Educational Administration: Theory and Practice
2024, 30(5), 7366 - 7376
ISSN: 2148-2403
https://kuey.net/

Research Article

Development In Limb Prosthetics Using Machine Learning Algorithms

Subhashree Darshana[1*], Aditi Srivastava[2], Rohan Agarwal[3], Siddharth Swarup Rautaray[4], Manjusha Pandey[5]

[1*,2,3 4,5]School of Computer-Engineering, KIIT (Deemed) University, Bhubaneswar, Odisha, India 751024

***Corresponding Author:** Subhashree Darshana
Email: subhashree.darshanafcs@kiit.ac.in

Citation: Subhashree Darshana et.al (2024), Development In Limb Prosthetics Using Machine Learning Algorithms, *Educational Administration: Theory and Practice, 30(5), 7366 - 7376*
Doi: 10.53555/kuey.v30i5.4164

ARTICLE INFO

ABSTRACT

Machine learning is revolutionizing the healthcare industry by enabling real-time prediction of ortosis needs and enhancing prosthetic selection, training, fall detection, and socket temperature control. Artificial limbs can help people with disabilities to regain their ability to move. An important method in artificial limb technology is pattern identification of limb movement intention.
This paper highlights how machine learning algorithms play an important role in upper and lower limb amputation.
For upper limb EEG and EMG processing enhances classification accuracy in prosthetics, enhancing upper limb movement control performance and control performance in paraplegics with elbow replacements, demonstrating an 8.9% improvement ratio in real-time applications.
For lower limb prosthetics we implemented various algorithms namely KNN, SVM, and QDA, Random Forest, Random Forest with estimator 100, and Random Forest with activator 60 to test the best accuracy.

Keywords: Brain-Computer Interface (BCI), EEG, EMG, EEG paradigm, SVM, K-N-N, Bi-LSTM, quadratic discriminant analysis (QDA), Random Forest

INTRODUCTION

A prosthetic is a device that replaces a missing lower limb, aiding in mobility, balance, and functional tasks. The various types include hemipelvectomy, hip disarticulation, and Symes prostheses. Customized to fit the individual's residual limb, it meets specific needs.
Bioengineers use machine learning and pattern recognition to separate human
locomotion into motor tasks and phases, detecting and manipulating neuro-mechanical features using embedded sensors to ensure controller stability and reliability.
A mind-controlled prosthetic limb is made by hybrid brain-computer interfaces(BCIs), which integrate EEG and machine learning. The brain produces electrical impulses that are captured by BCIs and converted into a machine-learning model when a movement has to take place. By analyzing brainwave patterns, the ML model provides exact instructions to the prosthetic device so that it can replicate the user's intended movement. This technology can decode complicated hand movements and restore function and independence for individuals who have lost upper limb function. It also offers prospective benefits, including tailored training and continual learning.
Lower-limb amputations (LLA) are a prevalent issue with high mortality rates and lower quality of life. Machine learning techniques like Naïve Bayes and Random Forest are being used to classify arm and hand motions.

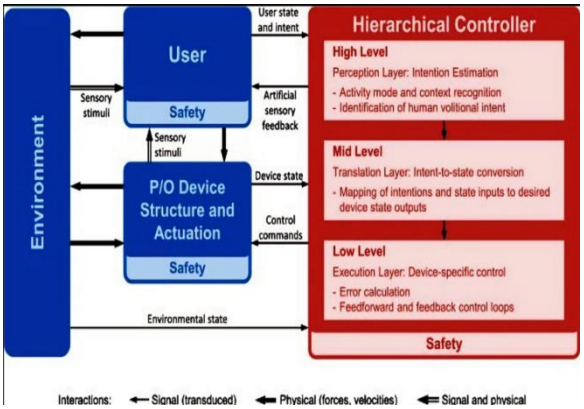
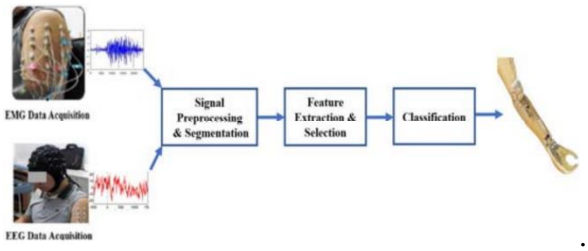


Fig 1: Working structure for lower limb prosthesis

Machine learning holds promise for enhancing the precision and dependability of prosthetic control systems, particularly in intricate movements. Previous methods, such as EMG-driven control, mechanical control, microprocessor-controlled prostheses, variable damping prostheses, and passive prostheses, have various limitations such as low signal-to-noise ratio, high sensitivity to noise, and face difficulty in accurately detecting muscle activation patterns during complex movements. By incorporating machine learning, these methods can address limitations in user comfort, range of motion, accuracy in detecting intent, and complexity of movements in lower limb prosthetics.

