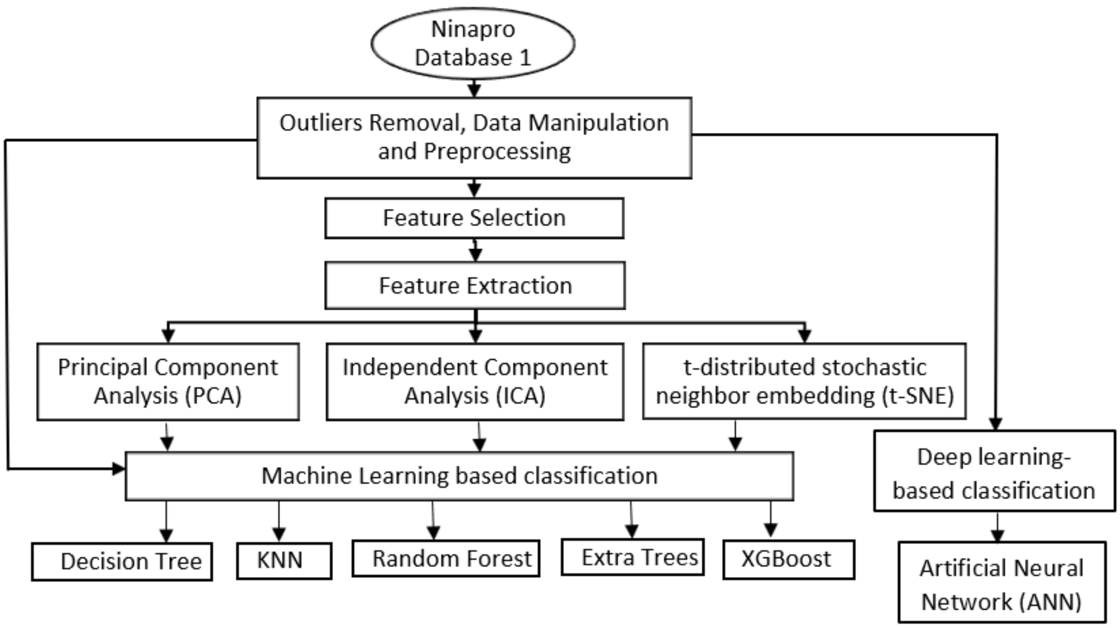

FIGURE 4. Implementation Methodologies.

TABLE 1. Variables in a single matlab file of Ninapro DB1 of each exercises for each subject.

Variable	Contents
subject	Subject number
exercise	Exercise number
emg	Columns 1 to 8 includes the sEMG signals of the electrodes around the forearm; columns 9 and 10 includes sEMG signals of the electrodes placed on Flexor Digitorum Superficialis and Extensor Digitorum Superficialis, respectively.
glove	The 22 columns of this variable includes uncalibrated signals of 22 Cyberglove sensors.
stimulus	The single column includes the original label of the movements repeated by the subjects
restimulus	The single column includes the a-posteriori refined label of the movements
repetition	The single column includes the stimulus repetition index
rerepetition	The single column includes the restimulus repetition index

IV. METHODOLOGIES

We propose the methodologies to develop the sEMG-based muscular movement recognition system for hand prostheses. The methodologies include outliers detection and removal, data manipulation, preprocessing, feature selection, feature extraction, and classification. The proposed methodologies are developed in Python programming language using a variety of popular data analysis libraries and tools, including pandas, scipy.io, loadmat, numpy, matplotlib, seaborn, and the Keras, TensorFlow and Scikit-Learn tools. Since Ninapro dataset belongs to the big data category, it is not practical to apply the entire dataset for classification due to very high processing time. Therefore, we apply the outliers removal, data manipulation, preprocessing, and dimensionality reduction methods to reduce the size of the data. The implementation methodologies with the system flow are shown in Figure 4 and explained in the following.

A. OUTLIERS DETECTION AND REMOVAL

The required variables for our proposed movement recognition system are the *emg* and the *stimulus*. The *emg* variable includes the sEMG signals of 10 electrodes and the *stimulus* variable includes the labels of 53 movements (including rest). To keep the required variables and to remove the outliers, we apply the following methods:

1) First, the variables except *emg*, *stimulus* and *restimulus* are removed.
2) Then, to extract the required data, we apply outliers detection and removal method. To detect the outliers, we apply a matching algorithm to find the matches between the *stimulus* and *restimulus*. If any value of the *stimulus* matches with the *restimulus* values, then the corresponding sEMG signals (of 10 electrodes) and the *stimulus* values are kept unchanged. Else if the stimulus doesn't match the restimulus, then the corresponding sEMG signals and the stimulus values are considered as outliers and dropped. In this way, the unnecessary variables and outliers are removed and the required values of the *emg* and *stimulus* are extracted for further processing.

B. DATA MANIPULATION

We apply the following three steps for data manipulation before the preprocessing of data.

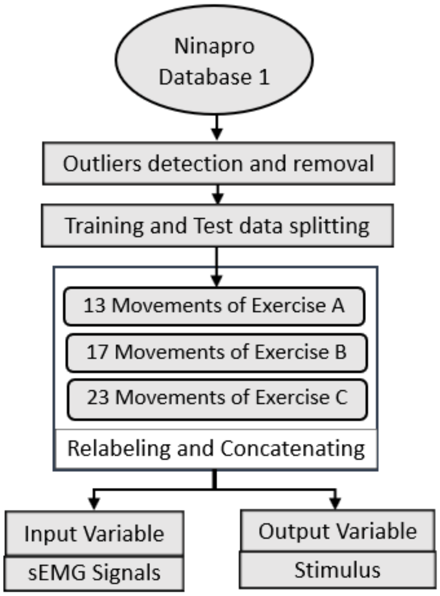

FIGURE 5. Outliers removal and data manipulation.

1) DATA SPLITTING

First, we split data into training and test phases where 90% of the data are used for training and 10% data are used for testing. All the variables of the dataset are not required for the sEMG based movement recognition.

