AI-Based Classification of Handheld Object Weight Using Forearm IMU Data During Human Motion

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

Xianzhi Zhang

27th June 2025

AI-Based Classification of Handheld Object Weight Using Forearm IMU Data During Human Motion

Xianzhi Zhang
Supervisor: Arno Stienen

***Abstract*—Accurate classification of handheld object weight during different human motion is crucial for applications in health monitoring, injury prevention and exoskeleton systems. This study investigates the feasibility of using only a single forearm mounted inertial measurement unit (IMU) combined with AI algorithms to classify both movement types and the weights of handheld objects. A series of experiments with one subject were conducted to collect IMU data under various combinations of movements and object weights. Multiple feature extraction techniques including time, frequency, and time-frequency domains were applied, followed by classification using machine learning methods (SVM, KNN) and deep learning models (1D-CNN-LSTM, Wavelet-CNN). A genetic algorithm was used for optimal feature selection in machine learning pipelines, while open set classification capability was implemented using the Convolutional Prototype Network (CPN). Result shows that deep learning models, particularly 1D-CNN-LSTM method, outperform machine learning methods, achieving up to 94% classification accuracy. Moreover, the CPN model effectively rejected unknown movement patterns in open set scenarios. The proposed framework shows promising potential for wearable systems capable of intelligent workload classification in real world environments.**

***Index Terms*—Weight Classification, IMU, Open set Recognition, Machine Learning, Deep Learning**

I. Introduction

Strain injury is a common problem caused by repetitive modern physical loads, which seriously affects the health and work efficiency of the workforce[1]. With the rapid development of medical and health management, human-machine collaboration and intelligent assistance systems, the demand for intelligent systems that can monitor human physical load continues to grow. Accurately identifying the load level in physical tasks is important for applications such as preventing strain injuries and workload recorder.

Previous studies have used IMU sensor either alone or in combination with other sensors such as surface electromyography (sEMG) and force sensor to classify and predict hand gestures[2], muscle force[3], movement types and joint angles[4]. However, relatively little attention has been given to the estimation of handheld object weight during movement and the simultaneous classification of movement types by using IMU sensors. C Crema et al. used Linear Discriminant Analysis, a machine learning method, to classify different activities and achieved an average classification accuracy of 85%[5]. In contrast, Peter Hausberger et al. applied Hidden Markov Model, Support Vector Machines (SVM), and K-Nearest Neighbours (KNN) classifiers to classify different postures in weightlifting activities, achieving a segmentation misdetection rate of 1.5%, a classification accuracy of 99.7%, and an average response time of approximately 300 ms[6]. Martina Ravizza et al. use 6 IMUs and machine learning methods to perform classfication of resistive exercises and achieved an accuracy of 89.03%[7]. Most existing studies focus on classifying different types of movement, without jointly considering both the type of movement and the weight of the handheld object.

To fit this challenge, this study explores the potential of using only IMU sensor with AI techniques to perform classification of both object weight and movement type during dynamic human activities. The IMU is a compact, low-cost sensor capable of measuring linear acceleration and angular velocity, making it highly suitable for integration into wearable systems. Also, most smartwatches are already equipped with IMU sensors, allowing for cost effective implementation on existing platforms[8]. Compared to vision-based systems, force sensors, or sEMG based solutions, IMUs offer greater portability, lower deployment cost, and fewer environmental constraints, which makes them attractive for real world applications.

Accurately classification of handhold object weight and movement types based on acceleration signals has a significant challenge due to the subtle and highly individual dependent differences in arm dynamics under different loads. Acceleration signals, like many natural signals, are nonstationary and nonlinear. The statistical properties of a nonstationary signal such as mean and variance change over time, especially during human motion, where arm acceleration patterns can vary widely across different movement states. Even for the same action, acceleration intensity may fluctuate at different time points[9]. The acceleration signals of human motion are also nonlinear. Human motion involves complex and nonlinear interactions that arise from multiple biomechanical factors, including the coordinated movement of bones and muscles, the dynamic effects of ground reaction forces, and the interdependent behavior of multi joint systems. These intertwined mechanisms collectively contribute to the complexity of the acceleration signals, making them difficult to model and interpret using traditional linear approaches. This makes the generation process of acceleration signal cannot be simply fitted by linear model, especially in complex movements or multiple task state.This nature needs advanced signal engineering and robust modeling approaches to accurately extract and classify useful patterns from raw IMU data. Traditional threshold-based or rule-based approaches[10] often fail when applied to complex human motion, particularly when subtle variations in movement patterns are induced by different carried loads. In contrast, AI based methods especially those using neural networks can automatically learn discriminative features from time series sensor, enabling robust and generalizable classification of both object weight and movement type.

The hypothesis is that different weight loads and movement types have distinguishable motion signatures, which can be effectively captured by an IMU sensor and subsequently used for classification. By using proper feature engineering and algorithms, particularly machine learning and deep learning, these complex motion patterns can be learned from data, achieving robust classification over combination of weights and movement types.