II.METHODOLOGY-

The suggested hybrid model is divided into four stages: gathering EEG and EMG data; pre-processing and segmentation; feature extraction and selection; and classification. 1. Model Schematic



A. Collected data

The data used in this study were collected by Li et al.[3,4]. This data is explained as the below mentioned: The average residual length was found to be 25.50 ± 4.20 cm (measured from below the scapula) in the humerus test in 4 elderly men (mean age 41.50 ± 7.05). patients[4]. Five movement groups were tested: open hand (HO), closed hand (HC), wrist pronation (WP), wrist supination (WS), and weakness (NM).In order to prevent patients from suffering from physical discomfort or mental sickness, that might result in distortions and damage the quality of the signal, each movement is held for five seconds, with a five-second rest (such as NM) scheduled in between two adjacent movements. During the recording period, each subject performed five motions (one of HO, HC, WP, and WS) 10 times. According to the EEG and sEMG data acquisition diagram [3] displayed in Figure 1, SEMG and EEG were acquired concurrently during the task. A high-speed sEMG device with 32 mono-polar electrodes was used to gather sEMG signals.

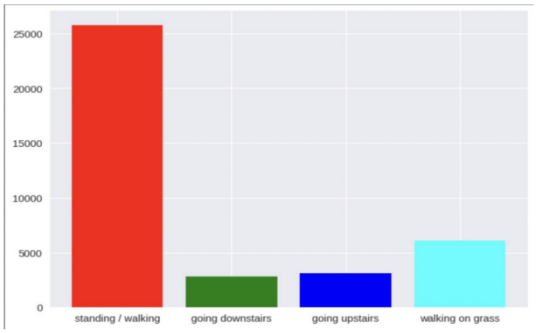
Each subject's remaining arm. EEG signals are obtained from a 64-channel EEG head coupled with the Neuroscan system, which represents the most advanced technique for EEG data acquisition and analysis.When the target action was indicated on the screen, each subject executed it; when the screen vanished, they stopped. When a moving target image appears/disappears on the screen, the EEG data recorder can draw a vertical line on the EEG recording as the start/end point of the movement. The 64-channel Al-Ag Cl electrodes are distributed according to the 10-20 system standard, which is a recognized method of providing scalp electrodes for EEG recording [3].

Now for Lower Limb knowledge training data, includes data from 8 subjects (Subject001-Subject008), and test data includes data from 4 subjects (Subject009-Subject012). Each topic has profiles associated with ratings, grades, and sessions.

The x time archive contains the timestamps of the accelerometer and gyroscope measurements, and the y time archive

contains the timestamps of the text. Units are seconds, time x is sampled at 40 Hz, time y is sampled at 10 Hz. The Y file contains letters 0 through 3, where letter 0 represents standing or walking on the ground, 1 represents descending stairs, 2 represents climbing, and 3 represents walking on grass.

First, we combine all x data for each subject into a single file with the time column. Again we get the y data. Then, for each subject, we combine the data containing the meter and gyroscope with the data containing the time column (outside). Here, when externally joining, we only have a label for a quarter of the x samples due to the rate difference (40 Hz for x and 10 Hz for y).
To fill in the missing text, we use the filled interpolation technique, where the missing label of the model is the label of its previous model. We now put all the information into a form that will be used for training and use. By plotting this data, we found that there was a large uncertainty in the data set and that the log of most data samples was 0. Therefore, we use the weight cross-entropy on unemployment to compensate for this.



B. Data preprocessing and segmentation

The EEG signal is amplified at 1000 Hz and filtered with a band pass filter with a frequency range of 0.05 to 100 Hz. The SEMG signal is filtered using a band pass filter ranging from 10 to 500 Hz and sampled at a rate of 1024 Hz. Subtract the average of each channel in the EEG signal and extract the EEG epochs for each movement group along the start-end line. Each EEG period for the movement group contains 5 seconds of EEG data plus 120 milliseconds of EEG data before movement onset, for a total of 5120 data points [3]. The conversion factor to detect artifacts such as eye movements and blinks in each epoch of the EEG data was equal to 1.2 times the value of the epoch data.
In addition, a 50 Hz notch filter was used to remove noise in the electrical lines of EMG and EEG.