2) RELABELING

There are 13 movements of Exercise A, 17 movements of Exercise B and 23 movements of Exercise C. The 13 distinct movements of Exercise A are labeled as 0 to 12, 17 movements of Exercise B as 0 to 17 and 23 movements of Exercise C as 0 to 23, respectively in the corresponding *stimulus* variables. However, 53 movements of Exercises A, B, and C should have 53 distinct labels. Therefore, the movement labels of Exercise A (1 to 12), Exercise B (1 to 17), Exercise C (1 to 23) are relabeled as 1 to 12, 13 to 29 and 30 to 52, respectively to get distinct movement labels for the 53 movements.

3) CONCATENATION

Finally, the sEMG data of all 53 movements (including rest position) of 3 exercises and the corresponding stimulus labels are combined in a single data file by concatenation process. Thus, the concatenated training and test data files include all the sEMG signals of 53 distinct movements.

The *emg* variable including 10 sEMG signals of each movement is used as the input in both training and test phase. The *stimulus* including the movement labels is used as the output in the training phase. In the test phase, the *stimulus* is used to evaluate the accuracy of the predicted movements.

The system flow diagram of the outliers removal and data manipulation is shown in Fig. 5.

C. DATA PREPROCESSING

The following preprocessing methods are applied to the training and test data.

1) STANDARD SCALING

After data manipulation, we apply standard scaling on the sEMG signals of input variable to remove the mean and to scale the data to unit variance. The standard score is calculated as

$$z = \frac{(x-\mu)}{s}, \tag{1}$$

where, μ is the mean and s is the standard deviation of the samples. Standard scaling is applied to the input variable of both machine learning and deep learning models.

2) ONE HOT ENCODING

One hot encoding is applied to encode the movement labels of the output variable *stimulus* prior applying to the deep learning model for classification.

D. DIMENSIONALITY REDUCTION

To reduce the dimensionality of the dataset, feature selection and feature extraction methods are applied. Feature selection discards less important features of the data whereas feature extraction creates new features which summarize the contents of the original features. Feature extraction improves the accuracy, reduces overfitting risk, speeds up in training and improves data visualization.

1) FEATURE SELECTION

To reduce the dimensionality of the dataset, we apply the Recursive Feature Elimination with Cross Validation (RFECV) which is a wrapper-based method of feature selection. RFECV ranks the features among the 10 sEMG signals as per importance and then selects the optimal features by cross validation [18].

2) FEATURE EXTRACTION

We apply and compare the following feature extraction techniques to reduce the dimensionality of the dataset.

- Principle Component Analysis (PCA): PCA is a common linear feature extraction method used to reduce the dimensionality of data by finding the best combination of input features and capturing maximum information about the dataset. However, if the components are not selected with care, it may miss some information [19], [20].
- Independent Component Analysis (ICA): ICA is a linear method which takes the mixed independent components and correctly identify them removing the unnecessary noises. However, ICA algorithms are sensitive to measurement noise [21].
- t-distributed Stochastic Neighbor Embedding (t-SNE): t-SNE is a very effective non-linear dimensionality

reduction algorithm. t-SNE tries to minimize the divergence between the different probability distribution of the input features in the original high dimensional space and in the reduced low dimensional space. The dissimilarity of two different distributions is measured using Kullback-Leiber (KL) divergence and it is minimized using gradient descent. The original dimensional space is modelled using a Gaussian Distribution, while the lower-dimensional space is modelled using a student's t-distribution. t-SNE handles non-linear data efficiently and it preserves local and global structure of the data. However, t-SNE is slow and computationally complex as it has a quadratic time and space complexity in the number of data points. It also requires a lot of calculations for computing the pairwise conditional probabilities for each data point and to minimize the sum of the difference of the probabilities in higher and lower dimensions. It involves hyper-parameter tuning which may cause unwanted results due to incorrect tuning [19], [22].

E. CLASSIFICATION AND MOVEMENT RECOGNITION

In the training phase, the sEMG signals are trained to the corresponding stimulus labels of movements. In the test phase, the sEMG signals are applied to the classifier to detect the corresponding movements. In the machine learning model, we compare the accuracy by applying both preprocessed data and the features extracted. As the deep learning model performs better with raw data, only the preprocessed data are applied for classification.

1) MACHINE LEARNING METHODS

For machine learning based classification, we employ the Decision Tree (DT), KNN, Random Forest (RF), Extra Trees (ET), and Extreme Gradient Boosting (XGB) classifiers [23], [24], [25], [26], [27]. To fine-tune the parameters, we apply GridSearchCV with three levels of cross validation. The default settings of the DT classifier, RF classifier, and ET classifier, perform the best. For the XGB, we use the "gpu predictor" to speed up the training process. For the KNN classifier, we use the "ball tree" algorithm as the optimum parameter. For KNN, we considered 2 nearest neighbors (K = 2), manhattan distance as a measure of distance and the wights measured by distance. However, there is space for improvement by boosting the estimators for all tree-based methods which could not be implemented due to a lack of computing configurations.

2) PROPOSED DEEP LEARNING ARCHITECTURE

We apply Artificial Neural Networks for deep learning-based classification. The architecture of the ANN is shown in Fig. 6. We start with 10 emg signals as inputs, then add a dense layer of 3000 neurons with a relu activation function, followed by another dense layer of 1500 neurons with a relu activation function. Next, a 0.2 dropout layer is added. 3 dense layers