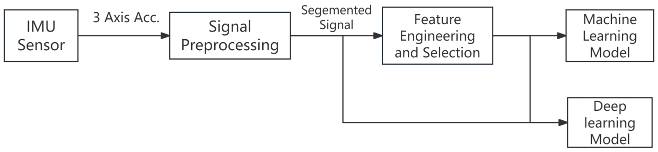

Fig. 1: Overall Process of Classification

The overall process for classifying different movement types and object weights of this study is illustrated in the figure1. After collecting raw acceleration data using an IMU sensor, the signal performed preprocessing steps such as denoising and normalization. Subsequently, the continuous signal is segmented into fixed length windows, each serving as a sample for further analysis. The segmented data obtained from raw acceleration signals can be directly used as input for some neural network based models or undergo feature extraction to use as input for both traditional machine learning algorithms and deep learning models. Feature engineering is a crucial part in Al methods, especially for mathematics-based machine learning methods which can not automatically learn features from raw data. Without informative and discriminative features, the performance of these models can degrade significantly. Although neural network based methods can automatically extract features, using well-designed features can still enhance model performance.

In this study, multiple types of features were used, including time-domain, frequency-domain, and time-frequency domain features based on wavelet transform. The choice and combination of these features have influences on the classification accuracy[11]. In addition to manual selection of different feature subsets, various automated feature selection methods can effectively select the most informative features and improve model performance. Principal component analysis (PCA) and genetic algorithm (GA) are widely used in previous research[12][4], which are used for unsupervised dimensionality reduction and feature subset optimization based on classification performance, respectively. In summary, a proper selection of features is therefore crucial for achieving high classification accuracy.

In this study, a variety of AI algorithms were used, including machine learning methods such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), as well as neural network based methods like Convolution Neural Networks (CNN) and Long Short Term Memory(LSTM). The genetic algorithm was applied to the machine learning methods to select the optimal subset of features for the SVM and KNN methods. In terms of neural network based algorithms, three different methods were used. First, a 1D-CNN-LSTM was used to directly process the acceleration signals after preprocessing, enabling the model to automatically learn temporal features from the original time series data. Second, a CNN architecture was also applied to input derived from wavelet-transformed representations, where the transformed signals were visualised as pseudo images. In addition, the model should be able to reject unknown patterns in real-world applications. The system may receive movement or weight categories that have not appeared in the training set. If the model cannot recognize these unknown inputs, it may force them to be misclassified as known categories. Regrading systems with humans involved, misclassification is more dangerous than rejection. So this study used the Convolutional Prototype Network (CPN) method to achieve open-set recognition. By using different machine learning, deep learning, and open set recognition techniques, this study provides an explorative framework of AI application in classification of handheld Object Weight during human motion.

To sum up, this study investigate whether multiple AI models trained on IMU data can effectively classify the weight of handheld objects across different movement types, thereby enabling a workload recording system to prevent strain injury or be applied to the control algorithm of exoskeleton.

This thesis is structured as follows: the next section provides an explanation of feature engineering techniques and the AI methods used. Section III describes the experimental setup used to obtain the dataset and details the implementation of the pipeline of classification and various AI methods. The experimental results are presented and discussed in Sections IV and V. Finally, conclusions are drawn in Section VI.

II. Preliminaries

In this section, the working principles of several feature extraction techniques and AI methods are explained.

A. Wavelet transform

Wavelet Transform is a powerful signal processing technique used to decompose a signal into components with different frequency contents at various resolutions. Unlike the Fourier Transform, which provides only frequency-domain information and assumes signal stationarity, the Wavelet Transform offers both time and frequency localization, making it especially effective for analyzing non-stationary signals.

The Wavelet Transform operates by scaling and translating a fundamental function known as the mother wavelet to generate a family of wavelet functions. These wavelets are then used to analyze different parts of the signal, capturing both short-term high-frequency details and long-term low-frequency trends.

Wavelet transform can be divided into Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). CWT performs transformation on a signal in continuous time, applying the wavelet function across the entire time domain and analyzing the signal at all possible scales. It provides rich time-frequency information. However, CWT involves high computational complexity and contains a large amount of redundant information. To improve computational efficiency and reduce redundancy, DWT uses discrete scales and translation parameters. To improve computational efficiency and reduce redundancy, the Discrete Wavelet Transform (DWT) uses discrete scales and translation parameters. It typically decomposes the signal using a filter bank consisting of high-pass and low-pass filters. As shown in Figure 2, the input signal $X[n]$ is passed through both filters, and each output is downsampled by a factor of 2. The high-pass filter extracts the detail coefficients $X_{1,D}[n]$, while the low-pass filter yields the approximation coefficients $X_{1,A}[n]$. This process can be recursively applied to the approximation coefficients to enable multi-level time-frequency analysis of the signal.

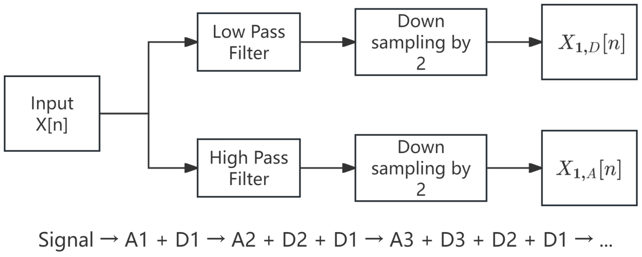

Fig. 2: Principle of DWT

B. Genetic algorithm

Genetic Algorithm (GA) is an evolutionary optimization method. As illustrated in the figure 3, the algorithm begins with population initialization, where a set of candidate solutions is randomly generated. Each individual in the population is then evaluated using a fitness function during the evaluation step. Based on their fitness, individuals are selected in the selection phase to become parents for the next generation. The selected individuals undergo crossover to produce offspring, combining features from two parents. Subsequently, random alterations are introduced during the mutation step to maintain diversity in the population. After mutation, the new generation is evaluated again, and this loop continues until a predefined termination criterion is met, such as reaching a maximum number of generations or achieving a satisfactory fitness level. Finally, the algorithm outputs the best individuals as the solution set.

