Grenze International Journal of Engineering and Technology, July Issue

Controlled Prosthesis for upper Limb Amputees using Pattern Recognition Techniques

Anisha.C. D[1] and Arulanand.N[2]
[1-2]PSG College of Technology/Dept. of CSE, Coimbatore, India
Email: ani.c.dass@gmail.com, naa.cse@psgtech.ac.in

Abstract—Upper **limb amputees are individuals who lost their hands due to trauma and injury. Controlled prosthesis based on surface Electromyography (sEMG) signals recovers the lost functionality for upper limb amputees. Several pattern recognition techniques help amputees in controlling prosthesis by classifying different upper limb movements intuitively. The proposed framework performs an analysis of classification of upper limb movements on real time and retrieved surface Electromyography (sEMG) signal data. Band pass filter is used in pre-processing stage and Time Domain features are extracted. The Features selection analysis is also performed wherein Extra Tree classifier and histogram-based features is used for retrieved and real time data respectively. The pre-processed real time and retrieved data with features and classes are fed to the classification stage. The hyperparameters of the classifiers are tuned using Grid Search Method. The classifiers to be stacked are Adaptive Boosting, Gradient Boosting Machine, Quadratic Discriminant Analysis, Linear Discriminant Analysis, K Nearest Neighbor and Random Forest. The properties of the proposed stacking classifier are diverse and same error rate classifiers procured using McNemar's hypothesis testing. The evaluation metrics considered are Accuracy, Precision, Recall and F1 Score. The evaluation results signify that stacking classifier provides a highest accuracy in all experiments.**

Index Terms— **Controlled prosthesis, upper limb movements, sEMG, Stacking ensemble classifier.**

I. INTRODUCTION

The amputee population in India consists of around one million. There is a need for prosthesis with better functionality, intuitive control wherein pattern recognition techniques to be used to discriminate movements based on a surface Electromyography signals obtained through a non-invasive method.[14] Surface Electromyography signals(sEMG) are signals obtained from the muscles and values generated are the difference of electric potentials of the surface electrodes mounted on the skin of the subject arm.[7][9]

Pattern recognition is the method of analysing patterns through machine learning techniques. There are two types of models parametric and non-parametric model. The parametric model assumes the distribution of data to be gaussian and performs classification.[13] The non-parametric model does not have any assumption of the distribution of the data. Ensemble classifiers combines the predictions of diverse classifiers which is known as heterogenous classifiers or combines the predictions of same classifiers known as homogeneous classifiers.

Grenze ID: 01.GIJET.6.2.513_1

The paper is organized as follows section II discusses about the related works, section III presents the proposed work methodologies, section IV presents the results analysis with all results shown along with the inferences and section V provides conclusion and future work.

II. Related Works

In [1] chengcheng Li et al, presents the recognition of hand movements based on nine kinds of actions that are extracted from the surface EMG signals of the fore arm muscles. SVM classifier and Generalized Regression Neural Network (GRNN) classifier has been used for hand movement classification. The nine actions considered in this paper are Rest(re), Hand Close (HC), Hand Open (HO), Pronation (PR), Wrist Extension (WE), Wrist Flexion (WF), Thumbs Up (TU), Thumb Index finger contact (TI), Thumb and Middle finger contact (TM).In [2] Manfredo Atzori et al, focuses on creating a benchmark scientific database for researchers to test the hand movement recognition and force control algorithms. The data has been collected from 27 intact subjects and 11 transradial amputees with 52 movements performed. The 52 movements are categorized into 12 basic movements of fingers, 8 isometric and isotonic hand configurations, 9 basic movements of wrist and 23 grasping and functional movements. The validation of collected data is performed using Support Vector Machine (SVM) classifier, Random Forest (RF) and K Nearest Neighbour (KNN).Yuanfang Wan et al [3], presents a data structure to convert a raw EMG data into matrix format and then Convolutional Neural Network (CNN) classifier has been applied. The CNN is trained using discrete surface electromyography signals generated from three persons with fourteen gestures. The performance of the classifier is also evaluated using Ninapro database. The method has also been applied with data collected from an amputee. The classifier is also trained with the continuous surface electromyography signals to drive the bionic manipulator (robotic hand). The trained classifier is deployed to the bionic manipulator. Sebastian Amsuess et al [4], presents a novel algorithm for controlling the multiple degrees of freedom of a prosthetic hand. The challenges in EMG recording are identifying the recording location and EMG cross talk. The novel proportional estimator proposed in this paper is a solution to overcome the above-mentioned challenge. The common spatial patterns proportional estimator (CSP-PE) is the proposed method presented in this paper. The data has been obtained from ten non-amputees (healthy) subjects and four amputee subjects. F. Riillo et al [5], presents a methodology for the classification of EMG based hand gesture. The two classification methods unsupervised Principal Component Analysis (PCA) and supervised Common Spatial Pattern (CSP) was compared to identify the best classification strategy and its tuning parameters for using in feature extraction process. Six hand gestures have been considered in this analysis. Overlapped Segmentation is the feature segmentation method used. Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Artificial Neural Network (ANN) are the classifiers used in classification stage. Patrik kutilek et al [11] presents a design of the control system of the semiautomatic myoelectric arm which is based on Neural Network (NN). The MATLAB tool has been utilized for signal processing. The designing methods in order to predict the object positions by Artificial Neural Network (ANN) has been presented. Zahit Evren Kaya et al [12] focuses on development of dynamical model and a frame which includes interactions with the objects for a tendon driven based under actuated hand. The method used for modelling the dynamic behaviour of the underactuated fingers is the lagrangian method.