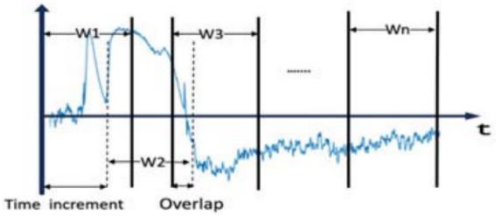


Fig. 2. Segmentation scheme overlap

Of 150 ms, 200 ms, 250 ms and 100 ms increment overlapping segmentation scheme is used. Figure 2 shows an example of how to complete the overlapping segmentation process.

a. Training and validation section/segmentation of preprocessing for lower limb:

1) Session extension: We adopt session extension method for training-validation section; Here we consider the validation set and the remaining data as 2 times the entire process learning. As a training set We use this data to train the model and evaluate its performance. We found that we could do better using the window approach, where we define the window size and treat the data extension of the window size as a single window.
2) 2) Window size method: If we randomly distribute the data set, the sorting behavior disappears. We split the data into windows and then split the data into training and validation. In this way we ensure that data connections are protected. We worked with different sizes and decided that the size that looked best was 120.

C. Feature Extraction

This step involves the calculation of three sets of time domain, frequency domain, and entropy-based IT features, as well as auto-regressive (AR) coefficients for each segment. Time domain features include Root Mean Square (RMS), Wavelength Length (WL) and William Amplitude (WAMP), while frequency features include Auto regressive (AR) coefficients, Frequency Ratio (FR), Set mean and Set mean of the signal frequency (MMNF, MMDF). Model entropy and log energy entropy are entropy-based features calculated for each partition. The features of each channel are extracted from segments of different lengths and then combined to form feature vectors as shown in figure 3.

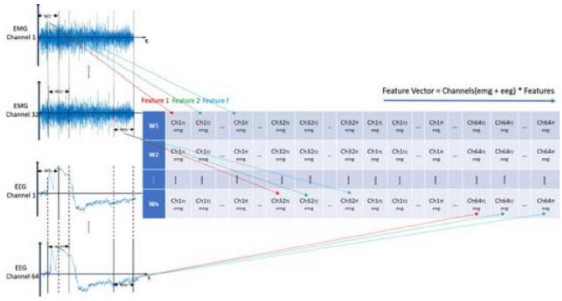


Figure 3. Create feature vectors consisting of 32 EMG channels and 64 EEG channels; where -f- features and -n-time windows (-W--) are extracted from each channel.

1) Auto regressive coefficients

Auto regressive (AR) model calculates each sample signal from past samples plus free
Noise is defined as a combination of time errors. The AR model can be represented as:

$$x_k = \sum_{i=1}^{p} a_i x_{k-i} + e_k \qquad (1)$$

where a_i is the AR coefficient, p is the order of the AR model, and e_k is the additional external noise which is independent according to the previous perspective, p is the order of the AR model. In this study, we used the fourth order in the AR model as suggested in [9, 14] and the sixth order..

2) Entropy-based properties

Entropy is the measure of the complexity and randomness of a dynamic system and defines the value of information creation [17]. Considering the non linearity and non locality of the signal, nonlinear parameters can be used to describe the evolution of EEG and EMG signals [15]. We determine the standard entropy and log energy entropy for each part.

a) Sample entropy

The definition of sample entropy is the negative of the width of the logarithm of the chance that two similar m points in the sequence will not change at the next point in the tolerance interval r, where the result is calculated without taking into account the self-comparison. The vector is constructed by first embedding the scalar time series {x1, x2,..., xi,..., xn} into the extended m-dimensional space in order to compute the entropy model. This is similar to it:

$$x(p) = [x(p+K)]_{k=0}^{m-1}, p = 1,2,\dots,n-m+1 \qquad (2)$$

The probability $B^m(r)$ that two arrays are equal to m points is calculated by calculating the average of pairs of vectors whose distance is smaller than the tolerance r. Similarly, the embedding dimension of $A^m(r)$ is m + 1 [17].

The formula for sample entropy is:

$$Sample\ Entropy(x,m,r) - -\ln(A^m(r)/B^m(r)) \qquad (3)$$

The standard deviation of the initial time, or (0.15 - 0.25) * SD, is used to choose the tolerance r. We used m = 2 and r = 0.25 * SD in this investigation.

b) Logarithmic energy entropy:

In general, the entropy of finite length discrete random variables, x = [x(0), x(1), ..., x(N-1), with probability The distribution function is denoted as p(x).
The definition is as follows:

$$H(x) = -\sum_{i=0}^{N-1} p_i(x) log_2(p_i(x)) \qquad (4)$$

the Log Entropy entropy of x is given by:

$$H_{LogEn}(x) = -\sum_{i=0}^{N-1} (log_2(p_i(x)))^2 \qquad (5)$$

3) Root mean square (RMS):

The root mean square is related to the standard deviation and can be said as follows:

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2} \quad (6)$$

4) Willison Amplitude:
Willison Amplitude (WAMP) is the difference between EEG or sEMG signal amplitudes between two adjacent segments that exceeds a threshold to reduce noise effects. It can be shown that:

$$WAMP - \sum_{t=1}^{N} f(|x_t - x_{t+1}|) \quad (7)$$

$$f(x) = \begin{cases} 1, & if\ x > threshold \\ 0, & otherwise. \end{cases}$$

In this study, we use a threshold value of 0.025 for (WAMP).