C. KNN

The k-Nearest Neighbors (kNN) algorithm is a supervised learning approach extensively utilized for both classification and regression tasks. Its fundamental principle involves determining the class of an unseen data instance by analyzing its proximity to previously labeled data. Specifically, the algorithm identifies the k closest samples in the feature space and assigns the class most frequently represented among these neighbors to the new instance.

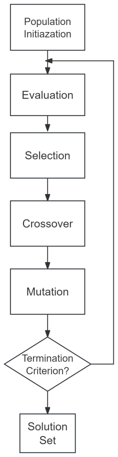

Fig. 3: Schematic Diagram of Genetic Algorithm

D. SVM

Support Vector Machine (SVM) is widely used supervised machine learning algorithms designed to find an optimal hyperplane that separates data points belonging to different classes. The ideal hyperplane maximizes the margin between itself and the nearest data points from each class, thereby increasing the likelihood of correctly classifying unseen data. However, in many real-world scenarios, perfectly separating the classes is not feasible due to overlapping data or noise. In such cases, the model can be adjusted to tolerate a certain degree of misclassification by introducing a soft margin, which balances the trade-off between maximizing the margin and minimizing classification errors.

E. CNN

Convolutional Neural Network (CNN) is a deep learning model specifically designed to process data like images or time series signals. Inspired by the visual processing mechanism of the human brain, CNNs are highly effective at automatically learning spatial hierarchies of features through the use of convolutional operations. A Convolutional Neural Network consists of multiple layers, typically including convolutional layers, activation layers, and pooling layers. In the convolutional layers, the input is convolved with learnable filters to automatically extract local features. The activation layers apply non-linear functions, such as ReLU, which enable the network to learn complex, non-linear relationships. Pooling layers downsample the feature maps, selecting the most significant features while reducing computational complexity and the risk of overfitting. By stacking these layers, the model gradually extracts more abstract and high-level features. Finally, the output is passed through fully connected layers and a softmax function to produce probability values for classification.

F. LSTM

The Long Short-Term Memory (LSTM) network is a special type of recurrent neural network designed to effectively capture long-range dependencies in sequential data. As shown in the figure 4, at each time step t, the LSTM unit receives the previous hidden state H_{t-1}, the previous cell state C_{t-1}, and the current input x_t. Within the cell, three gate mechanisms—forget gate, input gate, and output gate—control the information flow. The forget gate determines how much of the previous cell state should be retained, the input gate regulates how much new information is stored in the cell state, and the output gate decides what information from the cell state is passed to the next hidden state H_t. These gates together update the cell state to C_t and produce the new hidden state H_t, enabling the model to maintain and update long-term information over time.

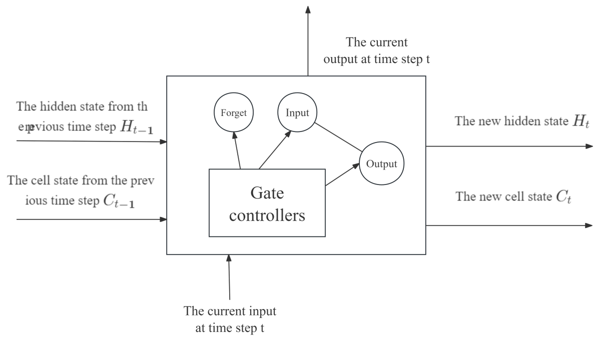

Fig. 4: Schematic Diagram of LSTM Model

G. CPN

The Convolutional Prototype Network (CPN) aims to construct a CNN feature extractor and multiple prototypes of known classes[13]. The implementation of the CPN can be divided into two stages which are the training phase and the testing phase, as illustrated in the figure 5.

In the training phase, labeled data from known classes is used to train a CNN model, which learns a feature extractor to convert input samples into deep representations. These learned representations are used in the subsequent testing phase. At the same time, the model learns a prototype for each class, which serves as the center of that class in the feature space, based on the distribution of samples belonging to that class.

In the testing phase, the trained CNN model is used to extract deep representations from new input samples. These representations are then matched against the set of learned prototypes from the training phase. As illustrated in the figure 6, the distances between the extracted feature and each prototype are calculated. If the distance to one of the prototypes is sufficiently small within a predefined threshold. then the sample is classified into the corresponding known class, and its label is returned. If the distances to all prototypes exceed the threshold, the sample is considered to belong to an unknown class and is thus rejected.

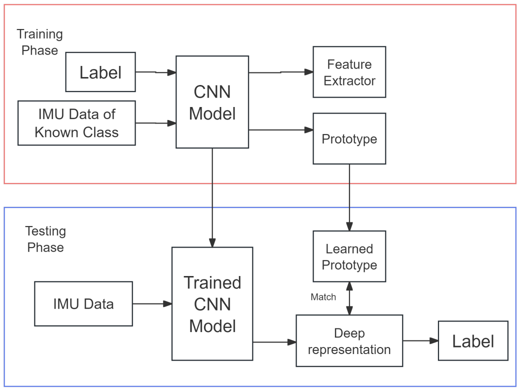

Fig. 5: Overview of CPN Architecture

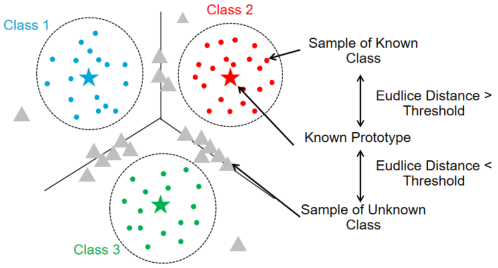

Fig. 6: Classifying of Known Class and Rejecting Unknown Class

III. Methodology

A. Experiment Design

To investigate the feasibility of using only the IMU sensor and AI methods to identify the weight of handheld objects, a series of controlled experiments with one subject were carried out in which single participants performed repetitive motion tasks under different load conditions. The experiments were designed to simulate real upper limb movements while minimizing Interfering variables such as fatigue and environmental noise.

