[1]M. Jeyasudha

[2]Dr. S. Prakash

Integrated Sensor Fusion and Multi-Modal Hierarchical Neural Network for Activity Recognition in Lower Limb Prosthetics

Journal of Electrical Systems

Abstract: - Activity recognition plays pivotal role in enhancing functionality for prosthetic devices, ensuring seamless integration with users' movements. However, the complexity arises from the diverse data sources, including acceleration, angular velocity, joint angles, orientation, electromyography (EMG), and marker data, necessitating a robust approach to overcome challenges in information integration. The primary challenge lies in the effective utilization of multiple sensor modalities, each with unique characteristics and potential noise sources. The proposed solution addresses this by employing advanced sensor fusion techniques, such as Kalman filtering, during data collection. Synchronization and resampling ensure temporal consistency, while noise reduction techniques, such as low-pass filters, mitigate signal distortions. To further refine the process, a hybrid optimization-based feature selection Adaptive Step Size in Marine Predators Algorithm (ASSMPA) is introduced, focusing on marker data features. ASSMPA synergizes Marine Predators Algorithm (MPA) and Pathfinder Algorithm (PFA) for optimal feature selection in marine predator pathfinding tasks. The Feature Fusion step integrates attention mechanisms to dynamically weigh the significance of different sensor modalities during the fusion process. This strategic fusion enhances the overall performance of the Multi-Modal Hierarchical Neural Network (MMHNN). The proposed model is implemented using Python.

Keywords: Lower Limb Prosthetics; Kalman Filtering; ASSMPA; MPA; PFA; MMHNN.

I. INTRODUCTION

In the area of lower limb prosthetics, the quest of seamlessly blending artificial limbs with the natural rhythm of human movement has fuelled a wave of innovation. The intricate dance between man and machine requires a profound understanding of user intent, and at the heart of this understanding lies the pivotal concept of activity recognition [1,2]. This paper delves into the transformative landscape of activity recognition in lower limb prosthetics, where technological strides intersect with the human experience, reshaping the possibilities for those seeking enhanced mobility [3]. The rising incidence of lower limb amputations, attributed to diverse factors such as chronic diseases and accidents, has underscored the need for prosthetic solutions that go beyond mere functionality [4,5]. Restoring a sense of natural, intuitive movement has become a paramount goal, pushing researchers to explore novel avenues in the field of activity recognition [6]. Unlike traditional prosthetics, which often operate in a passive mode, contemporary lower limb prosthetics are undergoing a paradigm shift by embracing active control mechanisms [7,8]. This shift not only marks a departure from convention but holds the promise of empowering users with unprecedented control over their prosthetic limbs.

At the core of this evolution is the human-machine interface, a complex interplay where user intention meets technological prowess [9,10]. Traditional methods of activity recognition, relying heavily on surface electromyographic signals (sEMGs), have encountered challenges that necessitate a re-evaluation of approaches [11]. The journey from sEMGs to alternative mechanical sensors, including accelerometers, gyroscopes, and pressure sensors, reflects the ongoing endeavour to enhance the reliability and accuracy of activity recognition systems [12,13]. The significance of this pursuit extends beyond the technical intricacies of signal processing and sensor fusion; it encapsulates the essence of restoring not just mobility but a holistic sense of autonomy and participation in daily life. Individuals with lower limb amputations aspire not merely to move but to move seamlessly, adapting to diverse terrains, adjusting gait parameters, and effortlessly transitioning between various activities [14]. This aspiration propels the research community to explore innovative avenues that decode the intricacies of human motion intention. While challenges persist, including the nuances of muscle fatigue and the variability in contraction force, recent strides in prosthetic technology underscore the feasibility of achieving real-time and intuitive control [15]. The fusion of pattern recognition techniques with advanced sensor technologies opens new frontiers, paving the way for prosthetics that respond dynamically to the user's environment and intent.

The primary contribution of this research work is:

[1] Assistant professor, Bharath institute of higher education and research, BIHER, Chennai -600072, India. jeyasudhamurali03@gmail.com

[2]Professor and dean, Bharath institute of higher education and research, BIHER, Chennai -600072, India. prakash.eee@bharathuniv.ac.in

- To implement hybrid optimization-based feature selection, the process includes extraction of coordinate features with marker data and incorporation of a hybrid optimization algorithm, ASSMPA.
- To dynamically weigh the importance of different sensor modalities, attention mechanisms are introduced in the feature fusion process. This step allowed for dynamic weighting at each time step, enhancing the adaptability of the system.
- To advance active lower limb prosthetics recognition, MMHNN is introduced. This network integrated an Long Short-Term Memory (LSTM) model, Incremental Dilations CNN model, attention-based model, and a dense neural network in a hierarchical structure.
- To optimize information fusion, attention mechanisms were implemented at the LSTM and Incremental Dilations CNN layers. These mechanisms dynamically weighed the importance of information, facilitating effective feature fusion based on attention weights.

The following is the arrangement of the remaining sections: The research and literature reviews that are relevant to the topic are covered in Section 2. The suggested framework is talked about in Section 3. Section 4 describes the outcomes that were recorded while Section 5 concludes this study.

II. LITERATURE REVIEW

In 2020, Li *et al.* [16] proposed a unique approach for identifying bicycling stages in lower limb amputees utilizing a Support Vector Machine (SVM) enhanced by Particle Swarm Optimization (PSO). Wireless accelerometers then knee joint angle sensor in prosthesis collect information, filtered with a soft-hard threshold to reduce noise. This method enhances prosthetic knee joint control during bicycling.

In 2019, Wang *et al.* [17] used a novel time–frequency feature extraction method, Microelectromechanical systems (MEMS) inertial measurement unit (IMU) captures periodic inertial data on the prosthesis. Fractional Fourier transform (FRFT) is applied for feature extraction, constructing an eight-feature vector. Experimental results show improved classification efficiency and reduced hardware computation requirements.

In 2022, Pergolini *et al.* [18] introduce an innovative locomotion realization algorithm for Active Pelvis Orthosis, aiming for enhance the walking patterns of lower-limb amputees. The algorithm, tested in real-time circumstances with four transfemoral amputees, combines rule-based and linear discriminant analysis classifiers to accurately identify quasi-static and dynamic locomotion modes.

In 2022, Lv *et al.* [19] focused on knee joint trajectory planning for lower limb prostheses, introducing a novel approach through experimental data mining. Coordination indexes, including mean absolute relative phase (MARP) and deviation phase, reveal a steady stage variance between hip and knee motions. A Motion-Lagged Coordination Mapping (MLCM) is established, utilizing polynomial model for efficient mapping from lagged hip to knee motion. The MLCM offers simplified and robust knee trajectory generation for prosthesis control based solely on healthy limb hip motion.

In 2020, Kim *et al.* [20] investigated clinical feasibility for utilizing wearable sensors to evaluate functional mobility for individuals by lower limb amputation. Involving 17 amputees and 14 non-amputee controls, the research uses IMU and global positioning system (GPS) data about two weeks to assess cadence, walking speed, and stride lengths in various settings.

In 2022, Nordin *et al.* [21] used a Magnetorheological Fluid (MRF) damper for reducing ground reaction force impact during heel strikes. Emphasizing an energy-efficient design for prolonged battery life, the research investigates different MRF properties and magnetic particle volume contents. Results demonstrate that a higher solid weight percentage in MRF yields more significant damping force with reduced applied current, supporting execution for energy-efficient MRF damper in prosthetic limbs using Fuzzy-PID controller.

In 2023, Hu *et al.* [22] investigated lower-limb cooperation in entities by unilateral transfemoral amputation (UTFA) during walking, employing continuous relative phase (CRP) analysis. Fourteen UTFA participants and age-matched non-invalid persons contributed. Coordination patterns revealed compensatory strategies, with distinctive traits thigh-shank coupling over stance and swing stages. Shank-foot coupling indicated compensatory foot-leading patterns in intact limbs during mid-stance to address prosthetic limb force generation limitations.

In 2023, Mishra *et al.* [23] addressed the challenge of estimating corrective torque needed for postural balance in unilateral lower limb amputees (LLAs) over weight-shifting trainings. Using an inverted pendulum (IP) model based on healthy individuals' dynamics, a Relative Integral Derivative controller is planned to evaluate corrective torque. The proposed control scheme is validated through simulations, aiding in understanding and addressing postural stability issues in LLAs and contributing to the development of cost-effective prosthetic solutions for rehabilitation in developing countries.

In 2021, Stolyarov *et al.* [24] presented progress and offline confirmation for heuristic algorithm of exact forecast for ground territory in lower limb prothesis. Utilizing a single inertial measurement unit, the heuristic method, based on prosthetic limb kinematics, achieved a low complete prediction fault of 2.8%, outperforming machine learning approaches.

In 2022, Vijayvargiya *et al.* [25] developed hybrid deep learning models for this purpose, incorporating discrete wavelet transform for noise suppression and employing convolutional neural networks of sequential learning, alongside long short-term memory or gated recurrent units of ordered learning.

2.1. Problem Statement

The increasing aging population worldwide has highlighted the need for effective fall detection systems to mitigate the severe health consequences associated with falls among the elderly. Existing systems often exhibit limitations regarding accuracy, adaptability, then robustness. Challenges arise from complexity for capturing nuanced human movements, variations in environmental conditions, and the dynamic nature of daily activities [5,11]. Conventional methods often struggle to discern between normal activities and genuine fall events, leading to false alarms or missed detections [20]. Additionally, there is a need for comprehensive systems that integrate advanced techniques for data augmentation, feature extraction, and optimized feature selection. The lack of a unified approach incorporating spatial and temporal information delayed the development of a reliable and efficient fall detection system. Addressing these challenges is crucial for the successful implementation of a fall detection solution that ensures timely intervention, thereby significantly impacting the overall well-being of the elderly population.