III. Methodologies

Figure 1 depicts the proposed framework with all techniques and analysis process carried out. The Proposed Framework performs analysis on six set of data:

1. Healthy retrieved dataset with Time Domain (TD) Features
2. Amputee retrieved dataset with Time Domain (TD) Features
3. Healthy retrieved dataset with Extra Tree selected Features
4. Amputee retrieved dataset with Extra Tree selected Features
5. Healthy Real Time dataset with Time Domain (TD) Features
6. Healthy Real Time dataset with Histogram based selected Features

A. Data Acquisition

i. Retrieved Dataset

The dataset is retrieved from Ninapro repository which is a benchmark database for EMG data [2]. The database 1 consists of sEMG data collected from 27 healthy subjects and database 2 consists of sEMG data collected from 11 amputees. The number of columns present in the data set is 11 and the number of rows present in the dataset is around 1,00,000.

ii. Real Time Data Collection

The muscle data generated from the subjects while performing movements are collected using surface electrodes, muscle sensor and Arduino board at different muscle positions: Flexor Carpi Radialis (FCR) and Pronotor Teres (PT), Pulmor Longus (PL), Brachioradialis (BR) and Biceps. The movements considered are Index Finger Flexion, Middle Finger Flexion, Ring Finger Flexion, Little Finger Flexion and Thumb Flexion. The movements are performed for a period of 30 seconds.

The another set of real time data is taken from the muscle position Extensor Digitorum (ED) where the movements considered are Hand Open, Index Extension and Little Finger Extension with 5 seconds of 10 readings.

B. Filtering of Acquired data

i. Retrieved Dataset

The Ninapro database 1 dataset does not require any filtering since the dataset itself is bandpass filtered data.

ii. Real Time Collected Data

The real time collected data is filtered using Butterworth bandpass filter since Butterworth filter provides a thin frequency response and bandpass filter provides average value of low pass and high pass filter.