1) Experiment Goal: The primary goal of the experiment is to collect high resolution, labelled time series data from a forearm mounted IMU during two main types of movement tasks while the participant holds objects of different masses. The resulting dataset will be used to train supervised machine learning models for multiple class weight classification and open set classification based on IMU data.

2) Experiment Setup and Device: The experiment contains two main movement types and one support movement type as a distractor. The two main movement types are walking and running in place. The walking task required participants to alternately lift their knees and swing their arms naturally at a walking like rhythm while remaining stationary [14]. The running task involved higher frequency jogging in place, which increased the dynamic activity of the upper limbs. All movements were controlled by the participants themselves, but the rhythm was kept within a consistent range. In each exercise mode, the tester held objects of different masses, which were 0kg (empty hands), 0.5kg, 1kg and 1.5kg. There were eight combinations in total and each unique combination of movement type and weight constitutes a different class label in the classification task. Each combination of movement type and weight perform two times. Arm swinging while standing still is used as a support movement. Under this situation, the tester stands stationary and continuously swings both arms forward and backwards without lifting the legs.

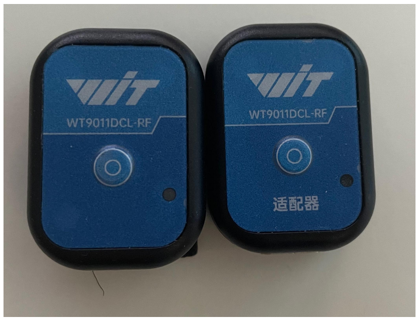

Fig. 7: IMU Sensor(left) and its Adapter(right)

The high precision WT9011DCL-RF IMU sensor [15] is used to collect the movement data as shown in figure 7. The IMU sensor was mounted on the mid-forearm of the dominant arm, with its Y-axis oriented toward the palm and the Z-axis aligned vertically upward, perpendicular to the ground and an elastic strap was used to ensure that the device was stable and fit during exercise as shown in figure 8. The sensor sampling frequency was set to 50 Hz, providing sufficient temporal resolution to capture rapid arm movements and subtle dynamic variations during different tasks.

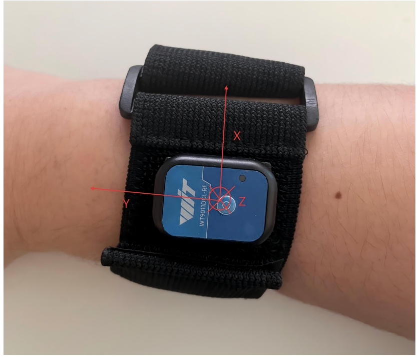

Fig. 8: Position and Orientation of IMU Sensor

3) Experiment Procedure and Data Management: The experiment was conducted on multiple non-consecutive days to introduce natural within-subject variability and reduce fatigue effects. A specific exercise weight condition was randomly selected and performed. There was a 20-minute break between each section to prevent fatigue interference.

The recorded data were stored in plain text (.txt) format, containing time-stamped entries for each sample. Each data point includes measurements of three axis linear acceleration (Ax, Ay, Az), resulting in a three dimensional time series signal associated with each movement.

The experimental conditions, including movement types, weight levels, duration, and repetition counts, along with their associated labels, are shown in the table 1.

Label ID	Movement Type	Weight (kg)	Duration (min)	Number of Experiment
C1	Walking in place	0	5	2
C2	Walking in place	0.5	5	2
C3	Walking in place	1	5	2
C4	Walking in place	1.5	5	2
C5	Running in place	0	5	2
C6	Running in place	0.5	5	2
C7	Running in place	1	5	2
C8	Running in place	1.5	5	2
C9	Swinging while standing still	None	5	1

Table 1: Experimental conditions for movement types, weights, durations, and repetitions.

B. Signal preprocessing

The raw acceleration signals require preprocessing and an example of raw acceleration data is shown in figure 9. Firstly, a butterworth high pass filter with a cutoff frequency of 0.5 Hz is applied to remove low frequency components, such as gravitational acceleration and slow drifts, thereby keeping high frequency information related to motion. Subsequently, the acceleration data are standardized to have zero mean and unit variance for each axis. The mean and standard deviation are computed independently along the temporal dimension for each axis. To ensure numerical stability, the standard deviation is set to 1 when it is less than 1×10^{-6}. This normalization is crucial for maintaining a consistent numerical range across input channels, which facilitates faster convergence and improved overall performance of the model.