III. PROPOSED METHODOLOGY

The integration of sensor fusion and a multi-modal hierarchical neural network in lower limb prosthetics enhances activity recognition. Implementing integrated sensor fusion and multi-modal hierarchical neural networks for lower limb prosthetics faces challenges such as computational complexity, data misalignment, and the requirement for substantial training data, impacting the balance between accuracy and real-time processing. Sensor fusion combines data from diverse sensors like accelerometers and myoelectric sensors, providing a complete view of user movements. The multi-modal hierarchical neural network processes this fused data, allowing the prosthetic device to recognize and adapt to various activities, improving functionality and user experience. Fig. 1 depicts the overall proposed architecture.

3.1. Data Collection

The dataset comprises gait analysis data from 14 Syrian above-knee amputees and 20 healthy subjects. It includes diverse files detailing subject information, spatio-temporal limits, lower limb joint angles, ground reaction force components, and lower limb joint moments. The data were collected using an optoelectronic system equipped with six cameras and two force platforms, with a sampling rate of 200Hz. The analysis followed a three-step process involving subject preparation, raw data acquisition during standing and walking trials, and subsequent calculation of gait parameters. The dataset, published on May 25, 2021, is associated with a DOI (Digital Object Identifier) and is available under Creative Commons Attribution 4.0 International License. The research was carried out[RP] at Damascus University and contributes valuable insights to the fields of biomedical engineering, gait analysis, amputation of lower limbs, and prosthetics.

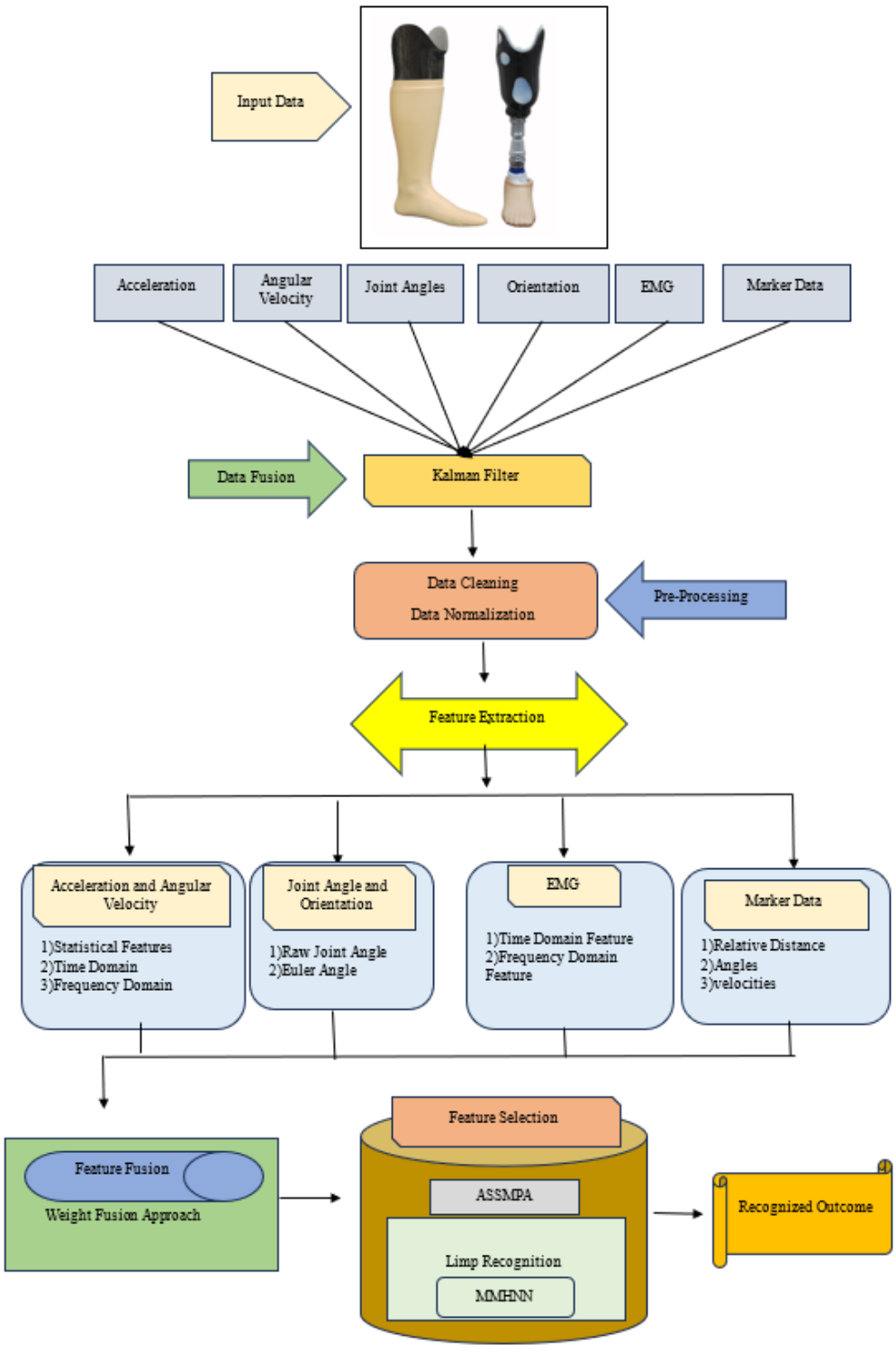

Figure 1: Overall Proposed Architecture

3.1.1. Kalman Filtering[RP]

In this study, advanced sensor fusion techniques, specifically Kalman filtering, were employed to seamlessly integrate data from multiple sensors. The dataset includes diverse data types: acceleration (Acc), angular velocity (Gyr), joint angles (Ang), orientation (Ori), electromyography (EMG), and marker data (Mrk). In medical imaging, the presence of Salt and Pepper noise significantly impacts the performance of detection and classification methods. Consequently, this issue is effectively addressed through the implementation of a highly efficient Kalman filtering technique. Kalman filter, a recursive algorithm used for estimating and predicting the state of a dynamic system observed through noisy measurements. This adaptability allows the filter to better account for changing conditions and uncertainties, resulting in improved accuracy and performance. The primary goal of Kalman filtering is to

update the filter's estimates and covariance matrices in real-time, ensuring that the filter remains well-calibrated to the evolving system dynamics. This is especially valuable when the system's characteristics vary over time or are not fully known. The Kalman filter involves two main steps in each iteration as given in Eq. (1) to Eq. (5).

1. Prediction step

Predicted state estimate: $\hat{x}_{k|k-1} = F_k\hat{x}_{k-1|k-1} + B_k u_k$ (1)

Predicted error covariance: $P_{k|k-1} = F_k P_{k-1|k-1} F_k^t + Q_k$ (2)

2. Update step

Kalman gain: $K_k = P_{k|k-1} H_k^t (H_k P_{k|k-1} H_k^t + R_k)^{-1}$ (3)

Corrected state estimate: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k\hat{x}_{k|k-1})$ (4)

Corrected error covariance: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$ (5)

Kalman filtering distinguishes itself through the dynamic adjustment of matrices Q_k (representing process noise covariance) and R_k (representing measurement noise covariance) based on the alignment between the filter's predictions and actual measurements. This adaptability is achieved through techniques like recursive least squares, covariance matching, or gradient-based approaches. This flexibility in parameter adaptation empowers the filter to sustain accurate state estimates even in the face of changing system behavior or uncertainties. Continuous parameter adjustments enable adaptive Kalman filtering to excel in tracking and estimating the evolving state of dynamic systems, making it a versatile asset across a range of applying.

3.2. Pre-Processing

In this study, the preprocessing of data involves essential steps such as data cleaning to address inconsistencies, data normalization for standardization, and handling null values to ensure completeness.

3.2.1. Data Cleaning

Data cleaning is a essential phase in preparing a dataset for analysis, in the activity recognition in lower limb prosthetics. This process involves systematic identification and rectification of inconsistencies, errors, and missing values to ensure dataset reliability and accuracy. By scrutinizing each data point for outliers and cross-referencing against medical guidelines, discrepancies are addressed, enhancing the dataset's integrity. Missing values are handled through imputation techniques like mean or regression imputation, maintaining data completeness. Standardizing data formats and ensuring feature consistency eliminate disparities in units or scales, fostering accurate analysis. Thorough data cleaning establishes a solid foundation for subsequent modeling and analysis, enabling the development of effective risk assessment tools and interventions.

3.2.2. Data Normalization

Data normalization is the important preprocessing methodology employed to manipulate and regulate the numerical attributes of a given dataset. The principal impartial is to equalize all the attributes to a comparable scale, thereby averting the ascendency of certain attributes over others when training the model.

Several forms of normalization methods are presented. we have compared three normalization methods namely Min-Max, Z-Score, and Decimal Scaling normalization.

1) Min-Max Normalization:

The unique data is revised progressively through min-max normalization. Inside the specified choice, the values are normalized. The calculation is as follows for translating w value for an trait B from a range $[Min_B, Max_B]$ toward new range $[New\_Min_B, New\_Max_B]$ and present as per the Eq. (6)

$$\frac{w - Min_B}{Max_B - Min_B}(New\_Max_B - New\_Min_B)New\_Min_B \quad (6)$$

where w' is updated value inside a necessary range. Min-Max normalization has benefit for annealing entire values within definite range.

2) Z-Score Normalization:

Z-score normalization, alternatively referred to as standardization, represents an essential data preprocessing method employed to rescale and standardize attributes through their conversion into a standardized normal distribution and present in the Eq. (7)

$$h' = \frac{h - mean(n)}{std(n)} \quad (7)$$

Where $Mean(n)$= sum of the all-attribute values of n

$std(n)$=Standard deviation of all values of n

3) Decimal Scaling normalization

Decimal scaling normalization is a method for preprocessing data that encompasses the manipulation of feature values through shifting and scaling, intending to confine them within a predetermined range denoted by decimals.

$$h' = \frac{h}{10^s} \quad (8)$$

Where, "s" is the smallest integer that max ($h' < 1$)

3.2.3. Null Value Handling

A null value indicates that there is either little or no information available. Null values are independent of zero or empty values. Null value handling is a crucial aspect of data preprocessing in various fields, including data science and analytics. When dealing with datasets, null or missing values can impact analysis and model performance. Several strategies exist for handling null values, such as removal, imputation, or interpolation. Removal involves eliminating rows or columns with null values, but it may lead to data loss. The choice of strategy depends on the dataset's characteristics and the desired outcome, with careful consideration required to maintain data integrity and enhance the reliability of subsequent analyses. There exist multiple methodologies to address this quandary; disregarding entities that possess null values, physically bridging the gaps, replacing the absent values by the ongoing variable, employing mean for entities, also utilizing maximum likely value to rectify absence, however, such techniques are not entirely efficacious in managing Null value predicaments.