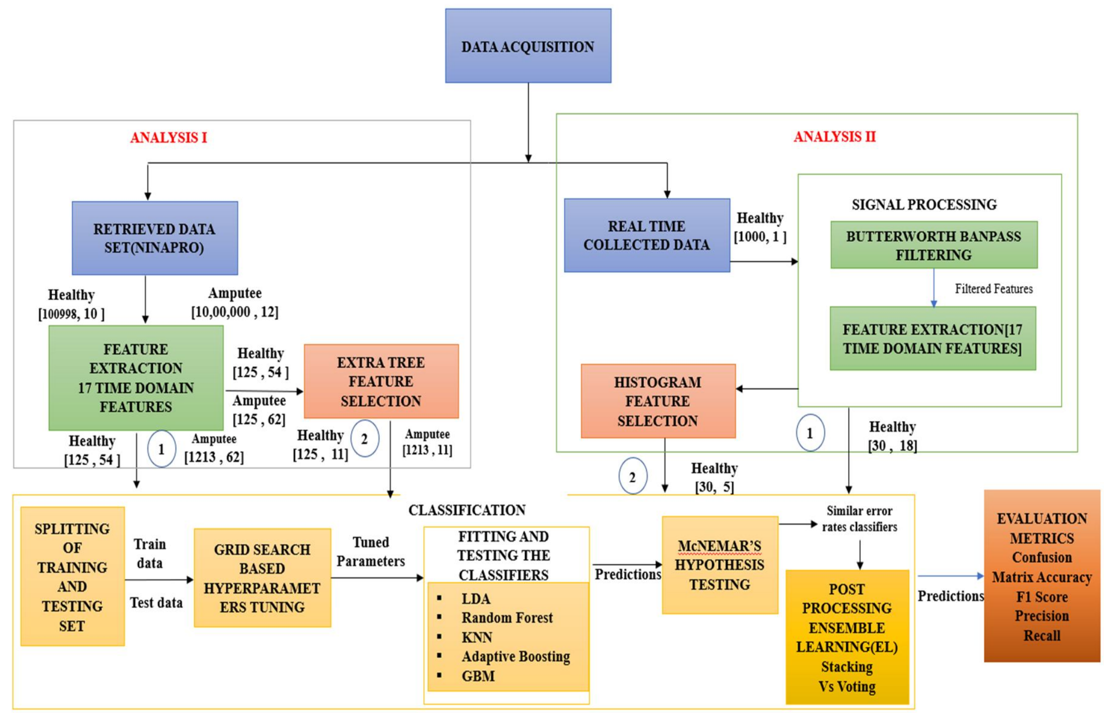

Figure 1 – Block Diagram of the Proposed Framework

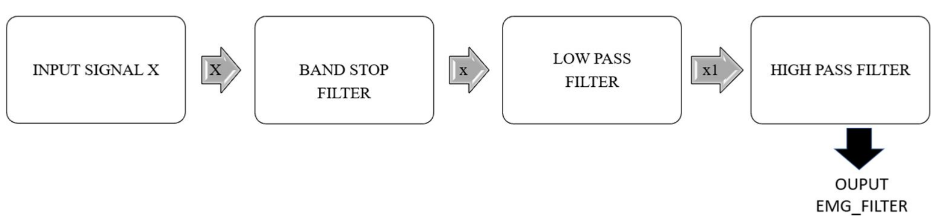

Figure 2 Process Flow of Butterworth Band Pass Filter

C. Feature Extraction

The EMG data is a time series data which will not be useful in classification of movements, hence useful information has to be extracted from EMG data [10][15]. 17 Time Domain (TD) features are extracted for Ninapro dataset and real time dataset and they are: Average Amplitude Change (AAC), Difference Absolute Standard Deviation Value (DASDV), Enhanced Mean Absolute Value (EMAV), Enhanced wavelength(EW), Log Detector(LD), Mean Absolute Value, Mean Absolute Value(MAV), Modified Mean Absolute Value 1(MMAV),Modified Mean Absolute Value 2(MMAV) , Myopulse Percentage Rate(MPR), Maximum Fractal Length (MFL), Root Mean Square(RMS), Simple Square Integral(SSI), Slope Sign Change(SSC), Variance of EMG, Wavelength(W),Willison Amplitude (WA), Zero Crossing(ZC).

D. Feature Selection Analysis

i. Retrieved Dataset

The top ten prominent features are selected from the Time Domain Features based on the Gini Index parameter of the Extra Tree Classifier. The Extra Tree classifier is chosen because the dataset is of high dimension.

ii. Real Time Dataset

The Histogram based Feature selection technique is used for real time data since the dataset is of low dimension.

E. Splitting of Training and Testing data

The data is split into training and testing, the training data is given as input to the classifier for training and the testing data is given as input to the classifier for prediction.

Table 1 specifies the number of data present in the training and testing split for retrieved and real time data.

TABLE I. TRAINING AND TESTING SET SPLIT

DATASET	TRAINING SET	TESTING SET
HEALTHY RETRIEVED DATASET	Number of Instances =100	Number of Instances=25
AMPUTEE RETRIEVED DATASET	Number of Instances = 971	Number of Instances =242
REAL TIME DATASET	Number of Instances =24	Number of Instances=6

F. Hyper parameters Tuning

The important parameters of the classifier known as hyperparameters will be tuned using Grid Search technique in order to provide accurate classification of movements.

i. Hyperparameter Input Space

1. Random Forest Classifier:
   Random Forest: The values in the n_estimators is chosen based on the number of features in the dataset.
   Experiment 1: n_estimators=[1,53]; bootstrap=[True,False]
   Experiment 2: n_estimators=[1,61]; bootstrap=[True,False]
   Experiment 3: n_estimators=[1,10]; bootstrap=[True,False]
   Experiment 4: n_estimators=[1,10]; bootstrap=[True,False]
   Experiment 5: n_estimators=[1,2]; bootstrap=[True,False]
   Experiment 6: n_estimators=[1,2]; bootstrap=[True,False]
2. K-Nearest Neighbor (KNN):
   The values in the n_neighbors are specified based on the formula sqrt(n) where n is the number of instances in the dataset. Parameter ‘p’ is a Boolean value wherein 1 specifies Manhattan distance 2 specifies Euclidean distance
   Experiment 1: n_neighbors:[1,11];p=[1,2]
   Experiment 2: n_neighbors:[1,34];p=[1,2]
   Experiment 3: n_neighbors:[1,11];p=[1,2]
   Experiment 4: n_neighbors:[1,34];p=[1,2]
   Experiment 5: n_neighbors:[1,5];p=[1,2]
   Experiment 6: n_neighbors:[1,5];p=[1,2]
3. Adaptive Boosting Machine (AdaBoost)