5) Waveform Length:
Waveform length (WL) refers to the length of a waveform over time.

$$WL = \sum_{n=1}^{N-1} |x_{n+1} - x_n| \quad (8)$$

6) Modified Medium Frequency (MMDF) is a frequency that divides the spectrum into two regions of equal amplitude. It can be shown that:

$$\sum_{j=1}^{MMDF} A_j = \sum_{j=MMDF}^{M} A_j = \frac{1}{2}\sum_{j=1}^{M} A_j \quad (9)$$

where A_j is the EEG amplitude spectrum at frequency bin j [7].

4) Modified Mean Frequency:
Modified Mean Frequency (MMNF) represents the mean frequency, computed by dividing the total of amplitude spectrum and frequency by the sum of spectral density. This relationship can be expressed as:

$$MMNF = \sum_{j=1}^{M} f_j A_j / \sum_{j=1}^{M} A_j \quad (10)$$

where f_j is the frequency of spectrum at frequency bin j [7].

5) Frequency ratio (FR) :
It is the ratio of the low frequency component and the high frequency component of the signal.
The equation is defined as:

$$FR = \sum_{j=LLC}^{ULC} P_j / \sum_{j=LHC}^{UHC} P_j \quad (11)$$

among them, ULC and LLC represent the upper and lower cuts at low frequencies, respectively, and UHC and LHC represent the upper and lower cuts at high frequencies, respectively. The initial distribution of low frequency and high frequency can be determined experimentally [14].

D. SELECTING THE FEATURES

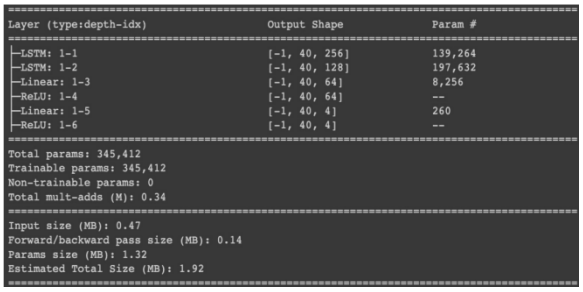
The attributes should be able to convey the qualities of individuals with dissimilar bodies. Load estimations should be considered in immediate applications. Feature selection selects a subset of essential features based on evaluation criteria, resulting in improved learning outcomes, lower costs, and better sample definition. Feature selection is performed in such a way that retrieved characteristics that are unsuitable for classification are deleted.

a. Model Architectures for lower limb

In the first stage of the terrain information project, we decided on the random forest classifier. In the next step, we will discuss LSTM and bidirectional LSTM [5] as it is based on data, i.e. time data. The main idea behind using bidirectional LSTM is that the current terrain estimate is based not only on some past measurements (xyz accelerometer and gyroscope values), but also on some past measurements in the sequence. We take this into account when using the dimension view method.
All models we use use batch size 512. Below are the different LSTM and Bi-LSTM models we covered in the 3 stages of round 2.

1) Bi-LSTM + LSTM + 40WS: Bi-LSTM + LSTM + 40WS Model
• Input fed to Bi-LSTM layer (hidden size = 128), then LSTM layer (hidden size = 128).
• LSTM output fed to thick layer (64x64).

• Activation function: ReLU of thick layer and product of thick layer.

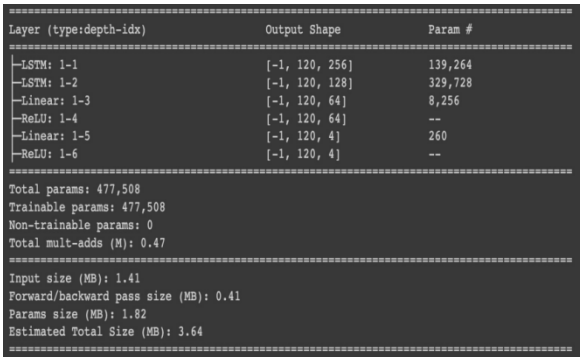


Layer (type:depth-idx)	Output Shape	Param #
├─LSTM: 1-1	[-1, 40, 256]	139,264
├─LSTM: 1-2	[-1, 40, 128]	197,632
├─Linear: 1-3	[-1, 40, 64]	8,256
├─ReLU: 1-4	[-1, 40, 64]	--
├─Linear: 1-5	[-1, 40, 4]	260
├─ReLU: 1-6	[-1, 40, 4]	--

Total params: 345,412
Trainable params: 345,412
Non-trainable params: 0
Total mult-adds (M): 0.34

Input size (MB): 0.47
Forward/backward pass size (MB): 0.14
Params size (MB): 1.32
Estimated Total Size (MB): 1.92

2) Bi-LSTM + LSTM (number of layers = 2, throughput = 0.5) + 40WS: In this model, we consider single-layer Bi-LSTM and 2-layer LSTM. Both have the same size, 128. 0.5 output is added to the first layer of the LSTM.