3.3. Feature Extraction

3.3.1. Acceleration (Acc) and Angular Velocity (Gyr):

For Acceleration (Acc) and Angular Velocity (Gyr) data, various statistical, time-domain, and frequency-domain features are extracted to characterize signals.

3.3.1.1. Statistical Measures

- Mean - The mean is calculated by dividing the sum by the count. This is mathematically shown in Eq. (9).

$$Mean = \frac{Sum\ of\ all\ observations\ of\ R^{pre}}{Total\ number\ of\ observations\ in\ R^{pre}} \quad (9)$$

- Standard Deviation - Standard deviation (SD) measures data deviation from the mean. Low SD means values are close; high SD, they're spread. This is mathematically revealed in Eq. (10).

$$\text{SD } (\sigma) = \sqrt{\frac{\sum (R^{pre} - \mu)^2}{N}} \quad (10)$$

Where R^{pre} is the input value (pre-processed data, μ is mean and N represents number of total elements.

- Skewness - Skewness is statistical assess that describes asymmetry of distribution of values. It measures extent to which the values are shifted or tilted to one side or the other of the mean. A normal spreading has skewness for zero, meaning the values are symmetrical around the mean. A positive skewness implies rightward shift

else a long right tail, while negative skewness suggests a leftward shift or a long-left tail in the distribution. Skewness is used to describe the distribution of data and to identify any asymmetries in the data as shown in Eq. (11).

$$skewness = \frac{3(Mean-Median)}{StandardDeviation} \quad (11)$$

- Kurtosis - Kurtosis is statistical portion that describes shape of the allocation for the set of values. It measures the "peakedness" or "flatness" for the data compared to normal spreading. Kurtosis zero characterizes a normal distribution, positive kurtosis denotes an increased peaked distribution, negative kurtosis denotes a flatter distribution. Kurtosis is commonly used on financial and economic analysis to describe the distribution of returns from investments or financial assets. In these applications, kurtosis is used to assess the risk associated with the investment, as high kurtosis may indicate a higher level of tail risk, or the risk of extreme outcomes, such as large losses. This is mathematically shown in Eq. (12).

$$Kurtosis\ (high\ order) = \frac{4^{th}Moment}{4^{th}Moment^2} \quad (12)$$

3.3.1.2. Time domain features

- RMS - The Root Mean Square (RMS) is statistical measure which quantifies medium magnitude or amplitude for the set of values. It is particularly common in signal processing, where it provides a representative value for the overall magnitude of a varying signal. The RMS is calculated using Eq. (13).

$$X_{rms} = \sqrt{\frac{1}{n}x_i^2} \quad (13)$$

- Variance - Variance measures numerical variation, indicating how data points deviate from the mean and each other, as in Eq. (14).

$$Variance = \frac{\sum(R^{pre}-\mu)^2}{N} \quad (14)$$

- Signal Magnitude Area (SMA) - is a feature extraction method calculated by summing the absolute values of acceleration signals along x, y, and z axes, indicating overall signal fluctuation. SMA is calculated using Eq. (15).

$$sma = \frac{1}{n}\sum_{i=1}^{n}(|x(i)| + |y(i)| + |z(i)|) \quad (15)$$

SMA quantifies the fluctuation degree of the acceleration signal by summing the absolute values of acceleration along each axis within the specified window. A higher SMA value indicates more pronounced fluctuations in the signal, suggesting more dynamic or vigorous movement during that window.

3.3.1.3. FFT

FFT, which stands for Fast Fourier Transform, is an important mathematical algorithm used to convert an indication of its time domain depiction for its frequency area representation. In other words, it analyses a signal and decomposes it into its individual frequency components. the Fourier Transform provides frequency information about the image. By analysing the magnitude of FFT coefficients, heat maps can be generated to visualize the distribution of frequencies. High magnitudes in the heat map suggest the presence of significant periodic patterns or structures in the image. The main idea behind FFT is to efficiently compute the Discrete Fourier Transform (DFT) for signal. The DFT takes discrete indication as input and produces a complex-valued sequence as output, representing the amplitudes and phases of different frequencies present in the signal. The formula for the DFT of a signal $x[n]$, where n is the index of the sample, is given as per Eq. (16),

$$X[k] = (x[n] * e^{-j*2\pi*k*n/N}) \quad (16)$$

Where: $X[k]$ is the complex-valued DFT coefficient at frequency k, N is entire number for models in indication, j is imaginary unit, $e^{-j*2\pi*k*n/N}$ is the complex exponential term. Frequency-Domain Features are extracted by applying FFT to a signal. This process transforms sign from its time-domain depiction to its frequency-domain portrayal, revealing various frequency components. Some key frequency-domain features include:

Dominant Frequency: The frequency with the highest magnitude in the FFT spectrum, indicating the primary oscillation in the signal.

Spectral Entropy: A measure of the distribution of frequencies in the signal. Higher entropy suggests a more evenly distributed spectrum.

Power Spectral Density (PSD): Represents power dispersal across various frequencies in the signal, offering insights into the signal's energy distribution.

3.3.2. Joint Angles (Ang) and Orientation (Ori)

3.3.2.1. Raw joint angles

It serves as features representing the articulated positions of body joints. These angles directly capture the configuration of limbs and body segments. Optionally, computing angular differences between consecutive frames involves calculating the change in joint angles over time. This additional step provides dynamic information, reflecting how joints are moving between frames. Both raw joint angles and angular differences contribute to a comprehensive understanding of biomechanics, aiding in tasks such as human motion analysis. These features are valuable in applications ranging from sports biomechanics to rehabilitation, offering insights into movement patterns and facilitating the development of predictive models.

3.3.2.2. Euler Angles

Euler angles, denoted as (roll, pitch, and yaw), are a representation of positioning for an object in three-dimensional space. The Euler angles describe the rotation needed to transform from a mentioned organize system for object's manage system.

Roll (ϕ): Rotation about the X-axis.

Pitch (θ): Rotation about the Y-axis.

Yaw (ψ): Rotation about the Z-axis.

The alteration matrix R from object's coordinate system for reference organize system is given by the product of individual rotation matrices is given as per Eq. (17).

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \tag{17}$$

Where $R_x(\phi)$ is rotation matrix around the X-axis by angle ϕ, $R_y(\theta)$ is rotation matrix around the Y-axis by angle θ, and $R_z(\psi)$ is rotation matrix around the Z-axis by angle ψ. Extracting Euler angles involves decomposing the given rotation matrix R into its roll, pitch, and yaw components. This decomposition process may involve solving for the angles using trigonometric functions.

3.3.3. Electromyography (EMG)

3.3.3.1. Time-Domain Features

Time-domain features like MAV, waveform length, and RMS quantify signal characteristics, aiding in signal analysis and processing.

- MAV - MAV is a measure of average complete amplitude for the signal. It is calculated by getting the average for complete values of the signal samples.

$$MAV = \frac{1}{n}\sum_{i=1}^{n}|x(i)| \tag{18}$$

Where n is number of samples in indication also $x(i)$ represents $i-th$ sample.

- Waveform Length - Waveform length is the measure for cumulative length of signal. It quantifies the overall extent of the signal's waveform. For a discrete signal $x[i]$, the waveform length is calculated as per Eq. (19)

$$wl = \sum_{i=1}^{n-1}|x(i+1) - x(i)| \tag{19}$$

3.3.3.2. Frequency-Domain Features

- Median Frequency - The median frequency measure that splits the power spectrum into two equal halves. It is the frequency below which half of the total power is contained. Mathematically, it can be computed by finding the frequency at which half of the cumulative power is reached. For a discrete power spectrum $P(j)$, the median frequency is determined as per Eq. (20).

$$\sum_{j=1}^{mf} p(j) = \frac{1}{2}\sum_{j=mf}^{n} p(j) \quad (20)$$

- Spectral Entropy - Spectral entropy is a measure of the complexity or randomness of the frequency distribution in a signal. It is calculated using the Shannon entropy formula. For a discrete power spectrum $P(m)$, the spectral entropy is given by Eq. (21) and it quantifies the information content or uncertainty associated with the power spectrum. The Fourier Transform is implemented to the signal to change it from the time domain to frequency domain, providing the power spectrum.

$$se = -\sum_{m=1}^{n} p(m)\log_2 p(m) \quad (21)$$

3.3.4. Marker Data (Mrk)

Coordinate features extracted from marker data provide valuable information about the spatial relationships and dynamics of points in a motion capture system.

- Relative Distances: Calculate the distances between pairs of markers to capture the relative spatial positions. The Euclidean distance formula is commonly used, expressed as per Eq. (22).

$$rd = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (22)$$

- Angles: Determine the angles formed by three markers to understand joint configurations or segment orientations. Angle calculation can be based on trigonometric functions, dot products, or other geometric methods.
- Velocities: Compute the velocities of markers to capture the speed of movement. Velocity is often obtained by differentiating the coordinates with respect to time. For marker i, the velocity (v_i) can be calculated as per Eq. (23).

$$v_i = \sqrt{\left(\frac{dx_i}{dt}\right)^2 + \left(\frac{dy_i}{dt}\right)^2 + \left(\frac{dz_i}{dt}\right)^2} \quad (23)$$

where $x_i,\ y_i,\ z_i$ are the coordinates of marker i, and t is time.