The values in the n_estimators is chosen based on the number of features in the dataset and is adjusted based on trial and error method.
Experiment 1: n_estimators:[1,53];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
Experiment 2: n_estimators:[1,61];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
Experiment 3: n_estimators:[1,10];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
Experiment 4: n_estimators:[1,10];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
Experiment 5: n_estimators:[1,17];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
Experiment 6: n_estimators:[1,2];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]

4. Gradient Boosting Machine (GBM):
   The values in the n_estimators is chosen based on the number of features in the dataset and is adjusted based on trial and error method.
   Experiment 1: n_estimators:[1,53];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
   Experiment 2: n_estimators:[1,61];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
   Experiment 3: n_estimators:[1,10];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
   Experiment 4: n_estimators:[1,10];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
   Experiment 5: n_estimators:[1,17];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
   Experiment 6: n_estimators:[1,2];learning_rate=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]

ii. Hyperparameter Output Space

Table 2 presents the hyperparameter output for Random Forest , KNN ,AdaBoost and GBM for all six experiments.

iii. Working Principle of Grid Search Hyperparameter Tuning

The Grid Search Hyperparameter Tuning performs a complete exhaustive search on all possible combinations present in the hyperparameter space and provides the optimal parameters for the classifier.

Table II. Hyperparameter output Space

CLASSIFIER HYERPARAMETERS OUTPUT	
Random Forest: Experiment 1: n_estimators=47; bootstrap=True Experiment 2: n_estimators=35; bootstrap=False Experiment 3: n_estimators=4; bootstrap=False Experiment 4: n_estimators=34; bootstrap=False Experiment 5: n_estimators=4; bootstrap=False Experiment 6: n_estimators=1; bootstrap=False	Adaboost: Experiment 1:n-estimators=1 ;learning_rate=0.1 Experiment 2:n-estimators=7 ;learning_rate=0.23 Experiment 3:n-estimators=66;learning_rate=0.4 Experiment 4:n-estimators=7 ;learning_rate=0.1 Experiment 5:n-estimators=2 ;learning_rate=1 Experiment 6:n-estimators=1 ;learning_rate=0.1
K-Nearest Neighbor (KNN): Experiment 1: n_neighbors:1; p=1 Experiment 2: n_neighbors:15; p=2 Experiment 3: n_neighbors:1; p=1 Experiment 4: n_neighbors:1; p=1 Experiment 5: n_neighbors:2; p=1 Experiment 6: n_neighbors:1; p=1	GBM Experiment 1:n-estimators= 7;learning_rate=0.5 Experiment 2:n-estimators=42 ;learning_rate=0.3 Experiment 3:n-estimators=4 ;learning_rate=0.5 Experiment 4:n-estimators=48 ;learning_rate=0.4 Experiment 5:n-estimators= 7;learning_rate=0.4 Experiment 6:n-estimators= 3;learning_rate=0.1

G. Fitting the classifier

Retrieved Data set and Real Time collected data

The classification stage is similar for the retrieved dataset and real time dataset. The classifiers considered are categorized into parametric and non-parametric models. Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are parametric models are considered in this analysis as per the assumption that sEMG signals shows the gaussian distribution characteristics. The Non parametric models are Random Forest, Adaptive Boosting classifier wherein decision tree and Support Vector Machine (SVM) are boosted, Gradient Boosting Machine (GBM) are considered because it handles high dimensional data in an efficient manner and it works for any distribution data. The ensemble technique is used because it provides higher accuracy as it combines the predictions of same error rate diverse classifiers. The diversity in the stacking ensemble is ensured based on its type of classifier and the consistency of the stacking ensemble is ensured by combining same error rate classifiers which makes the stacking ensemble robust in nature. The stacking ensemble is compared with voting ensemble classifier.

1. Working Principle of Random Forest Classifier

   The training data is given as input to multiple decision trees and the testing data is predicted by each decision tree and the prediction is combined using majority voting of decision trees. The decision tree is constructed in a parallel manner.
2. Working Principle of Linear Discriminant Function (LDA)

   LDA forms a discriminant function from the training set, it constructs the lower dimensional space discriminant function based on the mean vectors, covariance, eigen values and eigen vectors of the training set. The prediction of testing set is performed in lower dimensional space.[6]

   i. Calculate the Mean Vector m for all classes residing in the dataset.
   ii. Create Scatter Matrices for two categories inside the class boundary and between the classes.
   iii. Compute eigen vectors and corresponding eigen values of each class.
   iv. Sort the eigen vectors in decreasing eigen values, choose n eigen vectors with the highest eigen value.
   v. Convert the samples into new dimensional space based on m*n eigenvector matrix.
3. Working Principle of K-NEAREST NEIGHBOR(KNN)