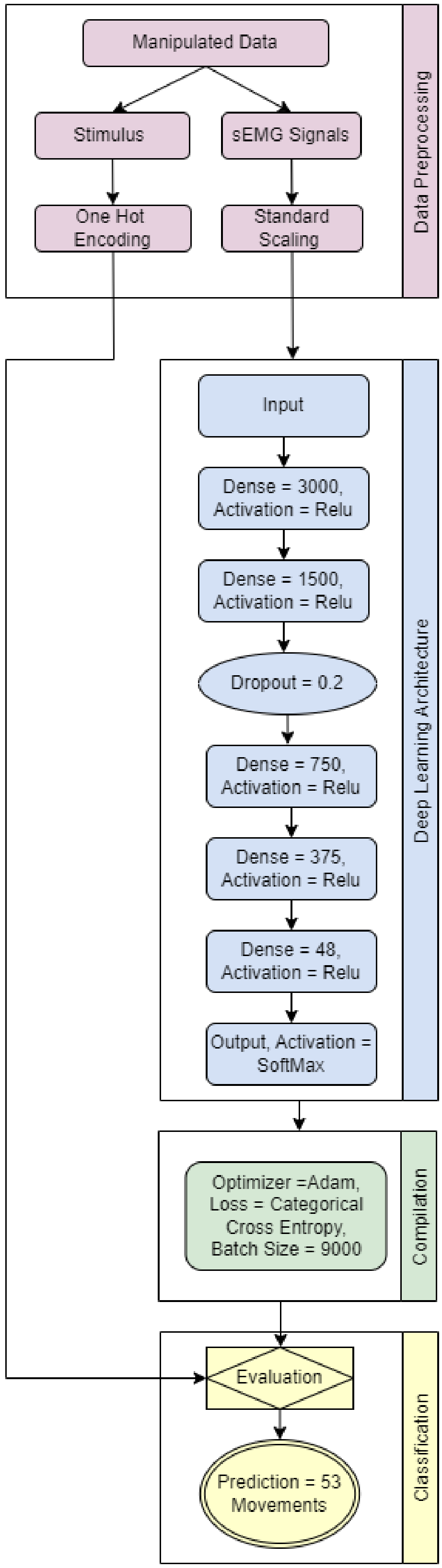

FIGURE 6. Deep learning architecture.

with activation functions of 750, 375, and 48 correspondingly are then added. A dense layer as 53 class with a SoftMax activation function is added last. Then, with batch size set to 9000 and 300 epochs, we compile the model using Adam as the optimizer and categorical cross entropy as the loss function.

V. PERFORMANCE EVALUATION

We apply the following performance metrics to evaluate the performance of the classification models [28].

A. CONFUSION MATRIX

Confusion Matrix is a visualization of ground-truth labels versus model predictions. Each cell in the confusion matrix represents any of the following evaluation factors:

1) True Positive (TP) signifies how many positive class samples are predicted correctly.
2) True Negative (TN) signifies how many negative class samples are predicted correctly.
3) False Positive (FP) signifies how many negative class samples are predicted incorrectly.
4) False Negative (FN) signifies how many positive class samples are predicted incorrectly.

B. PRECISION

Precision is the ability of a classifier not to label an instance positive that is actually negative. It is defined as

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2}$$

C. RECALL

Recall is the ability of a classifier to find all positive instances. For each class, it is defined as

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3}$$

D. F1 SCORE

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 score is expressed as

$$\text{F1 score} = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})}. \tag{4}$$

E. ACCURACY

Classification accuracy is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100. The classification accuracy is calculated as

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100. \tag{5}$$

F. CROSS ENTROPY LOSS FUNCTION

In classification model, a loss function is calculated based on the probability of how far the predicted class is from the actual expected value. Cross-entropy loss is used to optimize the model by adjusting the weights during training to minimize the loss tends to 0. Cross-entropy loss function is defined as

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i), \quad \text{for n classes} \tag{6}$$

where, n is the number of classes, t_i is the true label and p_i is the Softmax probability for the i^{th} class [29].

TABLE 2. Accuracy (intra-subject) after applying different improvement steps of outliers removal and preprocessing.

Machine learning algorithms	With outliers	Without outliers	Standard scaling
DT	66.394 %	86.131 %	86.577 %
KNN (K=2)	73.488 %	92.513 %	92.664 %
RF	76.192 %	93.752 %	93.834 %
ET	76.383 %	93.871 %	93.869 %
XGB	73.703 %	90.162 %	90.162 %

VI. SIMULATION AND RESULTS

The proposed movement detection and classification system is developed in Python programming language using a variety of popular data analysis tools and libraries. The libraries include pandas, scipy.io, loadmat, numpy, matplotlib, seaborn. The classifiers are developed using Keras, TensorFlow, and Scikit-Learn tools. The computational environment of simulation, training and testing is based on a 64-bit Windows operating system with an x64-based Intel(R) Core(TM) i7-5500U CPU with 2.40 GHz processor and 8.00 GB of installed RAM. In addition, we leverage the NVIDIA GeForce 840M GPU for efficient computation.

To verify the accuracy of the proposed movement detection and classification, we simulate the proposed methods using Ninapro DB1 which contains sEMG signals of 10 electrodes of 27 intact subjects for 53 movements. To analyze the effect of outliers removal, initially we apply the ML algorithms on the data of subject 1 without applying the outliers removal. As we can see that the accuracy is very low as the data contains outliers. Next, we apply the outliers removal, data manipulation and preprocessing on the data of subject 1. It is found that the accuracy significantly improves after applying the outliers removal. The accuracy further improves by applying the standard scaling as preprocessing. The findings are shown in Table 2.

Then, we apply RFECV for feature selection on the data of subject 1 only. We configure the Random Forest Classifier as estimator, step = 1, and cross validation = 5. Figure 7 shows the five cross-validation results for an increasing number of selected features, demonstrating how the accuracy improves with each emg value. The best number of features is then discovered to be 10, indicating that all the EMG signals of the 10 electrodes contains important features.

Next, to verify the accuracy of movement recognition, we apply the machine learning algorithms on all the EMG data of 27 subjects after outliers removal, data manipulation and preprocessing steps without applying any feature extraction. Table 3 summarizes the performance in terms of

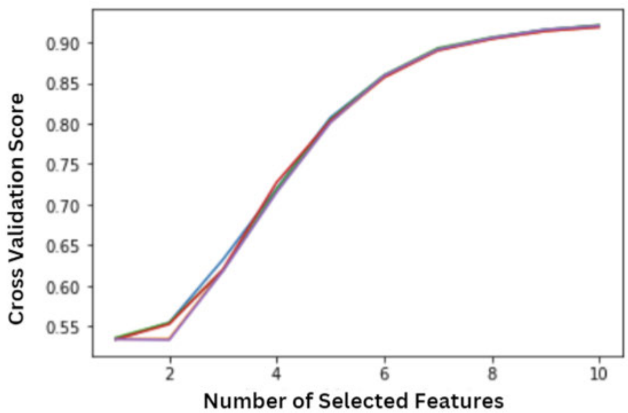

FIGURE 7. Optimal feature selection using cross validation score.