3.4. Feature Fusion

Feature fusion, the amalgamation of multiple feature vectors into a comprehensive one, typically relies on vector concatenation. However, direct concatenation can lead to shading issues due to varying value ranges from different extractors, impacting smaller values. To address this, normalization is performed before fusion, mitigating numerical shading concerns. Diverse dimensions and units in machine learning features necessitate data standardization for compatibility. Integrating attention mechanisms in the fusion process enables dynamic weighting of the importance of various sensor modalities at each time step. This adaptive approach permits system to focus optionally on pertinent details, enhancing overall performance and responsiveness of the fusion process in scenarios where certain modalities or features may be more critical during specific activities or phases. In this paper, Z-score normalization technique employed, examining feature similarities and calculated as per Eq. (24). Features are then weighted to optimize fusion, recognizing varied contributions from different approaches and ensuring an effective and balanced feature fusion strategy.

$$ff' = \frac{ff-\mu}{\sigma},\ \mu = \sum_{i=1}^{n_{ff}} ff_i,\ \ \sigma = \sqrt{\frac{1}{n_{ff}}\sum_{i=1}^{n_{ff}}(ff_i - \mu)^2} \quad (24)$$

Utilizing the mean (μ) and standard deviation (σ), features are normalized. The process culminates in vector concatenation for fusion as represented by Eq. (25).

$$FF = \bigcup_{i \in \{1,\ldots,N\}} ff_i = [ff_1 \oplus ff_2 \oplus \ldots \oplus ff_n] \quad (25)$$

Where n represents the count of features for concatenation, denoted by $\oplus$. The feature set $\{ff_1 \oplus ff_2 \oplus \ldots \oplus ff_n\}$ in Eq. [25] encompasses various combinations, incorporating deep learning, color, shape, and texture-based features. Features are balanced by assigning weights based on their significance, using Eq. (26). This ensures a fusion, harmonizing contributions from different approaches, crucial for optimizing the overall performance of the machine learning model.

$$FF = \bigcup_{i \in \{1,\ldots,N\}} we_i ff_i = [we_1 ff_1 \oplus we_2 ff_2 \oplus \ldots \oplus we_n ff_n] \quad (26)$$

where the scale value we_i means the weight of feature ff_i.

3.5. Feature Selection

The planned algorithm, ASSMPA, fuses MPA and PFA to enhance feature selection efficiency. It leverages the strengths of both algorithms for improved performance in optimization tasks.

3.5.1. Adaptive Step Size in Marine Predators Algorithm (ASSMPA)

The new hybrid optimization algorithm ASSMPA combines MPA and PFA for optimal feature selection. PFA is a nature-inspired optimization technique where a leader guides a group of particles, mimicking animal herds' collective behavior. It aims to find optimal solutions by combining guided and random movements within a solution space. MPA is nature-inspired optimization method which simulates hunting behavior of marine predators. It employs principles of prey pursuit, encircling prey, and cooperative hunting to solve optimization problems by iteratively updating candidate solutions.

3.5.1.1. Mathematical Model

In the proposed model, swarm members are positioned in 2D, 3D, or d-dimensional spaces. The leader, known as the pathfinder, is chosen based on their location in the most promising area. Swarm individuals represent candidate solutions for a problem and can navigate in various dimensions to search for prey, feeding areas, or follow the pathfinder's guidance. The PFA dynamically adjusts the step size parameter for predator movement, leading to improved search efficiency and optimization within the MPA. This adaptive update helps prevent local optima and premature convergence. The best-positioned member is chosen to serve as the leader and it is calculated as per the proposed Eq. (27).

$$p(t + \Delta t) = p^0(t) \cdot u + x_i + x_{path} + \varepsilon + P \cdot CF \quad (27)$$

$$CF = \left(1 - \frac{iter}{max-iter}\right)^{\left(2\frac{iter}{max-iter}\right)} \quad (28)$$

Where t represents time, p is the position vector, u is the unit vector without any angle. x_i represents pairwise interactions with neighbors p_i and p_j. x_{path} denotes the global force, which depends on the global optimum or the position of the pathfinder. ε is a vector of vibration, $P = 0.5$ denotes the constant number, CF regarded as an adaptive parameter for governing the step size in controlling predator movements. Eq. (29) determines how pathfinder's position is updated,

$$p_{path}(t + \Delta t) = p_{path}(t) + \Delta p + F \quad (29)$$

Here, p_{path} denotes the pathfinder's position vector, Δp signifies the distance it travels between points, and F represents the fluctuation rate vector. The collective swarm movement model mentioned earlier isn't directly applicable to solve optimization problems. Modifications are essential for adaptability. The primary adjustments involve transforming Eq. (27) and Eq. (29) into Eq. (30) and Eq. (31) in our approach. The initial modification is as follows:

$$p_i^{c+1} = p_i^c + Ra1 \cdot \left(p_j^c - p_i^c\right) + Ra2 \cdot \left(p_{path}^c - p_i^c\right) + \varepsilon, \quad i \geq 2 \quad (30)$$

Where c representing the current iteration, p_i and p_j denoting the position vectors of the ith and jth members, $Ra1$ and $Ra2$ which are random vectors. $Ra1$ is equal to $\alpha Ra1$ and $Ra2$ as $\beta Ra2$ where $ra1$ and $ra2$ are uniformly generated random variables within the [0,1] range and introduce random movements. α signifies the coefficient for interaction, determining the extent of movement influenced by neighbouring individuals. β represents the

coefficient of attraction, dictating the random distance to maintain proximity to the leader. There are two significant scenarios: When α and β are close to 0, individuals move randomly without interaction, essentially behaving independently. When α and β are significantly high follower members tend to move far from the leader, hindering the discovery of promising solutions. Ideally, α and β should be around 1 for balanced movement. These values can be either constant or randomly selected within a specific range during iterations. Additionally, a vibration term, represented as ε is generated in each iteration using Eq. (30). The second modification is as follows per the proposed Eq. (31).

$$p_{path}^{c+1} = p_{path}^{c} + 2ra3 \cdot (p_{path}^{c} - p_{path}^{c+1}) + F + P \cdot CF \tag{31}$$

Thus, using Eq. (33), $ra3$ is a random vector that is created evenly throughout the range [0,1] for each iteration.

$$\varepsilon = \left(1 - \frac{c}{c_{max}}\right) rv_1 \cdot d_{ij}, \qquad d_{ij} = \|p_i - p_j\| \tag{32}$$

$$F = rv_2 \cdot e^{\frac{-2c}{c_{max}}} \tag{33}$$

Where, p_i signifies the position vector for ith follower, p_j denotes position vector for jth follower, c denotes the present iteration, c_{max} is maximum amount of repetitions allowed. α and β are random values generated within the range [1, 2]. They also play a role in the algorithm's calculations. d_{ij} signifies the distance between the ith follower and the jth follower, rv_1 and rv_2 are arbitrary vectors with values ranging from -1 to 1. These vectors are used as part of the algorithm's calculations. The termination condition of the PFA can be based on either the extreme number of cycles (iterations) or the maximum number for function evaluations. The algorithm stops when one of these conditions is met. The PFA is an algorithm that utilizes these variables and concepts to achieve its specific objectives, which may involve path finding or optimization tasks. This enhances data protection by providing a robust defense against unauthorized access and bolstering the security of sensitive information. The steps of ASSMPA structure of pseudocode are presented in Algorithm 1.

Algorithm 1: ASSMPA
Initialize the parameters
Initialize the population
Find the pathfinder
While $c <$ maximum number of iterations
Adjust step size for MPA in PFA algorithm
Calculate best positioned member given as per Eq. (27)
Update the pathfinder position given as per Eq. (29)
Calculate the current iteration given as per Eq. (30)
Calculate the pathfinder position vector given as per Eq. (31)
Calculate fluctuation rate vector and update as per Eq. (33)
Generate F and ε
End

3.6. Active Prosthetics: MMHNN Powered Limb Recognition

This study employs a MMHNN for active lower limb prosthetics activity recognition. It integrates diverse models, including LSTM, CNN, Attention-Based, and Dense Neural Network, showcasing a comprehensive approach to effectively capture and interpret complex lower limb movements. Figure 2 illustrates the conceptual design of the MMHNN-powered limb recognition system for analysis and understanding.

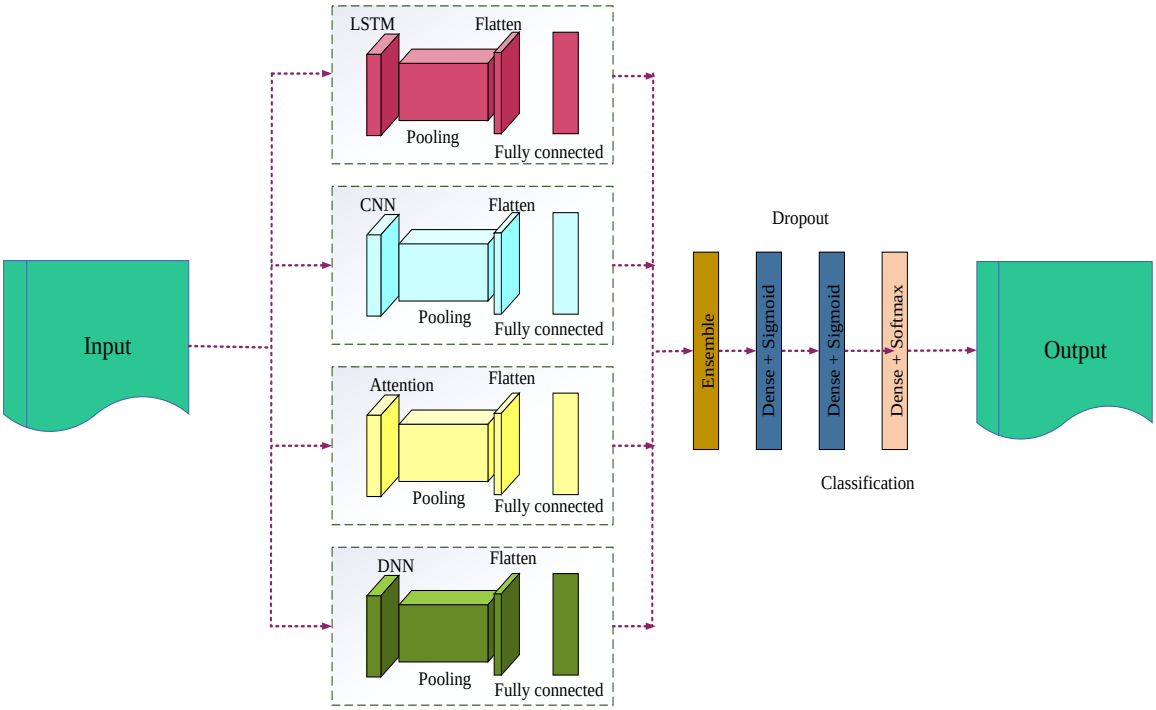

Figure 2: Proposed MMHNN Powered Limb Recognition

3.6.1. MMHNN

LSTM, a specialized RNN architecture, addresses vanishing/exploding gradient issues. It excels in processing time-series data by managing long-term dependencies. The LSTM cell, consisting of input, update, forget, and output gates, selectively stores, updates, forgets, and outputs information, maintaining a balance between long-term and short-term memory for effective sequential data processing. The input gate, which may be mathematically expressed per Eq. (34), determines which data must be supplied to the cell.