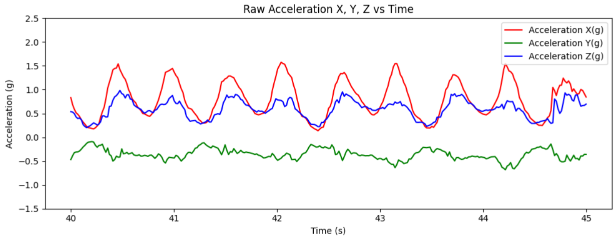

Fig. 9: Raw 3 axis acceleration data

Finally, the continuous IMU signals are segmented using a fixed length sliding window approach. Each window consists of 128 sampling points, with a 50% overlap between consecutive windows. The 3 axis acceleration data after signal preprocessing and segmentation is shown in the figure 10. The rectangular boxes outlined with double-dashed lines in different colors represent different sliding windows. Longer windows help capture the complete temporal characteristics of complex movements, while the overlapping strategy increases the number of training samples and smooths transitions between windows and improve the model's ability to perceive continuous temporal changes.

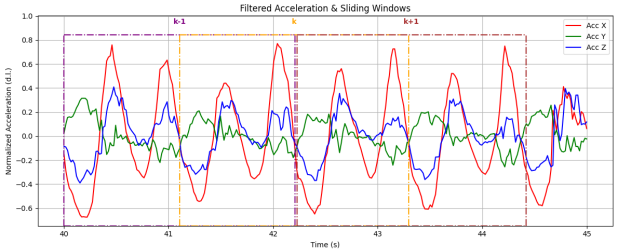

Fig. 10: Filtered, normalized and segmented acceleration data

After the above preprocessing steps, the resulting data structure is a tensor of shape $(N, 128, 3)$, where N denotes the number of samples obtained through sliding window segmentation, 128 represents the temporal length of each window, and 3 corresponds to the three acceleration channels. Each sample is also associated with a corresponding label indicating the handheld weight and movement type.

C. Feature Engineering

To effectively represent the underlying motion characteristics within each segmented IMU window, a comprehensive set of features is extracted from multiple domains, including time domain, frequency domain, and time-frequency domain. These features are designed to capture hidden patterns in acceleration signal. Specifically, four types of feature are computed as following.

1) Time domain features: For each segmented window $\mathbf{X} \in \mathbb{R}^{128\times 3}$, six time-domain features are computed independently for each of the three axis. These included the mean, standard deviation, root mean square (RMS), skewness, kurtosis, and zero-crossing rate (ZCR). The mean and standard deviation described the central tendency and dispersion of the signal, respectively. The RMS served as an energy-related descriptor, while skewness and kurtosis captured the asymmetry and peakedness of the signal distribution. The ZCR quantified the frequency of sign changes in the signal, thereby reflecting its variability [16]. As a result, a total of 18 time domain features (6×3) are obtained for each window.

2) Frequency domain features: In the frequency domain, the Fast Fourier Transform (FFT) is applied to each axis to obtain the magnitude spectrum. The spectral energy was then aggregated across three empirically defined frequency bands: low-frequency (bins 0–9), mid-frequency (bins 10–29), and high-frequency (bins 30 to the Nyquist frequency). These features were designed to reflect the energy distribution across different motion intensities. As a result, a total of 9 frequency domain features are obtained for each window.

3) Time frequency domain features: In order to get time frequency domain features, a 3 level DWT is applied to each signal axis using the Daubechies-4 (db4) wavelet basis. From each set of wavelet coefficients at every decomposition level, three features were extracted. The first one is energy, which represents the signal power within each frequency band. The second one is entropy, which quantifies the complexity or irregularity of the signal. The third one is the maximum absolute amplitude, which reflects the peak signal strength in each sub-band[17]. As a result, a total of 36 time frequency domain features are obtained for each window.

4) Wavelet based pseudo images: To support CNN based models, a wavelet based pseudo image is constructed for each segmented window. A 3 level discrete wavelet transform (DWT) was applied to each axis of the input signal using the Daubechies-4 (db4) wavelet basis. This decomposition resulted in four sets of wavelet coefficients per axis which are one approximation and three detail components.

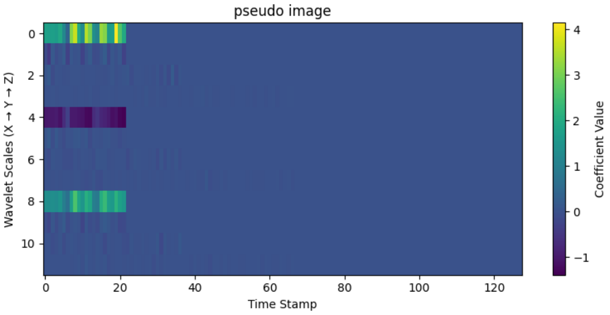

Fig. 11: Pseudo image based on DWT

To get the pseudo image, each coefficient set is zero-padded to match the original window length, ensuring a uniform temporal dimension across all levels and axis. These padded coefficients were then stacked along a new dimension, where the vertical axis of the resulting image corresponds to the decomposition scale and sensor axis, and the horizontal axis corresponds to time. An example of pseudo image is shown in figure11 and this image can be used as the input feature of CNN, similar to the RGB image input in image classification tasks. The vertical axis labeled "Wavelet Scales (X $\rightarrow$ Y $\rightarrow$ Z)" represents the concatenation of wavelet decomposition levels across the three sensor axis. These coefficients are stacked sequentially, with the four levels from the X-axis followed by those from the Y-axis and then the Z-axis, resulting in a total of 12 rows.