TABLE 3. Accuracy scores (intra/inter-subject) of different machine learning algorithms using raw sEMG of 27 subjects.

Machine learning algorithms	Accuracy (intra)	Accuracy (inter)	Precision (inter)	Recall (inter)	F1 score (inter)
DT	90.425 %	85.241 %	0.853	0.852	0.852
KNN (K=2)	97.458 %	93.174 %	0.931	0.932	0.931
RF	96.880 %	93.177 %	0.932	0.932	0.930
ET	96.755 %	93.376 %	0.935	0.934	0.932
XGB	93.409 %	88.439 %	0.882	0.884	0.881

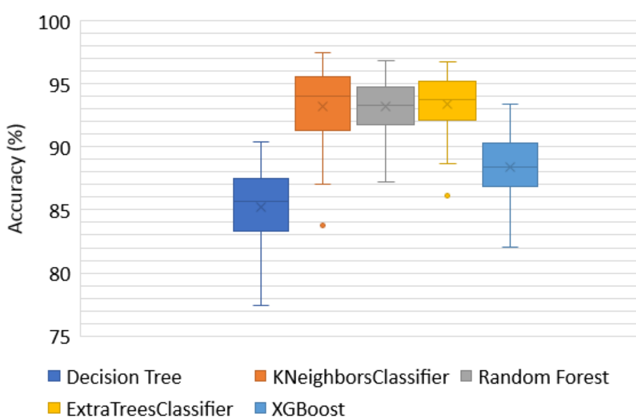

FIGURE 8. Accuracy comparison of different machine learning algorithms using raw sEMG of 27 subjects (Inter-subject).

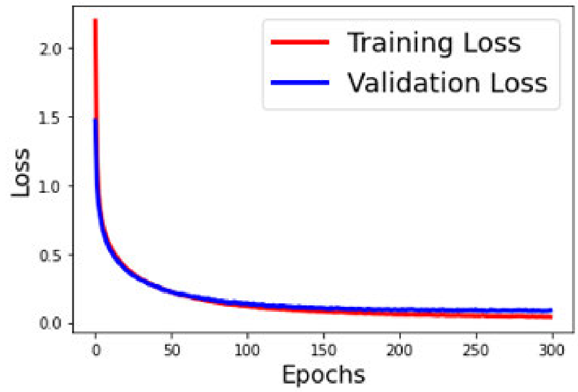

FIGURE 9. Loss curve of ANN-based deep neural network.

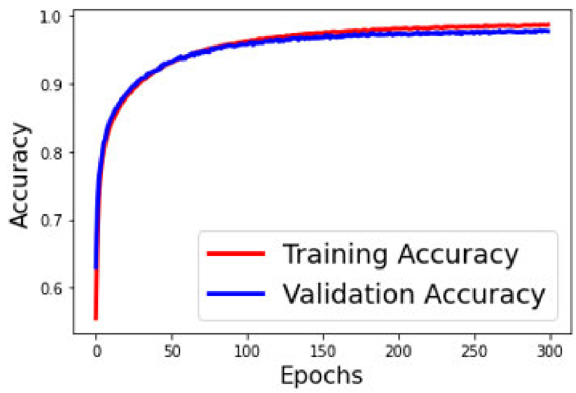

FIGURE 10. Accuracy curve of ANN-based deep neural network.

TABLE 4. Performance scores of deep learning ANN classifier with all 27 subjects.

Evaluation Parameters	Outcomes (intra-subject)	Outcomes (inter-subject)
Training Accuracy	98.71 %	95.32 %
Test Accuracy	97.73 %	93.92 %
Training Loss	0.0397	0.1487
Test Loss	0.0873	0.2098
Precision	0.9773	0.9391
Recall	0.9773	0.9392
F1 Score	0.9770	0.9380

Accuracy, Precision, Recall, F1-score for both intra-subject and inter-subject movement classification. Figure 8 compares the accuracy of inter-subject movement detection and classification using different machine learning algorithms. We observe that the KNN performs the best for intra-subject movement classification with 97.458% accuracy and the ET classifier performs the best for inter-subject with 93.376% accuracy using preprocessed sEMG data.

Next, we apply the ANN based deep learning classification method on the preprocessed sEMG data of all 27 subjects to detect the 53 movements. Figures 9 and 10 show how the loss decreases and the accuracy improves with increasing number of epochs. Table 4 summarizes the results of both intra-subject and inter-subject movement classification performance of the deep learning ANN. We can see the ANN can achieve highest intra-subject accuracy of 97.73% and inter-subject accuracy of 93.92% with preprocessed raw data.

Next, we apply three different feature extraction techniques PCA, ICA and t-SNE to sequentially reduce the dimensionality of the input dataset *emg* which contains 10 EMG features (E1 to E10) of different movements. Figure 11 shows the impact of dimensionality reduction on the accuracy of different classification methods. As we can see in Fig. 11 that the performance of PCA and ICA significantly drop if the dimensionality of the features are reduced below