   KNN is a nearest neighbour algorithm learns the complete training set and for each test data point it finds the k nearest neighbours and finds the distance and maps the class label and the frequent class label are given as output.[8]

   i. The KNN computes the squared distance from the test data point to current data points.
   ii. The squared distance is sorted and the ranking is provided, based on the number of nearest neighbours.
   iii. The Labels are given for the nearest neighbour's distance and the mode of the labels is provided as the result of the test data point.
4. Working Principle of QDA

   In Quadratic Discriminant Function (QDA) Analysis, Covariance matrix are not identical and the covariance is computed for each class. The quadratic function with second order terms is the discriminant function.
5. Working Principle of Adaptive Boosting Classifier

   The weak learners considered to be boosted in Adaptive Boosting Classifiers are decision Tree which is known as decision stumps and Support Vector Machines with different kernels. The weak learners are chosen such that it supports the sample weighting. Initially each instance in the dataset is weighted as 1/n where n is the number of instances present in the dataset. weighted error rate is computed as the number of wrong predictions divided by the total number of predictions. The wrong predictions are related with the instance's weights. The weight of decision tree is updated based on the two conditions:

   a) The weights remain the same if the instances are correctly classified
   b) The updated weights are computed as old weight ** weight of this tree if the instances are incorrectly classified.
6. Working Principle of Gradient Boosting Machine (GBM)Classifier

   Initially a decision tree is trained on the train data and the prediction is made for the test data. The residual error of this decision tree is computed wherein residual error = actual value – predicted value and the residual error is saved as new actual value. Similarly training of all specified decision trees is performed and predictions is computed. The final prediction is the addition of predictions of all specified decision trees.

H. Hypothesis Testing – McNemar's Test

McNemar's Test is used on the paired nominal data which is applied to 2 X 2 Contingency Matrix. The two classifiers predictions are categorized into correct and incorrect and converted into the contingency matrix. The Null Hypothesis specifies that the two classifiers are of same error rate and the alterative hypothesis specifies that the two classifiers are of different error rate. The hypothesis pass or fail is based on the p value. The Null and Alternate Hypothesis specified in this experiment are:
Null Hypothesis (H0): Same proportions errors (fail to reject H0)

Alternative Hypothesis (H1): Different proportions of errors (reject H0)

I. Post Processing: Ensemble Stacking Classifier

i. Properties of the classifiers in the Ensemble
   1. All the classifiers should be diverse in their type.
   2. All the classifiers should have same error rates which is procured from McNemar's Hypothesis Testing.

ii. Working Principle of Ensemble Stacking Classifier
   1. Different models are trained on the complete data set.
   2. Predictions from models are considered as meta features or probabilities to the metaclassifier.
   3. Metaclassifier provides the final prediction based on the learning from the meta features or probabilities obtained from the various models.

A. Comparison with Voting classifier

i. Working Principle of Hard Voting Classifier/ Soft Voting Classifier
   1. The different classifiers are trained on the training set and the model's predictions on the testing set are combined using hard voting or soft voting classifier.
   2. Hard voting classifier makes predictions based on the mode operations and soft voting classifier makes predictions based on the probabilities of the outcomes.

J. Testing and Evaluation of the classifier

The trained model is tested using the tested data and the prediction is obtained which is then evaluated using evaluation metrics: Confusion Matrix, Accuracy, Precision, Recall and F1 score.

i. **Confusion Matrix**: It constructs a matrix based on
   a. True Positive: Actual class = Predicted class
   b. True Negative: addition of rows and columns expect the rows and columns of that specific class.
   c. False Positive: sum of all values in the column of the specific class except the corresponding class column value.
   d. False Negative: sum of all values in the row of the particular class except the value of the particular class in that row.

ii. **Accuracy:** It is the measure of correct number of predicted instances to the total number of instances

iii. **Precision:** It is the measure of correctly classified instance to the total sum of True Positive (TP) and False Positive (FP).

iv. **Recall:** It is the measure of correctly classified instance to the total sum of True Positive (TP) and False negative (FN).

v. **F1 Score**: It is the weighted average measure of Precision and Recall.

IV. RESULT ANALYSIS

A. Confusion Matrix

i. Healthy Retrieved Dataset: It consists of 12 classes of movements. The test data consists of 10 classes for movements for Random Forest, Linear Disciminant Analysis, Adaboost SVM for experiment 1 and 3; 11 classes of movements for Adaboost Decision Tree, GBM, KNN for experiment 1 and 3 QDA (experiment 1); 12 classes of movements for QDA (experiment 3).