D. SVM and KNN method with GA

In order to improve classification performance and reduce model complexity, the genetic algorithm is used for feature selection before the model training. Each individual in the population is represented as a binary vector of length 63, where each gene corresponds to a specific feature. A total of 63 features were constructed during the feature engineering stage. A value of 1 indicates that the corresponding feature is selected, while a value of 0 denotes exclusion. The population is initialized with 50 individuals and evolved over 20 generations. During the evolutionary process, standard genetic operations are applied, including tournament selection (with a tournament size of 3), two-point crossover (with a crossover probability of 0.6), and bit-flip mutation (with a mutation probability of 0.2 per individual and 0.1 to 0.05 per bit).

The fitness of each individual is defined as the mean classification accuracy obtained from 5 fold cross validation using only the selected subset of features. Depending on the configuration, either a Support Vector Machine (SVM) classifier or a K-Nearest Neighbors (KNN) classifier with $k = 5$ is used to evaluate the fitness. In each evaluation, the classifier is retrained exclusively on the subset of features indicated by the individual's binary representation.

After the completion of the evolutionary process, the individual with the highest fitness score has the most effective feature subset and is selected for training the final classification model.

E. 1D-CNN-LSTM

To effectively model both local temporal dependencies and long-range sequential patterns in multivariate time-series data, a hybrid deep learning architecture combining one dimensional convolutional neural networks (1D-CNN) and bidirectional long shortterm memory (BiLSTM) layers is constructed.

The model is initialized with a 1D convolutional layer consisting of 64 filters with a kernel size of 3 and ReLU activation. This layer is followed by batch normalization and max pooling with a pool size of 2. Subsequently, a second convolutional layer with 128 filters and similar configurations is applied, again followed by batch normalization and pooling. Through these convolutional layers, hierarchical local patterns were learned and the temporal resolution is reduced.

To capture long-range temporal dependencies, the output of the convolutional block is passed into a bidirectional LSTM layer with 64 units in each direction. This structure allowed contextual information from both past and future time steps to be effectively integrated. Then the resulting sequence representation is processed by a fully connected dense layer with 64 units and ReLU activation, and a dropout layer with a rate of 0.3 is used to mitigate overfitting. Finally, a softmax-activated output layer was employed, with the number of neurons corresponding to the number of target classes.

The entire model is trained using the Adam optimizer with a learning rate of 0.001, and the sparse categorical cross-entropy loss function is used. Model performance is evaluated in terms of classification accuracy.

F. wavelet-CNN

To effectively utilize the time-frequency characteristics of non stationary signals, a convolutional neural network architecture is developed based on pseudo images generated from discrete wavelet transform. Unlike conventional image data, the input to the model consisted of synthetic images constructed by stacking multi-level wavelet coefficients. Specifically, each input sample was represented as a three dimensional pseudo image of shape (L, T, C), where $L = 4$ denoted the number of wavelet decomposition levels, $T = 128$ is the length of each wavelet sub-band, and $C = 3$ corresponded to the number of signal channels.

The model architecture followed a sequential design. It begins with a 2D convolutional layer containing 32 filters of size $(2, 3)$ and ReLU activation, aimed at capturing low-level spatiotemporal features across adjacent wavelet levels and time frames. This is followed by batch normalization and a max-pooling layer with a pooling size of $(1, 2)$ to reduce temporal resolution while preserving wavelet-level structure.

A second convolutional layer with 64 filters and the same kernel size is then applied to learn more abstract and complex features. Similar to the first block, this layer is followed by batch normalization and temporal downsampling via max pooling. After feature extraction, the resulting tensor is flattened and passed through a fully connected dense layer with 128 units and ReLU activation. A dropout layer with a rate of 0.5 is incorporated to prevent overfitting. Finally, a softmax-activated dense layer is used for classification, with the number of output neurons equal to the number of target classes.

This method used same optimizer and evaluation metrics that mentioned in 1D-CNN-LSTM method.

G. CPN with Open Set Recognition

To improve class discriminability and enable the rejection of unknown samples during classification, the CPN based on a 1D convolutional architecture with integrated prototype learning is used. The input to the network is a multivariate time-series of shape $(128, 3)$, where 128 is the temporal window length and 3 corresponds to the number of sensor channels. Feature extraction is performed using two 1D convolutional layers with 64 and 128 filters, each followed by batch normalization. A max pooling layer reduces temporal resolution, and global average pooling aggregates temporal features. The resulting vector is projected into a 32-dimensional embedding space via a dense layer and normalized to unit length.

To achieving feature learning, a custom Prototype Layer maintains a learnable set of class prototypes, one for each of the K target classes. During training, the model receives both the input signal and the ground truth label to compute a prototype alignment loss, encouraging features to cluster around their corresponding class prototypes. This loss is scaled by a factor $\alpha = 1.0$ and added to the overall training objective alongside the standard categorical cross-entropy loss. The final classification logits are produced by a dense layer on the normalized feature vector. During inference, class prediction is based on the nearest prototype in the embedding space, and a threshold can be used to reject unfamiliar samples based on distance.

IV. RESULT

A. SVM with GA Feature Selection

The confusion matrix for single movement type and different object weights (C1–C4) is shown in Figure 12. Each cell in the matrix represents the percentage of instances belonging to a true class (rows) that were predicted as a certain class (columns). For example, for the cell in the first row and first column, 86% of the samples that actually belong to C1 are correctly predicted to be C1.

The diagonal elements indicate the proportion of correctly classified samples for each class, which also known as the recall value. Specifically, the model correctly identified 86% of Class 1 (C1) instances, 87% of Class 2 (C2), 86% of Class 3 (C3), and 85% of Class 4 (C4). The average classification accuracy across the 4 classes is 86.3%.