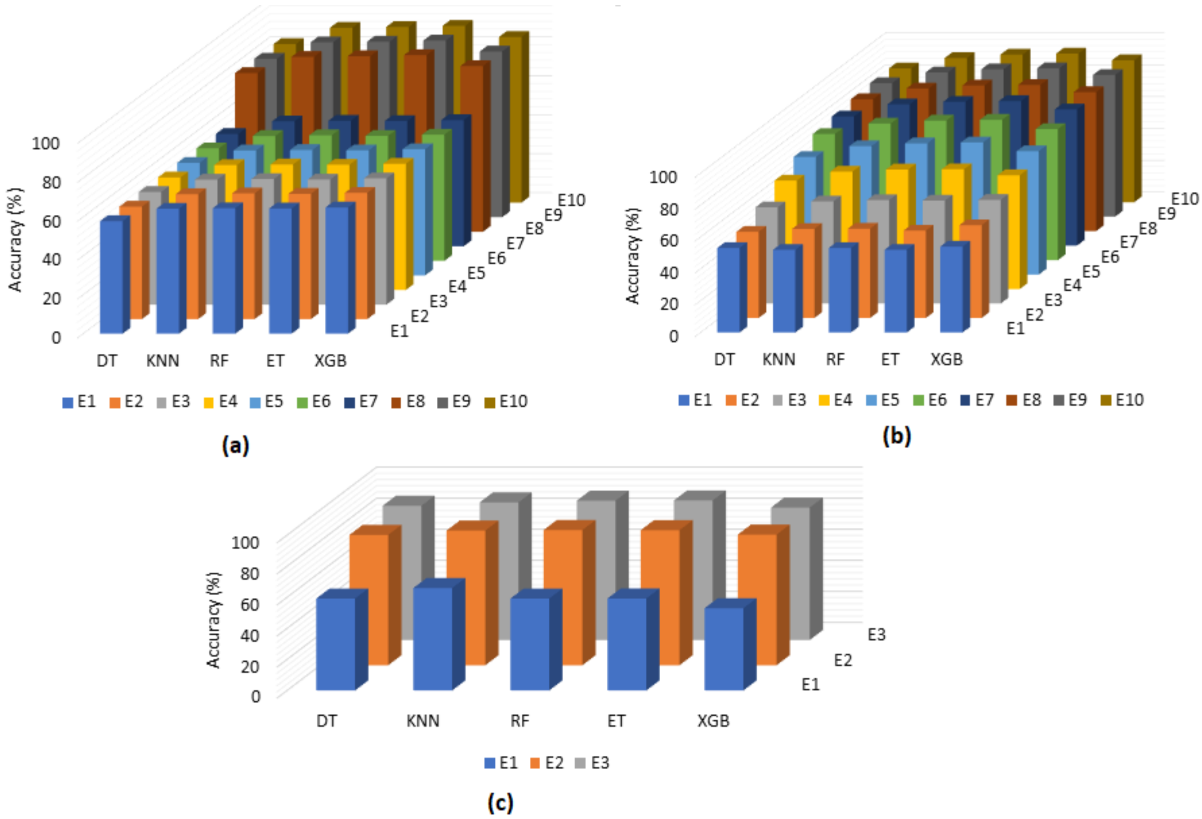

FIGURE 11. Accuracy using different dimensional features; (a) PCA, (b) ICA, (c) t-SNE.

TABLE 5. Classification results (intra or inter-subject) with 8 dimensional features extracted using PCA.

Machine learning algorithms	Accuracy (intra)	Accuracy (inter)	Precision (inter)	Recall (inter)	F1 score (inter)
DT	86.510 %	80.521 %	0.806	0.805	0.805
KNN (K=2)	94.207 %	89.594 %	0.894	0.896	0.893
RF	94.394 %	89.949 %	0.899	0.899	0.896
ET	95.163 %	90.769 %	0.908	0.908	0.905
XGB	89.693 %	84.525 %	0.841	0.845	0.839

TABLE 6. Classification results (intra or inter-subject) with 9 dimensional features extracted using ICA.

Machine learning algorithms	Accuracy (intra)	Accuracy (inter)	Precision (inter)	Recall (inter)	F1 score (inter)
DT	87.172 %	81.988 %	0.820	0.819	0.819
KNN (K=2)	94.394 %	90.437 %	0.903	0.904	0.902
RF	95.600 %	91.734 %	0.917	0.917	0.915
ET	96.257 %	92.661 %	0.927	0.927	0.925
XGB	91.740 %	86.868 %	0.866	0.869	0.864

TABLE 7. Classification results (intra or inter-subject) with 3 dimensional features extracted using t-SNE.

Machine learning algorithms	Accuracy (intra)	Accuracy (inter)	Precision (inter)	Recall (inter)	F1 score (inter)
DT	93.353 %	87.282 %	0.873	0.873	0.873
KNN (K=2)	95.036 %	89.844 %	0.896	0.898	0.895
RF	95.109 %	90.328 %	0.901	0.903	0.901
ET	95.272 %	90.429 %	0.902	0.904	0.902
XGB	91.026 %	84.872 %	0.846	0.849	0.843

TABLE 8. Training time comparison (in seconds) of different machine learning algorithms.

Machine learning algorithms	PCA 8 dimensions	ICA 9 dimensions	t-SNE 3 dimensions	Raw sEMG 10 dimensions
DT	17.895s	16.073s	19.463s	6.776s
KNN (K=2)	2.415s	3.002s	0.079s	2.492s
RF	296.319s	356.119s	552.169s	169.559s
ET	117.722s	148.147s	85.729s	138.323s
XGB	2516.577s	3509.021s	1530.20s	1187.911s

8 and 7, respectively whereas the performance of t-SNE drops if the dimension is below 2. Therefore, to extract the data features of all subjects, we apply linear dimensionality reduction methods PCA to extract 8 dimensional features and ICA to extract 9 dimensional features. We also apply non-linear dimensionality reduction method t-SNE to extract 3-dimensional features of all subjects. Tables 5, 6 and 7 show the intra-subject and inter-subject classification results using

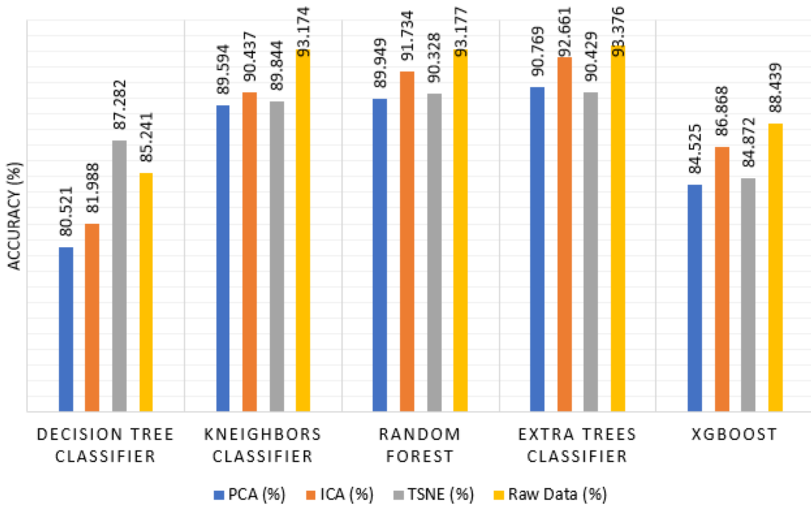

FIGURE 12. Accuracy comparison (inter-subject) using different dimensionality reduction techniques and by using raw sEMG.