$$i_t = \sigma(w_i * [h_{t-1}, X_t] + b_i) \tag{34}$$

The vectors are multiplied element by element through the operator $*$. The forget gate, that is statistically defined as per Eq. (35), checks the details to be disregarded from prior retention.

$$f_t = \sigma\left(w_f * [h_{t-1}, X_t] + b_f\right) \tag{35}$$

The update gate, denoted hypothetically as per Eq. (36) and Eq. (37), adjusts cell state.

$$\tilde{c_t} = \tan h\ (w_C * [h_{t-1}, X_t] + b_c) \tag{36}$$

$$c_t = f_t * C_{t-1} + i_t * \tilde{c_t} \tag{37}$$

The output gate, which is also able to update output as it provided through prior time step, updates the hidden layer of that previous step in time as per Eq. (38) and Eq. (39).

$$O_t = \sigma(w_o * [h_{t-1}, X_t] + b_o) \tag{38}$$

$$h_t = O_t * \tan h(C_t) \tag{39}$$

The proposed model introduces Dilated Convolutional Neural Networks (CNN) with incremental dilations, enhancing feature extraction through the introduction of zeros between filter elements. Dilation, controlled by the hyper-parameter dilation rate, expands the receptive field of filters, allowing the network to capture more relevant information. Unlike traditional CNNs with large filters like 5×5, the model uses dilations in 3×3 convolution layers, avoiding additional computational overhead. The Dilated CNN architecture consists of three convolution layers with varying dilation rates, followed by Maxpooling layers and dense layers with ReLU activation. The model is designed for binary classification using the Sigmoid activation function. The incremental dilation approach aims to

analyze the impact on model performance, emphasizing simplicity to understand the gridding effect induced by dilation.

- **Convolutional layer**

The typical neural network's matrix multiplication process is replaced with the convolution operation in the convolutional layer, that is utilized to extract picture information as well as learn the mapping among the input and output layers. The network may learn only one collection of parameters through exchanging parameters through the convolution process, significantly lowering the total number of variables and significantly increasing computing efficiency. A convolution process is defined as in Eq. (40)

$$f_{j,g} = \sum_{i=o}^{h} \sum_{l=0}^{h} k_{i,l\, m_{j+i,g+l}} \qquad (40)$$

where $k_{i,l}$ is weight for convolutional kernel at $mandl;\ m_{j+i,}$ is the pixel value for image at i and $g;\ h$ is height and width of convolutional kernel.

- **Activation Function**

To evade threatened gradients and hasten training, CNN naturally usages Rectified Linear Unit (ReLU) activation functions. Eq. (41) describes ReLU's goal.

$$\mathrm{ReLU}(n) = \begin{cases} f\ f > 0 \\ 0\ f \leq 0 \end{cases} \qquad (41)$$

- **Pooling Layer**

The pooling layer groups the input across feature maps and helps lower the computational complexity of the entire network. One typical pooling layer is maximum pooling shown in Eq. (42):

$$MxPl(o_{ot}, k_{ot}) = \begin{cases} o_{ot} = floor\left(\frac{(o_g + 2q - p)}{h} + 1\right) \\ k_{ot} = floor\left(\frac{(e_g + 2q - p)}{h} + 1\right) \end{cases} \qquad (42)$$

where $floor(m)$ is function for round up number, o_{ot} is yield height for feature map, k_{ot} is output width for feature map, o_{ot} is input height for feature maps, k_g is input size of feature maps, q is padding for feature maps, p is kernel size for max pooling, h is the kernel stride for max pooling.

- **Fully Connected Layer**

A type of neural network layer known as fully connected layers, often referred to as A dense layer is one where all of the neurons are coupled to all of the neurons in each layer that are above and below it. There is a learnable weight assigned to each link between neurons that is changed throughout training to enhance the performance of the network. Fully connected layers are utilized to recognize non-linear patterns then correlations in input data. These layers are capable of capturing detailed feature interactions. The outputs of a layer that is fully linked and has M input neurons and N output neurons can be calculated as follows:

The output value o_j is calculated for each output neuron $j(1 \leq j \leq N)$ by adding the weighted inputs from all input neurons $i(1 \leq i \leq M)$ and using an activation function,

$$o_j = f(s_j) = f\left(\sum(w_{ij} * x_i) + b_j\right) \qquad (43)$$

In Eq. (43), o_j is output of jth neuron, $f()$ is instigation function used element-wise to weighted sum for inputs, s_j is weighted sum of inputs to jth neuron, w_{ij} is weight associated with connection between ith input neuron and jth output neuron, x_i is input value of ith neuron, b_j is bias term for jth neuron.

The proposed model integrates LSTM for time-series data processing and Dilated Convolutional Neural Networks (CNN) for effective feature extraction. The LSTM cell manages long-term dependencies with input, update, forget, and output gates, ensuring a balance between memory types. Dilated CNNs use incremental dilations to enhance feature extraction by expanding the receptive field. The convolutional layer replaces matrix multiplication, pooling reduces computational complexity, and fully connected layers capture detailed feature interactions. ReLU activation

avoids vanishing gradients. The attention mechanism dynamically weighs LSTM and Dilated CNN information, ensuring adaptive fusion based on attention weights for comprehensive feature utilization.

A Dense Neural Network (DNN), or Fully-connected Neural Network, is a foundational architecture in artificial neural networks. It comprises layers of nodes where every node is intricately linked to all nodes on preceding and succeeding layers. The term dense stems from the comprehensive interconnections, creating a web of relationships within the network. The initial layer is the input layer, through which data is introduced. Hidden layers, situated between input and output layers, derive their name from the typically inaccessible and abstract nature of their logical representations. The output layer produces the final result.

IV. RESULT AND DISCUSSION

4.1. Experimental Setup

The implementation of proposed model utilized PYTHON, while Dataset [26] provided the information for evaluation. This section matches effectiveness for proposed procedure against established methods such as CNN, Recurrent Neural Network (RNN), SVM, and Gated Recurrent Unit (GRU). The evaluation aims to highlight the effectiveness and superiority of the proposed activity recognition of lower limb prosthetics methodology over established methods.

4.3. Overall Performance Analysis of Metrics in Proposed and Existing Methods

- **Accuracy:**

To determine a test's accuracy, the results from all the cases are examined and the percentage of true positives and negatives are determined.

$$Accuracy = \frac{TN+TP}{Tp+FP+TN+FN} \quad (44)$$

- **Precision:**

Precision is referred to the proportion of true positives that were successfully classified.

$$Precision = \frac{TP}{TP+FP} \quad (45)$$

- **Recall:**

Recall shows the number of right hits and true positives that were recalled or identified.

$$Recall = \frac{TP}{TP+FN} \quad (46)$$

- **F_Score**

The F-Score number achieves a compromise between accurately identifying every data bit and making sure that every stage only contains single kind of data item.

$$F\_Score = \frac{2(precision.Recall)}{Precision+ Recall} \quad (47)$$

- **Specificity**

Specificity is referring to the likelihood that a student will pass, assuming that it is affirmative.

$$Specificity = \frac{TN}{TN+FP} \quad (48)$$

- **FDR**

FDR is the “False Discovery Rate”. It is statistical portion which quantifies ratio for falsely identified positive data points to the overall count of positively identified data points

$$FDR = \left(\frac{Fp}{(Fp+Tp)}\right) \quad (49)$$

- **FNR**

FNR denotes the "False-Negative Rate". It can be described the percent of misclassified instances which falsely labelled as negative, comparative to total number of positive instances.

$$FNR = \left(\frac{Fn}{(Fn+Tp)}\right) \quad (50)$$

- **FPR**

FPR denotes the "False-Positive Rate". It can be clear as proportion for incorrectly classified positive data to total number of negative data.

$$FPR = \left(\frac{Fp}{(Fp+Tn)}\right) \quad (51)$$

- **MCC**

MCC refers to "Matthew's Correlation Coefficient". It is a statistical measure used to assess the quality of binary classification models. It is computed based on four values Tp, Tn, Fp, and Fn.

$$MCC = \left(\frac{((Tp \times Tn)-(Fp \times Fn))}{\sqrt{((Tp+Fp)(Tp+Fn)(Tn+Fp)(Tn+Fn))}}\right) \quad (52)$$

- **NPV**

NPV refers to the "negative predictive value". It can be described as the probability of accurately classifying genuine non-hit variants as non-hits, given a specific threshold.

$$NPV = \left(\frac{Tn}{(Tn+Fn)}\right) \quad (53)$$

The table 1 grants the recital metrics for four separate patterns MMHNN, CNN, Bi-LSTM, and DNN—on a classification job. The MMHNN model exhibits an impressive level of accuracy, with a score of 98.59%. It also displays a high level of precision, with a score of 98.90%, and a recall score of 98.74%. Additionally, the model achieves a balanced performance, as indicated by its F-score of 98.40%. CNN achieves a high accuracy of 94.00%, demonstrating precise measurements of precision, recall, and F-score at 92.11%, 92.31%, and 92.77% correspondingly. Both Bi-LSTM and DNN demonstrate comparable accuracies, with Bi-LSTM achieving 95.24% accuracy and DNN achieving 95.78% accuracy. The Bi-LSTM model achieves accuracy, recall, and F-score values of 92.11%, 92.94%, and 92.86% respectively, whereas the DNN model exhibits values of 93.02%, 93.02%, and 94.23% respectively. In addition, measurements like as specificity, sensitivity, MCC, NPV, FPR, and FNR offer a thorough assessment of the subtle variations in performance for each model. Stakeholders may utilize this comprehensive analysis to make well-informed decisions based on the precise needs and trade-offs associated with the categorization process.