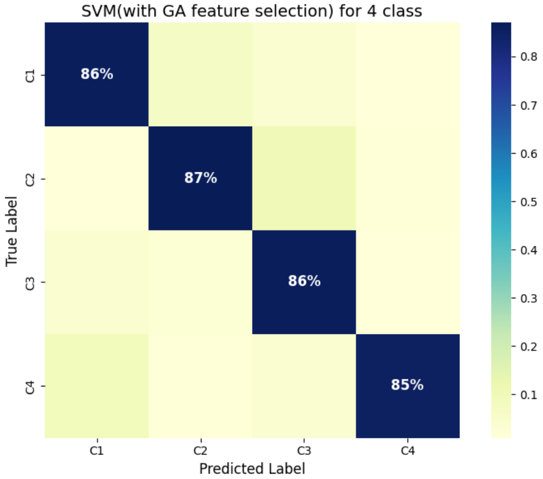

Fig. 12: Confusion Matrix of SVM+GA for 4 Class

The normalized confusion matrix for different movement types and different object weights (C1–C8) is shown in Figure 13. The average classification accuracy across the 8 classes is 82.1%.

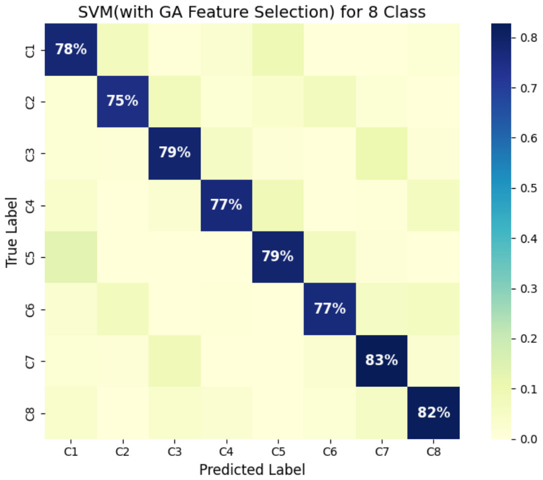

Fig. 13: Confusion Matrix of SVM+GA for 8 Class

B. KNN with GA Feature Selection

The confusion matrix for different movement types and object weights (C1–C8) is shown in Figure 14. The average classification accuracy across the eight classes is 68.4%.

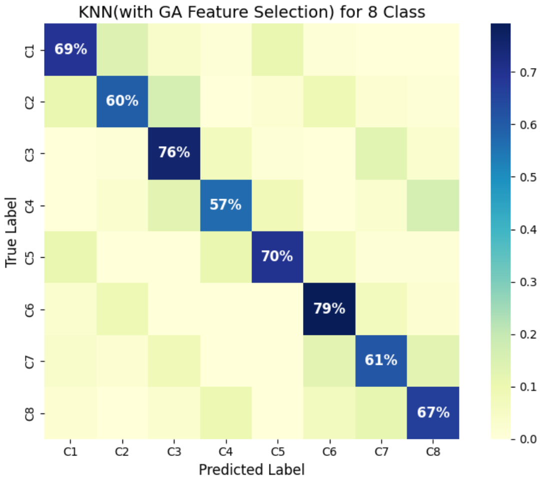

Fig. 14: Confusion Matrix of KNN+GA for 8 Class

C. 1D-CNN-LSTM

The 1D-CNN-LSTM model was trained on data including different movement types and object weights. The training loss and accuracy curves are shown in Figure15. The model performance stabilized after approximately 5 epochs. The validation accuracy reached a peak value of approximately 94%.

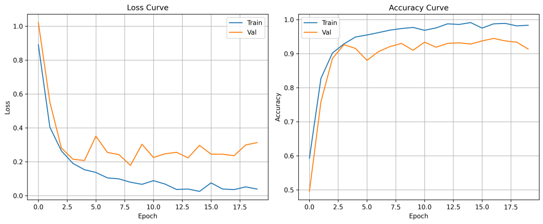

Fig. 15: Training Loss and Accuracy Curves of 1D-CNN-LSTM Method for 8 Class

D. Wavelet-CNN

The Wavelet-CNN model was trained on data including different movement types and object weights. The training loss and accuracy curves are presented in Figure 16. The validation accuracy reached approximately 91%.

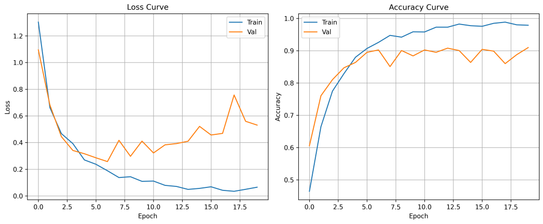

Fig. 16: Training Loss and Accuracy Curves of Wavelet-CNN Method for 8 Class

E. CPN Open set Classification

For the case of single movement type as known class, the CPN model was trained using data from a single movement type (walking) with different object weights (C1–C4) as known classes. An unseen movement type (C9) was used as an unknown class. The resulting confusion matrix is shown in Figure 17. The classification accuracy for known classes ranges from 74% to 86%, while the unknown class (C9) was identified with 71% accuracy.