1. 10 classes of movements**:**
   [0: Index Flexion, 1: Index Extension; 2: Middle Flexion; 3: Middle Extension ;4: Ring Flexion; 5: Ring Extension; 6: Little Finger Extension; 7: Thumb adduction; 8: Thumb abduction; 9: Thumb Extension]
2. 11 classes of movements**:**
   [0: Index Flexion, 1: Index Extension; 2: Middle Flexion; 3: Middle Extension ;4: Ring Flexion; 5: Ring Extension; 6: Little Finger Extension; 7: Thumb adduction; 8: Thumb abduction; 9: Thumb Flexion, 10: Thumb Extension]
3. 12 classes of movements**:**

[0: Index Flexion, 1: Index Extension; 2: Middle Flexion; 3: Middle Extension ;4: Ring Flexion; 5: Ring Extension;6: Little Finger Flexion;7: Little Finger Extension; 8: Thumb adduction; 9: Thumb abduction; 10: Thumb Flexion,11: Thumb Extension]

ii. Amputee Retrieved Dataset**:** It consists of 10 classes of movements in test data for all classifiers.
[0: Index Flexion, 1: Index Extension; 2: Middle Flexion; 3: Middle Extension ;4: Ring Flexion; 5: Ring Extension;6: Little Finger Flexion;7: Little Finger Extension; 8: Thumb adduction; 9: Thumb abduction]

iii. Real Time Dataset**:** It consists of 3 classes of movements in test data for all classifiers.
[0: Hand open, 1: Index Extension, 2: Little Finger Extension]

B. Inferences From The Results

i. In table 3, the class label 2 (Middle Flexion) is misclassified as Index Flexion. Out of 25 Instances in test set 24 Instances are correctly classified.

ii. From the Figure 3- Analysis I, it is clear that the stacking ensemble classifier accuracy is higher than individual classifiers accuracy for healthy and amputee dataset.

iii. From the Figure 3- Analysis II, it is evident that the stacking ensemble classifier accuracy is same as the Gradient Boosting Machine (GBM) Classifier for retrieved healthy features selected dataset and the stacking ensemble classifier accuracy is higher than the individual classifiers accuracy for amputee features selected dataset.

iv. From the Figure 3- Analysis III, it is obvious that the Stacking ensemble classifier accuracy is higher than the individual classifiers for time domain and histogram features.

v. From the real time data collection, First set of data collected at different positions for finger movements concludes that Individual finger movements classification is difficult with one channel data, therefore second set of data is collected the main muscle Extensor Digitorum provides an improved accuracy of 83%.

vi. From the Figure 4, it is clear that the Stacking ensemble classifier accuracy is higher than the Stacking ensemble classifier accuracy is higher than the Voting ensemble classifiers.

TABLE III. HIGHEST PRECISION, RECALL AND F1 SCORE OF THE STACKING CLASSIFIER

CLASS LABELS	CORRESPONDING MOVEMENTS	PRECISION	RECALL	F1 SCORE	SUPPORT (CORRECT/TOTAL)
0	Index Flexion	0.67	1.00	0.80	2/2
1	Index Extension	1.00	1.00	1.00	3/3
2	Middle Flexion	1.00	0.67	0.80	2/3
3	Middle Extension	1.00	1.00	1.00	2/2
4	Ring Flexion	1.00	1.00	1.00	3/3
5	Ring Extension	1.00	1.00	1.00	2/2
6	Little Finger Extension	1.00	1.00	1.00	3/3
7	Thumb adduction	1.00	1.00	1.00	1/1
8	Thumb abduction	1.00	1.00	1.00	3/3
9	Thumb Extension	1.00	1.00	1.00	3/3
Accuracy	-	-	-	0.96	24/25
Macro Average	-	**0.97**	**0.97**	**0.96**	**24/25**
Weighted Accuracy	-	**0.97**	**0.96**	**0.96**	**24/25**