TABLE 9. Accuracy and Training time comparison for different Exercises with t-SNE features and raw data.

Algorithms	Exercise A 13 Movements		Exercise B 17 Movements		Exercise C 23 Movements		All 53 Movements	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
With t-SNE features								
KNN (K=2)	88.661%	0.009s	89.956%	0.019s	94.082%	0.035s	89.844%	0.079s
ET	88.421%	13.503s	90.318%	15.302s	94.506%	17.373s	90.429%	85.729s
With raw sEMG								
KNN (K=2)	89.167%	0.268s	91.820%	0.297s	97.266%	1.335s	93.174%	2.492s
ET	91.246%	9.795s	93.499%	23.260s	97.164%	59.279s	93.376%	138.323s
ANN	92.139%	8562.05s	93.286%	12580.57s	97.899%	22565.78s	93.92%	43394.59s

the three feature extraction methods PCA, ICA and t-SNE, respectively. As we can see that the best accuracy using lowest dimensional features (3 dimension) can be achieved using t-SNE feature extraction method. Here we can see that using the 3 dimensional t-SNE features, highest intra-subject accuracy of 95.272% and inter-subject accuracy of 90.429% can be achieved using ET classifier. Figure 12 compares the inter-subject accuracy of the machine learning methods using the preprocessed raw data and by applying different dimensionality reduction techniques. It can be seen in Fig. 12 that the best accuracy can be obtained using the highest dimensional features of the preprocessed raw data whereas the best accuracy with lowest dimensional features can be obtained using the t-SNE features. Thus, it may be concluded from the results of Tables 5, 6 and 7 and Fig. 12 that the KNN, RF and ET classifiers perform with significantly high accuracy for both intra and inter-subject movement classification with preprocessed raw data as well as with 3-dimensional t-SNE features reducing the computational complexity.

Table 8 compares the training time of the extracted low dimensional features and the preprocessed raw data. It is observed that the training time of DT and RF classifier increases with the extracted feature which is not expected. It is also observed that the training time of the ET classifier is significantly reduced by using t-SNE features which can speed up the processing time and can make the system computationally more intelligent. We can also see that KNN performs the best in terms of time accuracy and computational intelligence. KNN requires only 2.492 seconds training time to classify preprocessed raw data of 53 movements of all 27 subjects with inter-subject accuracy of 93.174% and 0.079 seconds to classify the extracted t-SNE features with inter-subject accuracy of 89.844%. Thus, in terms of accuracy and time efficiency, KNN performs the best among all the machine learning algorithms for classification of the 53 movements.

From the above classification results and the time comparison in Table 8, it is observed that among the machine learning algorithms KNN performs the best with lowest computational time. It is also observed in Table 8 that using t-SNE feature extraction, the computational time of KNN and ET is reduced further. Thus, to compare the performance of machine learning based KNN, ET and the deep learning ANN, the accuracy of KNN, ET and ANN are further analyzed for different exercises using both t-SNE features and preprocessed raw sEMG data. The results are included in

TABLE 10. Accuracy and Training time for increasing number of movements with raw sEMG data.

Algorithms	Exercise C Movements 0-7		Exercise C Movements 0-15		Exercise C Movements 0-23	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
KNN (K=2)	98.749%	0.641s	97.778%	0.584s	97.266%	1.335s
ET	98.516%	26.867s	97.631%	23.468s	97.164%	59.279s
ANN	99.367%	14835.723s	98.461%	17799.559s	97.899%	22565.775s

TABLE 11. Accuracy comparison of the proposed methods to the state of the art works.

Algorithms	Dataset	Movements and subjects	Feature extraction	Classifier	Accuracy
Atzori et al [2] 2015	Ninapro DB1	53, 27	MAV5, VAR, WL, HIST, CC, STFT, mDWT	MLP, SVM, KN, LDA	76%
Park et al [3] 2016	Ninapro DB1	6, 27	RMS	CNN	94% (Inter-session)
Tsinganos et al [4] 2018	Ninapro DB1	53, 27	RMS	CNN	70.5% (Intra-session)
AbdelMashee et al [5] 2016	Ninapro DB2	40, 40	MD-DTW	Nearest neighbor based on MD-DTW	89%
Hu et al [6] 2018	Ninapro DB1	52, 27	Raw sEMG	LSTM	75.45% (Intra-session)
Cote-Allard et al [7] 2019	Ninapro DB5	18, 10	Raw data + TL Spectrograms + TL CWT + TL	Deep learning based ConvNet	68.98% 65.10% 65.57%
Y. Li et al [8] 2019	High density EMG database CapgMyo Ninapro DB4	8, 18	MAV	SVM, LDA, KNN	94.2%, 90.2%, 87% below 70%
Tsinganos et al [9] 2019	Ninapro DB1	52, 27	RMS	TCN	89.76% (Intra-session)
Kim et al [10] 2020	Ninapro DB2	50, 20	Spectrogram, PCA	CNN	49.76% (self-decoding) 52.52% (subject transfer)
Y. Li et al [11] 2021	Ninapro DB5	53, 10	MSR TL	SVM AlexNet	67.98% 70.40%
Zou et al [12] 2021	Ninapro DB6	8, 10	Multitime-scale features	TL-MKCNN	97.22% (within-session) 74.48% (cross-subject)
Rahimian et al [13] 2021	Ninapro DB2	50, 40	-	FS-HGR framework	86.3
Padhy et al [14] 2021	Nonapro DB1 CapgMyo DB-b CapgMyo DB-c CSL-HDEG	52, 27 8, 10 12, 10 27, 5	MLSVD	SVM-RBF	75.2%, 75.4% 68.3% 67.7%
Souza et al [15] 2021	Ninapro DB2 and DB3	12,10	LDA	RF	90.13% (with DB2) 85.44% (with DB3)
Proposed	Ninapro DB1	53, 27	t-SNE	KNN, ET	95.04%, 95.27% (intra)
Proposed	Ninapro DB1	53, 27	t-SNE	KNN, ET	89.84%, 90.43% (inter)
Proposed	Ninapro DB1	53, 27	Preprocessed sEMG	KNN, ET, ANN	97.458%, 96.755%, 97.73% (intra-subject)
Proposed	Ninapro DB1	53, 27	Preprocessed sEMG	KNN, ET, ANN	93.174%, 93.376%, 93.92% (inter-subject)
Proposed	Ninapro DB1	23, 27	Preprocessed sEMG	KNN, ET, ANN	97.27%, 97.16%, 97.89% (inter-subject)
Proposed	Ninapro DB1	7, 27	Preprocessed sEMG	KNN, ET, ANN	98.75%, 98.52%, 99.37% (inter-subject)