Layer (type:depth-idx)	Output Shape	Param #
├─LSTM: 1-1	[-1, 120, 256]	139,264
├─LSTM: 1-2	[-1, 120, 128]	329,728
├─Linear: 1-3	[-1, 120, 64]	8,256
├─ReLU: 1-4	[-1, 120, 64]	--
├─Linear: 1-5	[-1, 120, 4]	260
├─ReLU: 1-6	[-1, 120, 4]	--

Total params: 477,508
Trainable params: 477,508
Non-trainable params: 0
Total mult-adds (M): 0.47

Input size (MB): 1.41
Forward/backward pass size (MB): 0.41
Params size (MB): 1.82
Estimated Total Size (MB): 3.64

3) Bi-LSTM + LSTM(threshold = 2, loss = 0.5) + 120WS: The window size we received in the previous two models is 40 and is the only data corresponding to 1 second. Later we realized that 1 second corresponds to a scale that is too short to see the ground, because a person can only move one step per second. Therefore, in this model, we choose 120, that is, 3 seconds of data, as the window size and consider the same architecture as (2).

b. Model Training for lower limb

There are 8 subjects in our training profile, and xyz accelerometer and xyz gyroscope

measurements were recorded from different sections for each subject. In the first stage, we search for omission points and use this to classify the data. But since we have already determined the size of the window, we divide the entire object into several windows, each of which is a model. Now this formula has been modified and divided into 70% and 30% for training and validation purposes.

1) Cross-entropy weighting: In the previous stage, we used SMOTE technology to reduce the number of classes by creating synthetic data. This does not help us improve performance because the model is over fitted with synthetic data.

Therefore, at this stage, we consider weighted cross-entropy to avoid random objects. We know that class 0 is the majority class, and the other classes are minority classes. More weight is given to the minority class because the distribution of the minority causes a greater loss than the wrong class of the majority. In weighted cross-entropy, the weight of the cluster is given by:

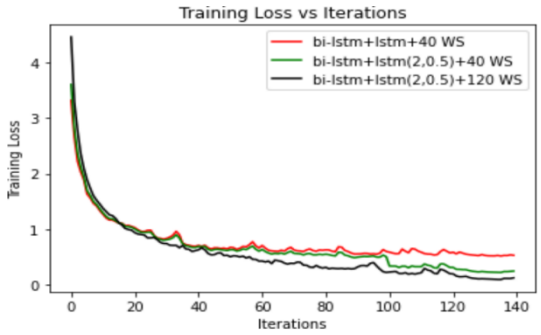
$$Class\_x\_Weight = 1 - \frac{\text{Class X Training Samples}}{\text{Total Training Samples}}$$

2) Adam Optimizer and StepLR Scheduler: We train the model only 140 times due to limited budget. We calculate the learning curve from 0.001 to 0.1 with Adam optimizer [6], but the loss is not the same. The model with learning rate 0.001 has the least learning loss, but after 50 epochs the gradient step size is out of range, so we use the timer using stepLR with a step size of 50 epochs and a gamma value of 0.5.

Below is a graph of learning loss versus time.

c. Model Selection for lower limb

In the first stage of this study, we used random forest classification [1], a classical machine learning technique. In the next stage, we use Bi-LSTM and LSTM architectures with thick layers. The hyperparameters we set include batch size, window size, latent size (for LSTM layers), training cost, number of LSTM layers, and output cost. Below is the learning curve over time for all three production units we used.



E. CLASSIFICATION

The suggested model's performance was evaluated using three groups: quadratic discriminant analysis (QDA), K-nearest neighbor (KNN), and quadratic support vector machine (quadratic SVM), with five categories of top sports identified throughout.

1) Quadratic Discriminant Analysis:

The formula used here is:

$$\delta_k(x) = -\frac{1}{2}\log|\sum k| - \frac{1}{2}(x-\mu_k)^T \sum k^{-1}(x-\mu_k) + \log\pi_k \quad (12)$$

Σk matrix number is the difference matrix that should not be x is the data set and μ_k the average is the estimate of the data, the discrimination law is given by the following formula:

$$\delta_K(x) = \min_{1\le k\le K} d_k(X) \leftrightarrow \max_{1\le k\le K} p(k/x) \quad (13)$$

2) K - Nearest Neighbor:

The nearest neighbor (KNN) method is a popular easy classification strategy that detects the k nearest neighbors in training data and returns the same information at the anticipated time. The KNN approach creates a new unclassified model for a class where the majority of its nearest neighbors are K [23].