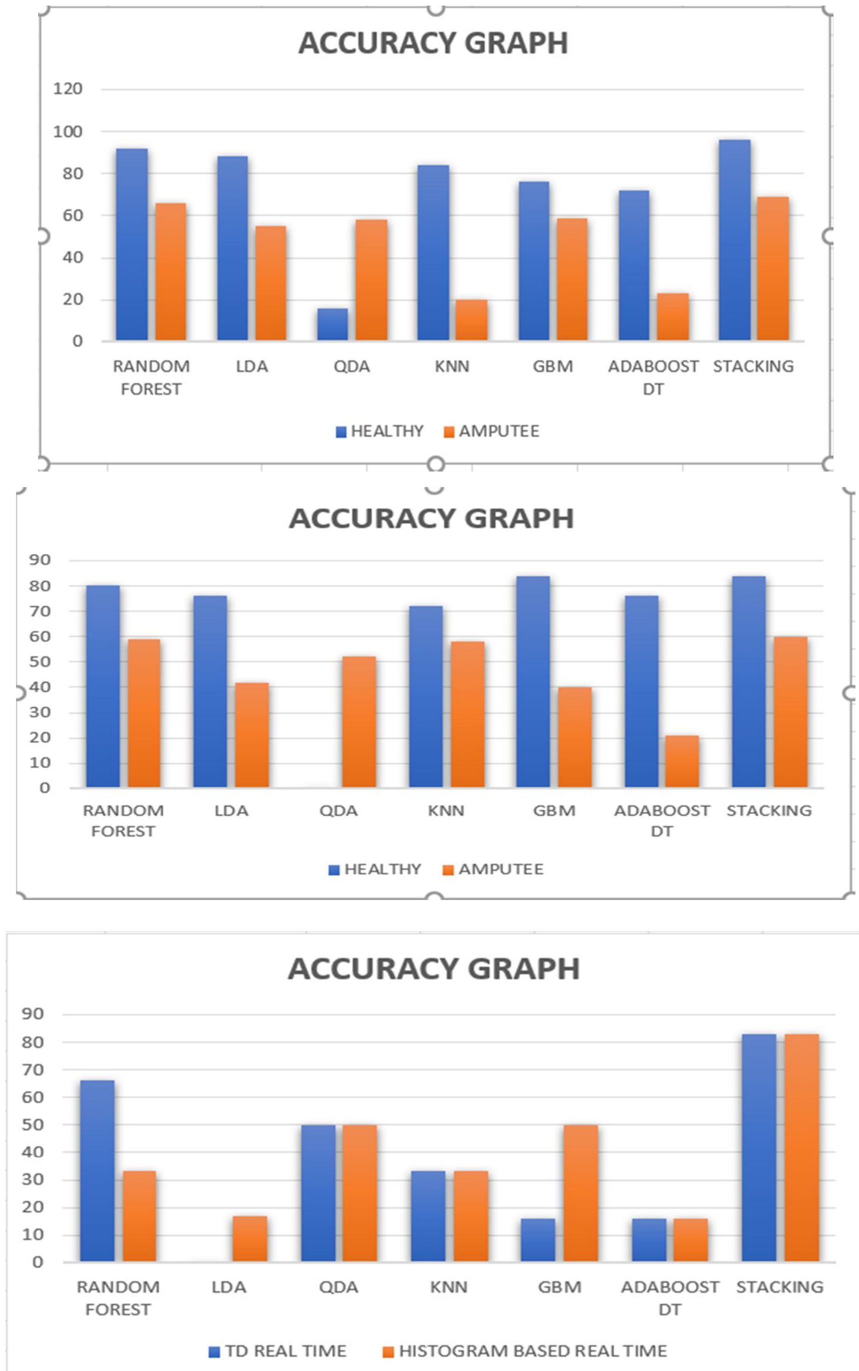

Figure 3- Accuracy Graph of Analysis I (Top), Analysis II(Middle), Analysis III (Bottom)

Figure 3- Analysis I represent Accuracy Graph of all classifiers with TD features, Analysis II Accuracy Graph of all classifiers with selected features for retrieved dataset. And Analysis IIII presents accuracy graph for all classifiers for Time Domain (TD) Features and Histogram based selected features of real time healthy subject data.