Table 9 which shows that for different exercises, the accuracy is different and the accuracy above 94% and 97% is obtained for Exercise 3 using t-SNE features and raw data, respectively. As the accuracy for Exercise 3 is comparatively higher, we further analyze the accuracy of Exercise 3 for increasing number of movements. Table 10 shows the accuracy and the training time for increasing number of movements using KNN, ET and ANN. As we can see that the KNN performs the best with 97.266% accuracy for 23 movements of Exercise C with the lowest training time of 1.335s using raw data. It is also observed that ET requires comparatively more time to obtain 97.164% accuracy and ANN requires extensively high processing time of 22565.775*s* to get the accuracy of 97.899% for 23 movements. Table 11 compares the accuracy of the proposed methods to the state of the art works in the literature. As we can see that our proposed methods can obtain significantly high accuracy for a large number of inter-subject and intra-subject movements classification compared to other works in the literature.

VII. CONCLUSION

In this paper, we have proposed sEMG-based movement detection and classification for hand prostheses using the data of Ninapro DB1 of 53 hand movements of 27 intact subjects. First, we have proposed the outliers detection and removal to remove the unnecessary data points. Then, we have applied data manipulation methods like data splitting, relabeling, and concatenation to rearrange the data for easy processing. Next, we have applied standard scaling on the input variables and one hot encoding (for ANN) on the output variables. Then, to reduce the computational complexity and time, dimensionality reduction methods PCA, ICA and t-SNE have been applied to extract 8 dimensional, 9 dimensional and 3 dimensional features, respectively. Then, we have applied five machine learning algorithms and the deep learning ANN on the preprocessed raw data and the extracted features for movement classification. Among the ML algorithms, KNN performs the best in terms of accuracy, dimensionality reduction and least processing time. Using the extracted t-SNE features, KNN shows 89.844% inter-subject and 95.04% intra-subject accuracy while using the preprocessed sEMG data, KNN shows 93.174% inter-subject and 97.458% intra-subject accuracy in classifying 53 movements. Using the preprocessed raw sEMG of 53 movements, ANN can obtain the highest accuracy of 93.92% for inter-subject and 97.73% for intra-subject classification. It has also been observed that using KNN on preprocessed raw data, it is possible to obtain 98.749% and 97.266% inter-subject accuracy for 7 and 23 movements, respectively while using ANN, it is possible to obtain 99.367% and 97.899% inter-subject accuracy for 7 and 23 movements, respectively. However, ANN requires an extensively high computational time of about 43394.59s to train the preprocessed raw data of 53 movements while KNN requires only 2.492s training time. Again, KNN can perform with high accuracy in a reduced computational time of 0.079s using the t-SNE feature extraction. Thus, it may be concluded that among the proposed methods, KNN performs the best with high accuracy in the least computational time. Therefore, the proposed methods recommend choosing machine learning-based algorithms for real-time applications such as controlling prosthetic hands whereas deep learning-based algorithms can be chosen for offline applications due to their high accuracy. To improve the quality and impact of the work in the future, it is recommended to collect adequate datasets for different types of disabilities of the amputee subjects. The results should also be verified using datasets transformed into the time-frequency domain.

REFERENCES

[1] W. Li, P. Shi, and H. Yu, "Gesture recognition using surface electromyography and deep learning for prostheses hand: State-of-the-art, challenges, and future," *Frontiers Neurosci.*, vol. 15, Jan. 2021, Art. no. 621885.
[2] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A.-G. M. Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo, "Characterization of a benchmark database for myoelectric movement classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 73–83, Jan. 2015.
[3] 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.
[4] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "Deep learning in EMG-based gesture recognition," in *Proc. PhyCS*, 2018, pp. 107–114.
[5] M. AbdelMaseeh, T. W. Chen, and D. W. Stashuk, "Extraction and classification of multichannel electromyographic activation trajectories for hand movement recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 6, pp. 662–673, Jun. 2016.
[6] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, and W. Geng, "A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition," *PLoS ONE*, vol. 13, no. 10, 2018, Art. no. e0206049.
[7] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Jan. 2019.
[8] Y. Li, Q. Zhang, N. Zeng, J. Chen, and Q. Zhang, "Discrete hand motion intention decoding based on transient myoelectric signals," *IEEE Access*, vol. 7, pp. 81630–81639, 2019.
[9] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "Improved gesture recognition based on sEMG signals and TCN," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1169–1173.
[10] 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, Oct. 2020.
[11] Y. Li, W. Zhang, Q. Zhang, and N. Zheng, "Transfer learning-based muscle activity decoding scheme by low-frequency sEMG for wearable low-cost application," *IEEE Access*, vol. 9, pp. 22804–22815, 2021.
[12] Y. Zou and L. Cheng, "A transfer learning model for gesture recognition based on the deep features extracted by CNN," *IEEE Trans. Artif. Intell.*, vol. 2, no. 5, pp. 447–458, Oct. 2021.
[13] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, "FS-HGR: Few-shot learning for hand gesture recognition via electromyography," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 1004–1015, 2021.
[14] S. Padhy, "A tensor-based approach using multilinear SVD for hand gesture recognition from sEMG signals," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6634–6642, Dec. 2021.
[15] J. O. de Oliveira de Souza, M. D. Bloedow, F. C. Rubo, R. M. de Figueiredo, G. Pessin, and S. J. Rigo, "Investigation of different approaches to real-time control of prosthetic hands with electromyography signals," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20674–20684, Sep. 2021.
[16] M. Atzori. (2014). *Ninapro Repository*. [Online]. Available: http://ninapro.hevs.ch
[17] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Sci. Data*, vol. 1, no. 1, pp. 1–13, 2014.
[18] (Jan. 22, 2021). *Feature Selection Techniques in Machine Learning*. Accessed: Dec. 2, 2022. [Online]. Available: https://www.geeksforgeeks.org/feature-selection-techniques-in-machinelearning/
[19] K. Vipin. (Dec. 30, 2020). *Dimensionality Reduction—PCA vs LDA vs T-SNE*. Accessed: Dec. 2, 2022. [Online]. Available: https://medium.com/analytics-vidhya/dimensionality-reduction-pca-vs-lda-vs-t-sne-681636bc686
[20] S. Rajvi. (Sep. 13, 2020). *Dimensionality Reduction using Principal Component Analysis (PCA)*. Accessed: Dec. 2, 2022. [Online]. Available: https://medium.com/analytics-vidhya/dimensionality-reduction-using-principal-component-analysis-pca-41e364615766