3) Support Vector Machine:

SVM uses a non-linear kernel to build an optimum hyperplane in high-dimensional space, resulting in a non-linear classifier despite using linear learning methods,

considerably improving learning and generalization capabilities.

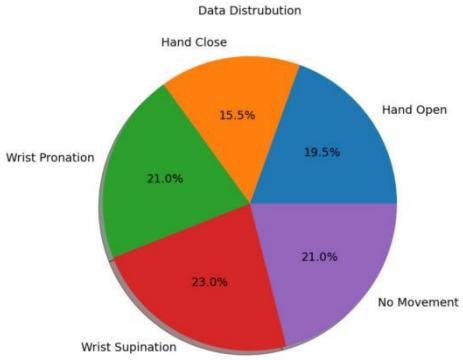
$$c = \sum_t a_t k(s_t x) + b \quad (14)$$

where S_i is the support vector, a_i is the weight, and b is the bias used to classify the feature vector x. Here k is the kernel function. In this study, quadratic support vector machines were tested. The numerical representation of a quadratic support vector machine is given by:

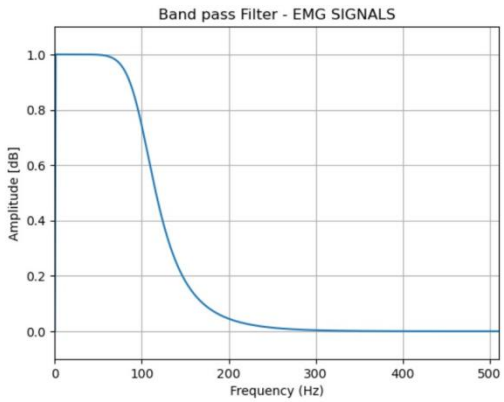
$$k(s_t.x) = e^{-\|s_t - x\|^2} \quad (15)$$

RESULT:

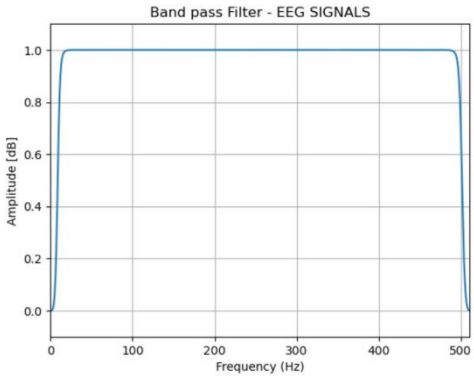
For upper limb prosthesis we have used EEG and EMG signals and the below graphs show the data distribution for five different movements namely open hand, closed hand, wrist pronation, wrist supination and no movement.



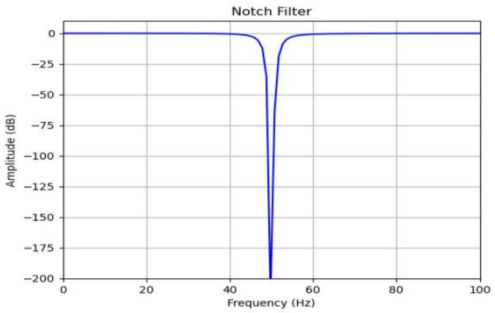
Frequency response of a band pass filter for EMG signal. The filter begins to attenuate the signal at around 100 Hz and 200 Hz and by 500 Hz the signal is attenuated by about 20dB. This particular filter is designed to pass a relative narrow band of frequencies within the typical range of an EMG signal.



The below filter is designed to pass a relatively narrow band of frequencies within the typical range of EMG signal.



The notch filter responses show a sharp decrease in amplitude around a specific center frequency. This indicates that the filter effectively attenuates weak signals at the particular frequency range while allowing signals at other frequencies to pass through with minimal attenuation.



b) For lower limbs, we have employed a confusion matrix to ascertain the accuracy.

The confusion matrix evaluates the performance of a K-Nearest Neighbor (KNN) classifier Here the rows represent the actual classes. Columns represents predicted classes by the KNN model. Values in each cell represent the number of instances.

By looking at the diagonal values (top left to bottom right), we see:

Model Predicts 7 Class 0 Instances

Predicts 4 Class 1 Instances.

By looking at the off -Diagonal Values

Misprediction of 2 instances of class 0 as class 1.

Misprediction of 3 instances of class 1 as class 0.

Overall, model correctly classifies 11 out of 16 instances.

Misclassification of 5 instances between two classes.