Table 1: Comparison analysis of the performed metrics of Training dataset 70/30

Model	MMHNN	CNN	Bi-LsTM	DNN
Accuracy	0.98592	0.94	0.95238	0.95775
Precision	0.98901	0.92105	0.92105	0.93023
Recall	0.98739	0.92308	0.92941	0.93023
F-Score	0.98404	0.92771	0.92857	0.94231
Specificity	0.9879	0.93478	0.93478	0.94231
Sensitivity	0.98684	0.92105	0.92683	0.92683
MCC	0.9879	0.9434	0.94118	0.94915
NPV	0.98846	0.94231	0.95082	0.95
FPR	0.03641	0.0741	0.0641	0.0811
FNR	0.04654	0.06523	0.0523	0.0913

Table 2 explains for the classification test, performance metrics for four models—MMHNN, CNN, Bi-LSTM, and DNN—were examined using an 80/20 training dataset split. MMHNN demonstrated a remarkable overall accuracy of 99.07%, along with excellent precision (99.01%), recall (98.86%), and F-score (98.49%) values, indicating a strong and well-rounded performance. The CNN model attained an accuracy of 95.00%, by precision score of 93.75% and recall score of 93.18%, resulting in a competitive F-score of 92.94%. The Bi-LSTM model demonstrated strong performance, achieving an accuracy of 95.89%. It also showed high precision, recall, and F-score values of 92.11%, 93.81%, and 94.23% respectively. The DNN achieved a superior accuracy of 96.30%, demonstrating high precision (93.62%) and recall (93.18%), resulting in an exceptional F-score of 94.34%. The models exhibited different degrees of specificity, sensitivity, MCC, NPV, FPR, and FNR, offering a detailed comprehension of their respective advantages and limitations. Stakeholders may utilize these extensive indicators to make well-informed judgments depending on the precise needs of the categorization assignment.

Table 2: Comparison analysis of the performed metrics of Training dataset 80/20

Model	MMHNN	CNN	Bi-LsTM	DNN
Accuracy	0.99071	0.95	0.9589	0.96296
Precision	0.9901	0.9375	0.92105	0.93617
Recall	0.98859	0.93182	0.93814	0.93182
F-Score	0.98485	0.92941	0.94231	0.9434
Specificity	0.988	0.93617	0.94643	0.94444
Sensitivity	0.98701	0.92683	0.92683	0.93478
MCC	0.98805	0.94915	0.95238	0.95082
NPV	0.98855	0.94915	0.95082	0.95161
FPR	0.03341	0.0641	0.0601	0.0711
FNR	0.03954	0.05523	0.0503	0.0813

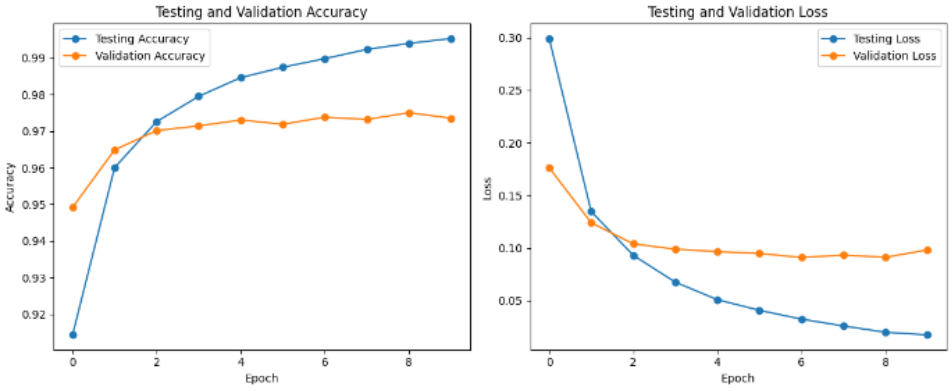

Figure 3: Testing and Validation for the Accuracy and Loss of Epoch

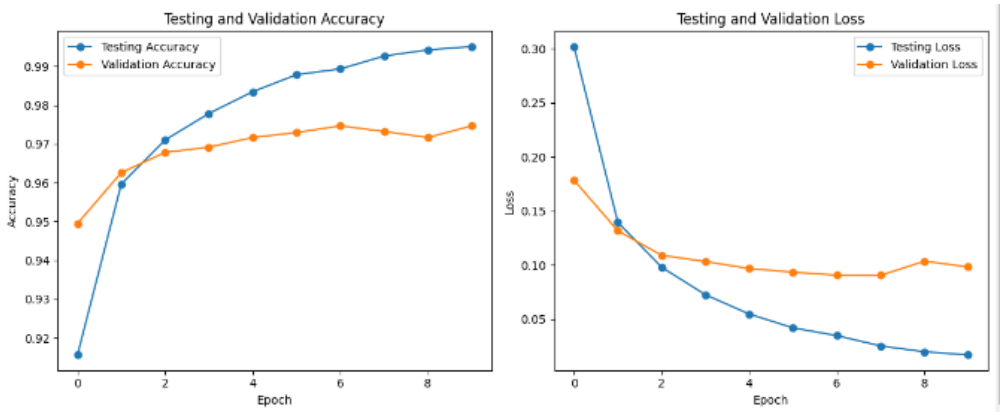

Figure 4: Testing and Validation for the Accuracy and Loss of Epoch

Figure 3 and 4 demonstrates the testing and validation's loss and accuracy for Epoch respectively.

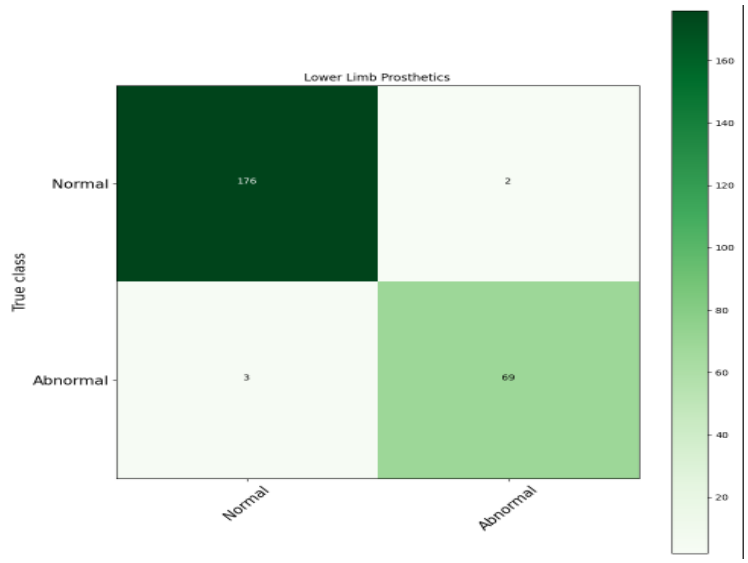

Figure 5: True class and Lower limb Prosthetics for normal and abnormal

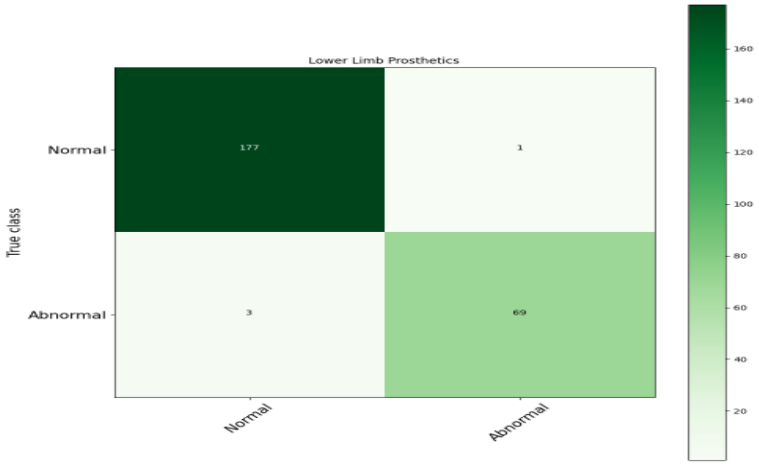

Figure 6: True class and Lower limb Prosthetics for normal and abnormal

Figure 5 and 6 explains the True class and Lower limb Prosthetics for normal and abnormal conditions respectively.