[21] K. P. Ashwini. *Diving Deeper Into Dimension Reduction With Independent Components Analysis (ICA)*. Accessed: Dec. 2, 2022. [Online]. Available: https://blog.paperspace.com/dimension-reduction-with-independent-components-analysis/

[22] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–12, 2008.

[23] L. Rokach and O. Maimon, "Decision trees," in *Data Mining and Knowledge Discovery Handbook*. Cham, Switzerland: Springer, 2005, pp. 165–192.

[24] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Confederated Int. Conferences Move Meaningful Internet Syst.* Cham, Switzerland: Springer, 2003, pp. 986–996.

[25] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[26] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Mar. 2006.

[27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[28] B. Aayush. *Performance Metrics in Machine Learning [Complete Guide]*. Accessed: Sep. 21, 2022. [Online]. Available: https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide

[29] W. Rose. *Cross-Entropy Loss and Its Applications in Deep Learning*. Accessed Sep. 2, 2022. [Online]. Available: https://neptune.ai/blog/cross-entropy-loss-and-its-applications-in-deep-learning

NAFE MUHTASIM HYE received the B.Sc. degree in electrical and electronic engineering (EEE) from the Ahsanullah University of Science and Technology (AUST).

He is currently a Graduate Research Assistant with the Department of EEE, AUST. His research interests include artificial intelligence, machine learning, deep learning, biomedical signals, and image processing. He was one of the founding members of the university's Mars Rover team, which finished 26th out of more than 65 competitors globally and fourth in Asia, the European Rover Challenge, and Poland, supported by NASA and the Mars Society. He also completed an internship in corporate banking and banking information technology from one of the most reputed banks in Bangladesh.

UMMA HANY received the B.Sc. degree in electrical and electronic engineering from the Ahsanullah University of Science and Technology, in 2004, the M.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology (BUET), in 2009, and the Ph.D. degree in electrical and electronic engineering (EEE) from the Bangladesh University of Engineering and Technology, in 2017.

After her graduation, she served with IICT, BUET, as a Hardware Engineer, and Huawei Technologies Bangladesh Ltd., as a Core Network Engineer. She was a Ph.D. Visiting Scholar with the Canadian Commonwealth Scholarship Program, University of Saskatchewan, Canada. She was with the Department of EEE, University of Asia Pacific, and the Bangladesh Army International University of Science and Technology, as a Lecture, an Assistant Professor, and an Associate Professor. Currently, she is an Associate Professor with the Department of EEE, Ahsanullah University of Science and Technology. She has several publications in peer-reviewed journals and conferences. Her research interests include artificial intelligence, wireless sensor networks, wireless communications, localization, pattern recognition, biomedical signals, and image processing.

SUMIT CHAKRAVARTY (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in electrical and computer engineering. He is currently an associate professor of electrical and computer engineering. He also served as a Postdoctoral Researcher for UPENN. He has used his expertise in signal processing machine learning and communications in his role as a principal scientist in industry and a faculty member in academia. His academic experience includes working at NASA-Goddard, the University of Maryland, and NYIT. His industry experience in engineering and research includes working in various roles such as an instrumentation engineer, a research intern, Siemens CAD and Apex Eclipse Communications, a Principal Scientist of SGT Inc., the Honeywell Research of the Automatic Control Solutions-Advanced Technology Laboratories. He has published multiple journals and peer-reviewed conference papers in venues like *Pattern Recognition*, IEEE SENSORS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and *Journal of Medical Imaging*. He has served as a Guest Editor for the Special Issues on Signal Processing in Wireless Communications and Towards Reliable and Scalable Smart Cities: Internet of Things Meets Big Data and AI of *Electronics* journal.

LUTFA AKTER received the Ph.D. degree in electrical engineering from Kansas State University, USA, in 2010. She is currently a Professor with the Department of Electrical and Electronic Engineering (EEE), Bangladesh University of Engineering and Technology (BUET). Her current research interests include biomedical signal processing, cell-free massive MIMO, wireless sensor deployment, the IoT, and the localization of wireless video capsule endoscopes.

IMTIAZ AHMED (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Vancouver, BC, Canada. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA. He works in the areas of wireless communications, signal processing, and computer networks. After finishing his Ph.D. degree, he was with Intel Corporation, San Diego, CA, USA, as a Wireless Systems Engineer, and Marshall University, Huntington, WV, USA, as an Assistant Professor. His research interests include artificial intelligence-aided physical layer designs, integration of aerial and terrestrial communication networks, communication with THz bands, and intelligent reflecting surfaces.