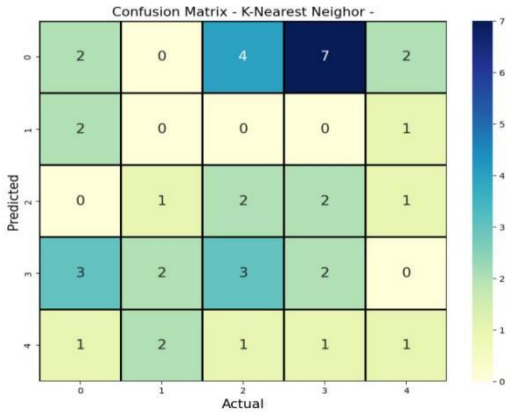


Fig: Confusion Matrix using KNN

The confusion matrix evaluate the performance of SVC model with non-linear kernel on a classification task. The model compares the number of times the classifier correctly predicted different classes, with high diagonal numbers indicating good classification and low off diagonal numbers indicating minimal confusion.

SVC Model Confusion Matrix Analysis

• Model incorrectly classified 2 instances of class 4 as class -2.

• Performs well for class 3 (predicted 8 correctly), struggles with classes 0 and 1 (predicted only 0 and 2 correctly).

• Confusions class 4 with class -2.

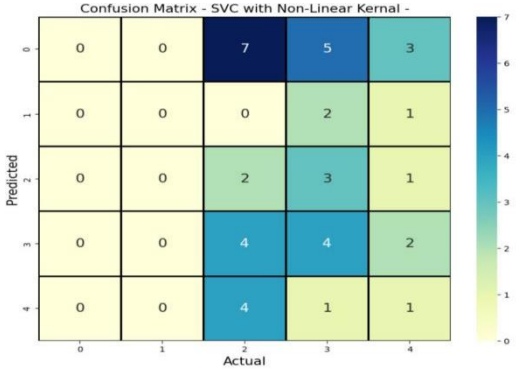
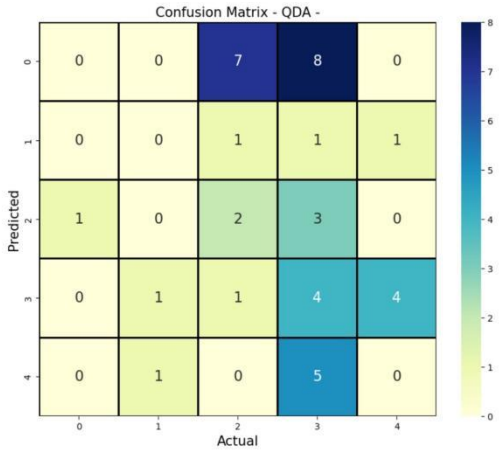


Fig: Confusion matrix using Non Linear Kernel

Confusion Matrix Analysis of QDA Classifier
• Compares correctly and incorrectly predicted classes.
• Good classification is identified by high diagonal elements.
• Low off-diagonal numbers indicate rarely confuses between classes.
• Classifier performs well for class O (predicted 8 correctly), struggles with class 1 (predicted only 2 correctly).



Algorithm	Accuracy	Precision
XG-Boost	**53·7**	**53·3**
Random Forest with estimator 100	**92.0**	**91.9**
Random Forest with estimator 60	**91.92**	**91.89**
Random Forest	**91.08**	**90.9**

CONCLUSION:

In this paper, we found that using Random Forest with estimator 100 performs the best of the lower limb data set and gives an accuracy of 92% and it generalizes the best on dataset. We used a confusion matrix to check the performance of algorithms namely KNN, Non linear Kernal and QDA and found out the prediction of each algorithm. Machine learning algorithms are being used to enhance walking patterns in prosthetic knee systems. However, issues like control strategies, portable power supplies, and lightweight actuators still need improvement. Further research is needed to understand commonly used techniques and develop intelligent prostheses with pattern recognition models for real-time performance validation for betterment in the medical science field.

REFERENCES:

1. D.G.K. Madusanka, L.N.S. Wijayasingha, R.A.R.C. Gopura, Y.W.R. Amarasinghe, and G.K.I. Mann, “A Review on Hybrid Myoelectric Control Systems for Upper Limb Prosthesis,” in Moratuwa Engineering Research Conference, Moratuwa, Sri Lanka, April 2015.
2. J. T.D. Lalitharatne, K. Teramoto, Y. Hayashi, K. Kiguchi, “Towards Hybrid EEG-EMG based control approaches to be used in Bio-robotics applications: current status, challenges and future directions, ” Journal of Behavioral Robotics, Pal
3. X. Li, O.W. Samuel, X. Zhang, H. Wang, P. Fang, and G. Li, “A motion-classification strategy based on sEMG-EEG signal combination for upper-limb amputees,” Journal of NeuroEngineering and Rehabilitation, vol. 14, no. 2, pp. 1-13, 2017.
4. O.W. Samuel, X. Li, Y. Geng, P. Feng, S. Chen, G. Li, “Motor imagery classification of upper limb movements based on spectral domain features of EEG patterns, ” Engineering in Medicine and Biology Society, 39th Annual International Conference of the IEEE, Seogwipo, South Korea, July 2017.
5. G. McGimpsey, T. Bradford, “Limb Prosthetics Services and Devices,”Bioengineering Institute Center for Neuroprosthetics, Worcester Polytechnic Institution, Worcester, May 2010.
6. S. Siuly, Y. Li, “Improving the Separability of Motor Imagery EEG Signals Using a Cross Correlation-Based Least Square Support Vector Machine for Brain–Computer Interface, ” IEEE Transactions on NeuralSystems and Rehabilitation Engineering vol. 20, no. 4, pp. 526 – 538, July 2012.

7. A. Phinyomark, C. Limsakul, P. Phukpattaranont, “A Novel Feature Extraction for Robust EMG Pattern Recognition, ” Journal of Computing, vol 1, no. 1, pp. 71-80, Dec 2009.
8. A. Phinyomark, P. Phukpattaranont, C. Limsakul, “Feature reduction and selection for EMG signal classification, ”, Expert Systems with Applications, vol 39, no. 8, pp. 7420-7431, June 2012.
9. Shandong University Jinan China chenzhenxin@mail.sdu.edu.cn Yongmei Hu School of Control Science and Engineering Shandong University Jinan China
10. Lara-Barrios CM et al (2018) Literature review and current trends on transfemoral powered prosthetics. Adv Robot 32(2):51–62
11. Lawson B E, Mitchell J, Truex D, et al. (2014). A Robotic Leg Prosthesis: Design, Control, and Implementation. Robotics & Automation Magazine IEEE, 21(4), 70-81.
12. Aeyels B, Peeraer L, Sloten J, Van der Perre G (1992) Development of an above-knee prosthesis equipped with a microcomputer-controlled knee joint: first test results. J Biomed Eng 14:199–202
13. Lobo-Prat, J., Kooren, P. N., Stienen, A. H., Herder, J. L., Koopman, B. F., & Veltink, P. H. (2014). Non-invasive control interfaces for intention detection in active movement-assistive devices. Journal of neuroengineering and rehabilitation, 11, 1-22.
14. Li, X., Chen, S., Zhang, H., Samuel, O. W., Wang, H., Fang, P., ... & Li, G. (2016). Towards reducing the impacts of unwanted movements on identification of motion intentions. Journal of Electromyography and Kinesiology, 28, 90-98.
15. Safari R. Lower limb prosthetic interfaces: Clinical and technological advancement and potential future direction. Prosthetics and Orthotics International. 2020;44(6):384-401.
16. McMullen, D. P., Fifer, M. S., Katyal, K. D., Armiger, R., Hotson, G., Beaty, J. D., ... & Wester, B. A. (2020). Design and preliminary evaluation of an augmented reality interface control system for a robotic arm. Johns Hopkins APL Tech. Dig, 35, 220-230.
17. Billard, A., & Kragic, D. (2019). Trends and challenges in robot manipulation. Science, 364(6446), eaat8414.
18. Darshana, S., Rautaray, S.S., Pandey, M. (2021). AI to Machine Learning: Lifeless Automation and Issues. In: Pandey, M., Rautaray, S.S. (eds) Machine Learning: Theoretical Foundations and Practical Applications. Studies in Big Data, vol 87. Springer, Singapore.
19. Mohanty D., Jena, B. Khuntia,T., Mohanty, P.K., Mohapatra, S., Behera, S. (2024). Green Transit: Harnessing Renewable Energy For Sustainable Integration. Educational Administration: Theory and Practice, 30(4), 7242–7254. https://www.kuey.net/index.php/kuey/article/view/2552
20. Adyasha Dash, Subhashree Darshana, Devendra Kumar Yadav, Vinti Gupta, A clinical named entity recognition model using pretrained word embedding and deep neural networks, Decision Analytics Journal, Volume 10, 2024,100426.
21. Jena, J.J., Satapathy, S.C. A new adaptive tuned Social Group Optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems. Multimed Tools Appl 83, 3021–3055 (2024).
22. S. Sharma, K. Baishya, M. Pandey and S. S. Rautaray, "Hybrid Product Recommendation System using Popularity Based and Content-Based Filtering," 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), Chennai, India, 2023, pp. 1-8
23. S. S. Rautaray, S. Nayak and M. Pandey, "A Machine Learning Based Employee Mental Health Analysis Model," 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India, 2023, pp. 1055-1059