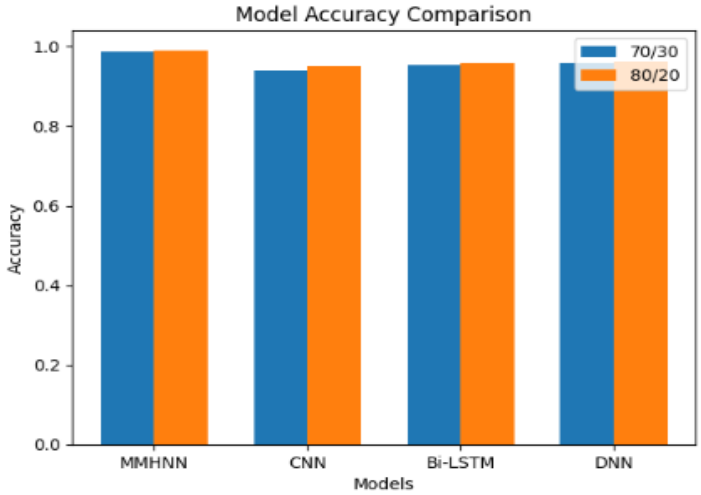

Figure 7: Graphical depiction for Accuracy comparison

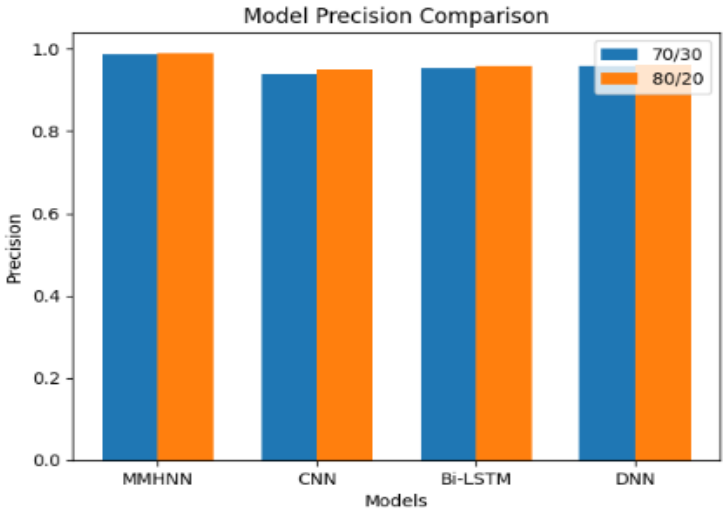

Figure 8: Graphical illustration for Precision comparison

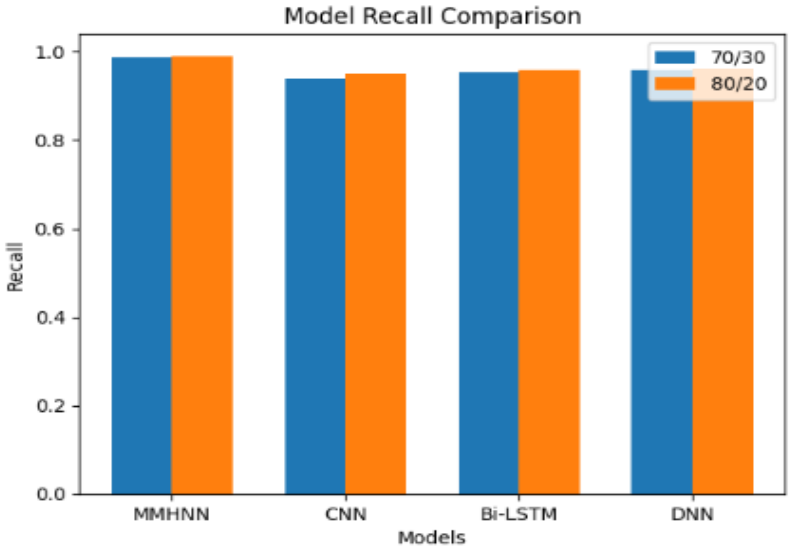

Figure 9: Graphical representation for Recall comparison

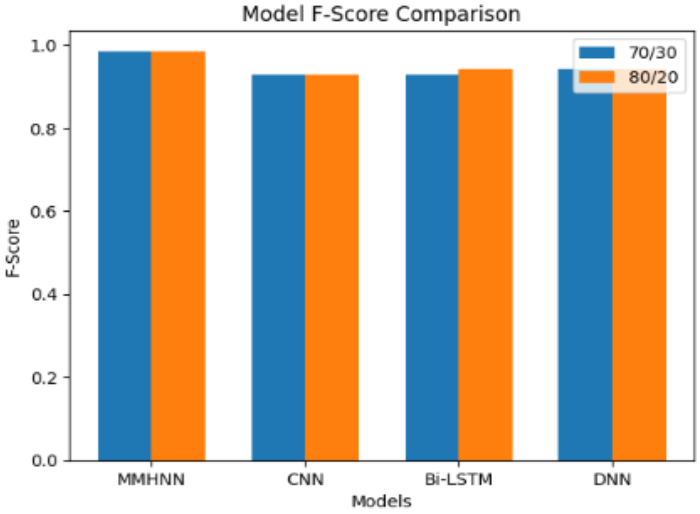

Figure 10: Graphical representation for F-score comparison

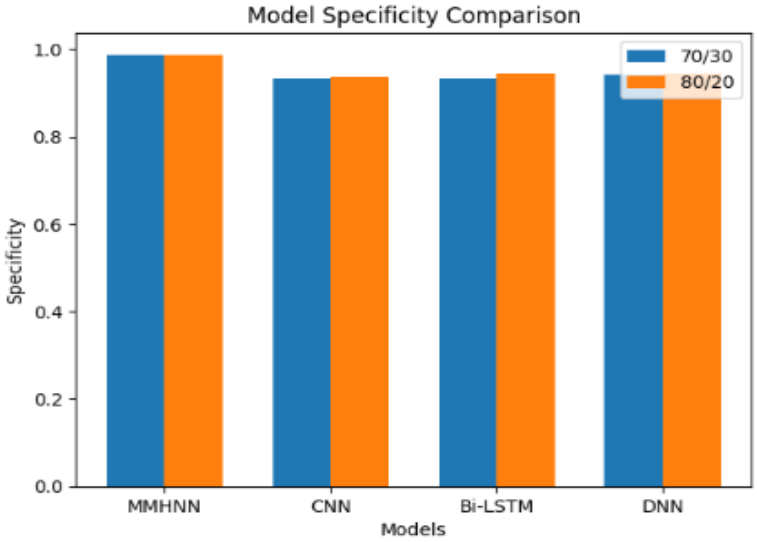

Figure 11: Graphical illustration for Specificity comparison

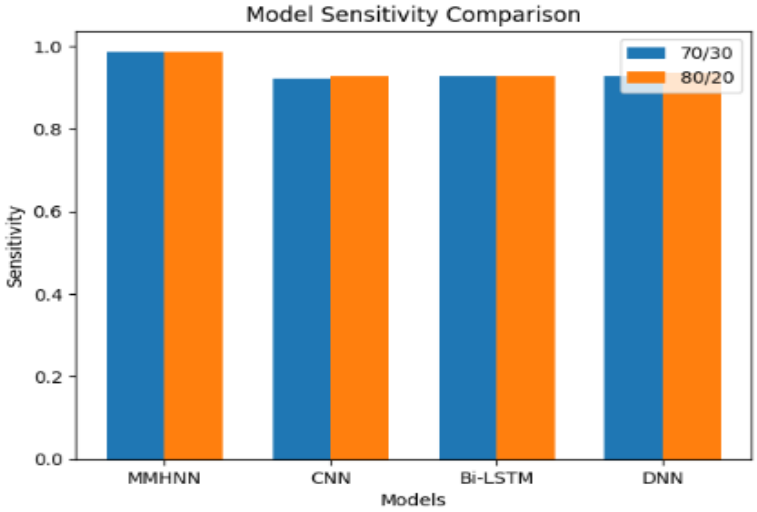

Figure 12: Graphical depiction for Sensitivity comparison

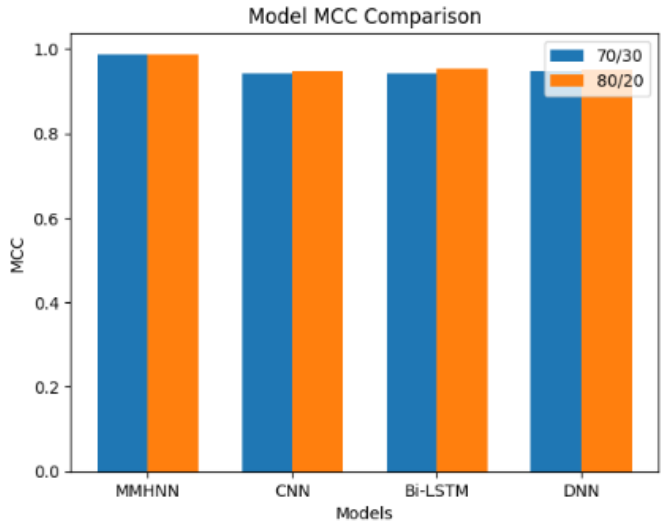

Figure 13: Graphical illustration for MCC comparison

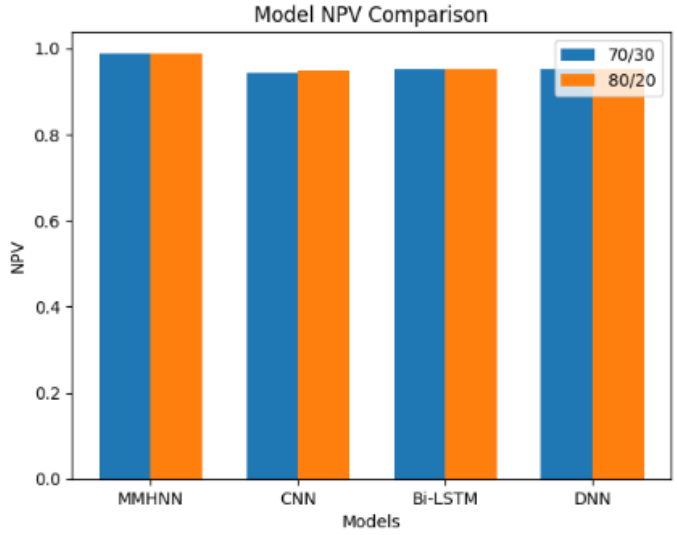

Figure 14: Graphical representation of NPV comparison

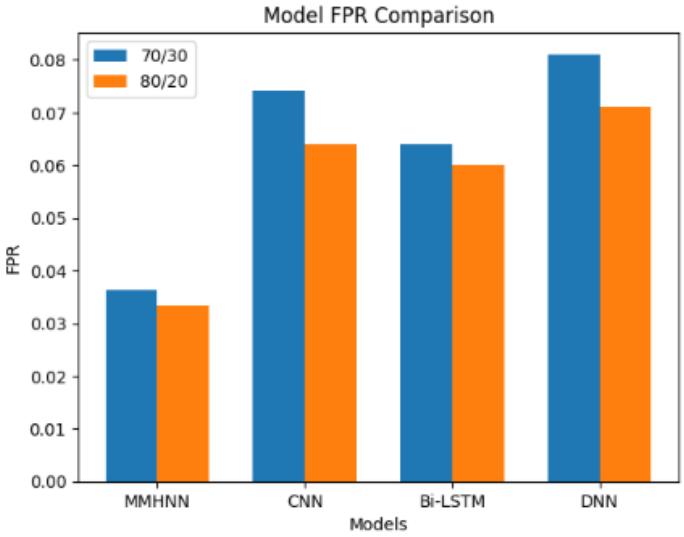

Figure 15: Graphical representation of FPR comparison

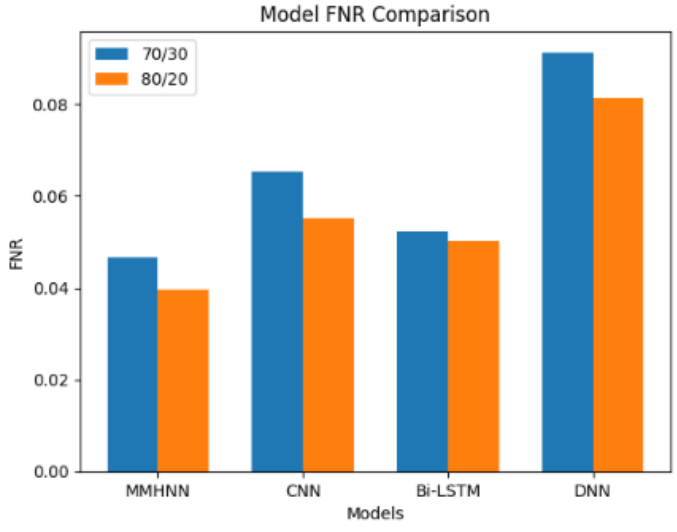

Figure 16: Graphical representation of FNR comparison

Figure 7 to 16 shows the comparative analysis of training dataset 70/30 and training dataset 80/20 respectively. The accomplishment metrics such as Accuracy, Precision, Recall, F-Score, Specificity, Sensitivity, MCC, NPV, FPR, and FNR were used for the comparison.

V. CONCLUSION

Integrating prosthetic devices with users' motions was made possible via activity recognition, which was crucial in improving their functionality. Nevertheless, the intricacy resulted from a variety of data sources, such as marker data, angular velocity, joint angles, acceleration, and orientation; therefore, a strong strategy was required to address information integration difficulties. Utilising various sensor modalities, each with distinct properties and possible sources of noise, effectively posed the main issue. Using sophisticated sensor fusion methods, like Kalman filtering, during data gathering, was how the suggested solution dealt with issue. While noise reduction methods like low-pass filters helped to lessen signal distortions, synchronisation and resampling made sure that data was consistent throughout time. The hybrid optimization-based feature selection technique known as ASSMPA, which focuses on marker data features, was established in order to further improve the procedure. In order to choose the best features for marine predator pathfinding tasks, ASSMPA combined MPA and PFA. In order to dynamically assess the importance of various sensor modalities throughout the fusion process, the feature fusion step integrated attention mechanisms. MMHNN performed better overall as a result of this strategic fusion. Python was used in the proposed model's implementation.

REFERENCES

[1] Slemenšek, J., Fister, I., Geršak, J., Bratina, B., van Midden, V.M., Pirtošek, Z. and Šafarič, R., 2023. Human gait activity recognition machine learning methods. Sensors, 23(2), p.745.

[2] Xu, G., Wan, Q., Deng, W., Guo, T. and Cheng, J., 2022. Smart-Sleeve: A wearable textile pressure sensor array for human activity recognition. Sensors, 22(5), p.1702.

[3] Li, C., Li, G., Jiang, G., Chen, D. and Liu, H., 2020. Surface EMG data aggregation processing for intelligent prosthetic action recognition. Neural Computing and Applications, 32, pp.16795-16806.

[4] Cao, T., Liu, D., Wang, Q., Bai, O. and Sun, J., 2020. Surface Electromyography-Based Action Recognition and Manipulator Control. applied sciences, 10(17), p.5823.

[5] Cui, J.W., Li, Z.G., Du, H., Yan, B.Y. and Lu, P.D., 2022. Recognition of Upper Limb Action Intention Based on IMU. Sensors, 22(5), p.1954.

[6] Hussain, T., Iqbal, N., Maqbool, H.F., Khan, M., Awad, M.I. and Dehghani-Sanij, A.A., 2020. Intent based recognition of walking and ramp activities for amputee using sEMG based lower limb prostheses. Biocybernetics and biomedical engineering, 40(3), pp.1110-1123.

[7] Vijayvargiya, A., Gupta, V., Kumar, R., Dey, N. and Tavares, J.M.R., 2021. A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition. IEEE Sensors Journal, 21(18), pp.20431-20439.

[8] Su, B.Y., Wang, J., Liu, S.Q., Sheng, M., Jiang, J. and Xiang, K., 2019. A CNN-based method for intent recognition using inertial measurement units and intelligent lower limb prosthesis. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(5), pp.1032-1042.