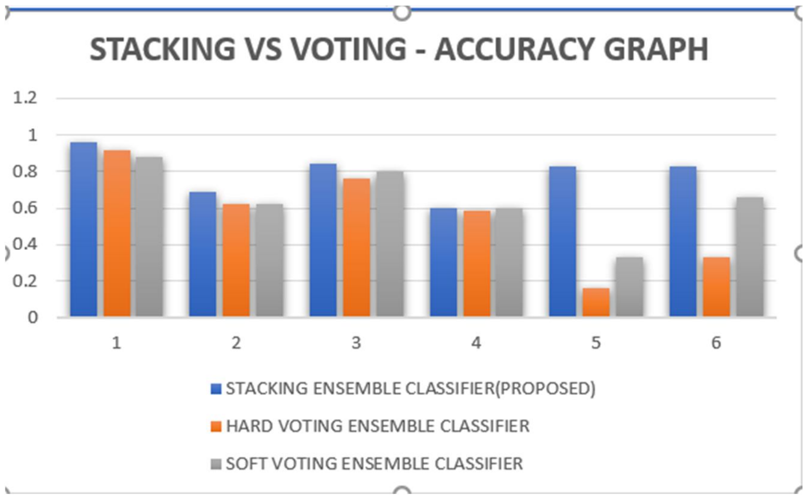

Figure 4 Accuracy Graph of Stacking and Voting Classifier

V. CONCLUSION AND FUTURE WORK

Amputee are individuals who have lost the arm due to trauma or injury. Prosthesis is an artificial arm replaces the lost hand functionality. The Pattern recognition is essential in prosthesis in order to make movements intuitively without the control switching to be done by the user. The prosthesis is sEMG based which means it receives the Electromyography signals (muscle signals) from the surface electrodes placed on the skin of the arm. The proposed framework conducted six experiments wherein four experiments is related with retrieved healthy and amputee dataset with time domain features and Extra Tree classifier selected features and two experiments is associated with real time data of healthy subject with time domain features and histogram- based features. The classifiers considered in this analysis are chosen based on two types Parametric (LDA, QDA) and Non-Parametric models (Random Forest, KNN, GBM, Adaboost). The stacking ensemble classifier is used for classification of movements because single classifier predictions is better than aggregating multiple classifier predictions. The evaluation results signify that stacking classifier outperforms in all experiments than the individual classifiers. The highest accuracy of 96%, Precision (0.97), F1 Score (0.96), Recall (0.97) is obtained in experiment 1 with healthy retrieved dataset with time domain features for stacking ensemble classifier. The future enhancement is to automate the complete process of feature extraction and classification of movements and to deploy the automated process to the robotic arm prototype. The additional enhancement includes developing a mobile application that acts as an interface for the user and the arm.

REFERENCES

[1] Chengcheng Li et al, "Surface Emg Data Aggregation Processing for Intelligent Prosthetic Action Recognition."- Neural Computing and Applications, Springer, 24 September 2018

[2] Manfredo Atzori et al , "Electromyography Data for Non Invasive Naturally-Controlled Robotic Hand Prostheses." – Scientific Data, 15 August 2014

[3] Yuanfang Wan et al, "Pattern Recognition and Bionic Manipulator Driving by Surface Electromyography Signals using Convolutional Neural Network."- International Journal of Advanced Robotic Systems, September- October 2018.

[4] Sebastian Amsuess et al, "A Multi-Class Proportional Myocontrol Algorithm for Upper Limb Prosthesis Control: Validation in Real-Life Scenarios on Amputees"- IEEE Transactions on Neural Systems and Rehabilitation Engineering

[5] F. Riillo et al, "Optimization of EMG Based Hand Gesture Recognition Supervised Vs Unsupervised Data Preprocessing on Healthy Subjects and Transradial Amputees."- Biomedical Signal Processing and Control 14(2014) 117–125, 27 March 2014

[6] Marina M.C. Vidovic et al, "Improving the Robustness of Myoelectric Pattern Recognition for Upper Limb Prostheses by Covariate Shift Adaptation."- IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 24, No. 9, September 2016

[7] Pawel Wojtczak et al, "Hand Movement Recognition Based on Biosignal Analysis."- Engineering Applications of Artificial Intelligence 22 (2009) 608–615, 1 August 2008
[8] Ali H. Al Timemy, et al, "Improving the performance against Force Variation of EMG Controlled Multifunctional Upper Limb Prostheses for Transradial Amputees.", IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE-2014-00179, R2),2014
[9] Chris Wilson Antuvan et al, "Role of Muscle Synergies in Real-Time Classification of Upper Limb Motions using Extreme Learning Machines."- Journal of Neuro Engineering and Rehabilitation (2016)
[10] Rami N. Khushaba et al, "Toward Improved Control of Prosthetic Fingers Using Surface Electromyogram (EMG) Signals."- Expert Systems with Applications 39 (2012)
[11] Patrik Kutilek et al, "Myoelectric Arm using Artificial Neural Network to Reduce Cognitive Load of the User."- Neural Computing and Applications, Springer, 29 October 2015
[12] Zahit Evren Kaya et al, "Modeling and Simulation of an Anthropomorphic Hand Prosthesis with an Object Interaction." - Computer Methods and Programs in Biomedicine 183(2020), 3 April 2019
[13] Kianoush Nazarpour et al, "A Note on the Probability Distribution Function of the Surface Electromyogram Signal."- Brain Research Bulletin90 (2013) 88–91, Elseiver, 25 June 2012
[14] Nawadita Parajuli et al, "Real Time EMG-Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation."- MDPI: Sensors 2019,19, 4596, 18 September 2019
[15] Nurhazimah Nazmi et al, "A Review of Classification Techniques of EMG signals during Isotonic and Isometric Contraction."- Sensors 2016- MDPI, 5 April 2016.