[9] Cheng, S., Bolívar-Nieto, E. and Gregg, R.D., 2021. Real-time activity recognition with instantaneous characteristic features of thigh kinematics. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 29, pp.1827-1837.

[10] Vijayvargiya, A., Khimraj, Kumar, R. and Dey, N., 2021. Voting-based 1D CNN model for human lower limb activity recognition using sEMG signal. Physical and Engineering Sciences in Medicine, 44, pp.1297-1309.

[11] Zhang, K., Xiong, C., Zhang, W., Liu, H., Lai, D., Rong, Y. and Fu, C., 2019. Environmental features recognition for lower limb prostheses toward predictive walking. IEEE transactions on neural systems and rehabilitation engineering, 27(3), pp.465-476.

[12] Iqbal, N., Khan, T., Khan, M., Hussain, T., Hameed, T. and Bukhari, S.A.C., 2021. Neuromechanical signal-based parallel and scalable model for lower limb movement recognition. IEEE Sensors Journal, 21(14), pp.16213-16221.

[13] Luan, Y., Shi, Y., Wu, W., Liu, Z., Chang, H. and Cheng, J., 2021. Har-semg: A dataset for human activity recognition on lower-limb semg. Knowledge and Information Systems, 63, pp.2791-2814.

[14] Peng, F., Zhang, C., Xu, B., Li, J., Wang, Z. and Su, H., 2020. Locomotion prediction for lower limb prostheses in complex environments via sEMG and inertial sensors. Complexity, 2020, pp.1-12.

[15] Wang, J., Cao, D., Wang, J. and Liu, C., 2021. Action recognition of lower limbs based on surface electromyography weighted feature method. Sensors, 21(18), p.6147.

[16] Li, X., Liu, Z., Gao, X. and Zhang, J., 2020. Bicycling Phase Recognition for Lower Limb Amputees Using Support Vector Machine Optimized by Particle Swarm Optimization. Sensors, 20(22), p.6533.

[17] Wang, T., Liu, N., Su, Z. and Li, C., 2019. A new time–frequency feature extraction method for action detection on artificial knee by fractional fourier transform. Micromachines, 10(5), p.333.

[18] Pergolini, A., Livolsi, C., Trigili, E., Chen, B., Giovacchini, F., Forner-Cordero, A., Crea, S. and Vitiello, N., 2022. Real-time locomotion recognition algorithm for an active pelvis orthosis to assist lower-limb amputees. IEEE Robotics and Automation Letters, 7(3), pp.7487-7494.

[19] Y. Lv, J. Xu, H. Fang, X. Zhang and Q. Wang, "Data-Mined Continuous Hip-Knee Coordination Mapping with Motion Lag for Lower-Limb Prosthesis Control," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 1557-1566, 2022, doi: 10.1109/TNSRE.2022.3179978.

[20] J. Kim, N. Colabianchi, J. Wensman and D. H. Gates, "Wearable Sensors Quantify Mobility in People with Lower Limb Amputation During Daily Life," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 6, pp. 1282-1291, June 2020, doi: 10.1109/TNSRE.2020.2990824.

[21] N. H. D. Nordin, A. G. A. Muthalif, M. K. M. Razali, A. Ali and A. M. Salem, "Development and Implementation of Energy-Efficient Magnetorheological Fluid Bypass Damper for Prosthetics Limbs Using a Fuzzy-Logic Controller," in IEEE Access, vol. 10, pp. 18978-18987, 2022, doi: 10.1109/ACCESS.2022.3149893.

[22] M. Hu, Y. He, G. Hisano, H. Hobara and T. Kobayashi, "Coordination of Lower Limb During Gait in Individuals with Unilateral Transfemoral Amputation," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 3835-3843, 2023, doi: 10.1109/TNSRE.2023.3316749.

[23] S. S. Mishra, A. T, B. K. Panigrahi and D. Joshi, "Neuromechanical Model-Based Corrective Torque Estimation During Weight Shifting in Lower Limb Amputees," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 4, pp. 2350-2366, Oct. 2023, doi: 10.1109/TASE.2022.3210569.

[24] R. Stolyarov, M. Carney and H. Herr, "Accurate Heuristic Terrain Prediction in Powered Lower-Limb Prostheses Using Onboard Sensors," in IEEE Transactions on Biomedical Engineering, vol. 68, no. 2, pp. 384-392, Feb. 2021, doi: 10.1109/TBME.2020.2994152.

[25] Vijayvargiya, A., Singh, B., Kumar, R., Desai, U. and Hemanth, J., 2022. Hybrid Deep Learning Approaches for sEMG Signal-Based Lower Limb Activity Recognition. Mathematical Problems in Engineering, 2022.

[26] Dataset collected from: “https://data.mendeley.com/datasets/k5y9jkx87y/1”, dated 29/11/2023.