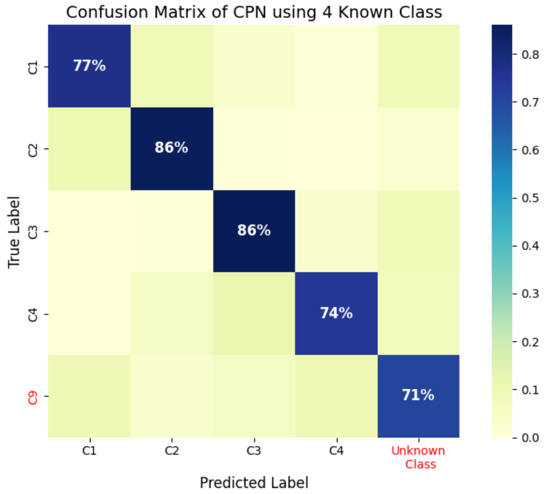

Fig. 17: Confusion Matrix of CPN using 4 Known Class

In the second open-set setting, the CPN model was trained using two movement types and object weights (C1–C8) as known classes. An unseen movement type (C9) was used as the unknown class. The confusion matrix is shown in Figure 18. The classification accuracy for known classes ranged from 54% to 72%, while the unknown class (C9) was identified with an accuracy of 53%.

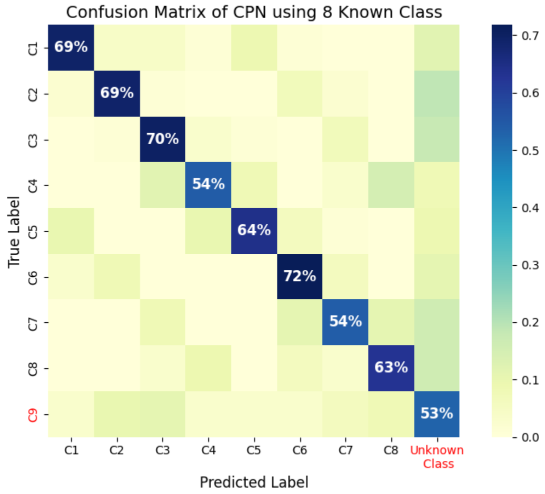

Fig. 18: Confusion Matrix of CPN using 8 Known Class

V. Discussion

This study extracted multiple features in different domains from acceleration signal and used both machine learning methods (including Support Vector Machine and K-Nearest Neighbors) and deep learning methods (including 1D-CNN-LSTM and Wavelet-CNN) to evaluate human activity classification performance under different combinations of movement types and handheld object weights. Additionally, the Convolutional Prototype Network (CPN) is used to counter open set classification tasks.

A. Performance of Machine Learning Methods

Under the scenarios involving only one movement type and varying object weights, the combination of GA-based feature selection and SVM achieved high classification accuracy,which showed strong discriminative power in relatively simple classification tasks. However, when the task complexity increased to include multiple movement types and associated weights, the accuracy of SVM declined slightly but still maintained acceptable classification performance, indicating its good generalization capability.

In contrast, KNN using the same GA-selected features showed relatively lower accuracy. This is likely due to KNN's sensitivity to local data distributions and vulnerability to inter class overlap and noise, which make it more difficult to build stable decision boundaries in high-dimensional feature spaces.

B. Advantages of Deep Learning Methods

The deep learning model based on 1D-CNN-LSTM performed excellently in classifying different movement types and corresponding object weights, achieving a validation accuracy of approximately 94%. This performance is attributed to the CNN component's ability to extract local temporal features and the LSTM component's strength in modelling long-term dependencies.

Compared to this, the Wavelet-CNN model also yielded satisfactory performance (around 91% accuracy), though slightly lower than 1D-CNN-LSTM. This may be due to the insufficient feature representation during the pseudo-image construction process. Also, the Wavelet-CNN method relied more heavily on spatial features within the image representation and lacks strong modelling capacity for temporal sequences, as provided by LSTM.

Overall, deep learning methods showed clear advantages in complex feature integration and sequence modelling, and have better performance than machine learning models in this study.

C. Challenges and Performance in Open-set Classification

For the open set classification, the CPN model was used to reject unknown movement patterns. When trained on a single movement type with varying weights, the model effectively distinguished unseen movement types, which showing a strong ability to separate known from unknown distributions. However, when the known class set was broadened to include multiple movement types and weight combinations, recognition performance declined markedly. In particular, the model's accuracy in identifying the unknown class (C9) dropped to around 53%. This decline indicates that greater intra- and inter-class similarity among the known categories blurs the boundaries between prototypes, thereby weakening the model's capacity to detect unfamiliar patterns.

D. Research Contributions and Comparative Analysis

Compared with previous studies that typically focused on movement type or object weight classification separately[5][7][6], this research integrated both aspects, constructing a joint classification task that is more aligned with real-world applications. When different weights under the same movement type were used as known classes, models achieved better performance. For tasks involving classification between multiple movement types, accuracy was slightly lower. This is because the two selected movements in this study are relatively similar in motion pattern. In comparison, prior works often used more distinct activities such as squats, deadlifts, rowing, and bench press, which have clearer differences in acceleration pattern. This observation is further supported by the confusion matrix results as shown in figure13 and figure14, where some samples originally belonging to classes C1–C4 were misclassified as their corresponding weight matched counterparts in C5–C8. Specifically, samples of C1 (0 kg, walking) were often confused with C5 (0 kg, running), suggesting some similarity between classes with the same weight but different movement types.

In addition, previous studies mainly used time domain features and machine learning methods, this study used advanced methods such as time frequency domain features, deep learning models, and open set classification using the CPN model, contributing both practical value and methodological innovation. Particularly, the open set capability offers a viable solution for intelligent wearable devices or fitness monitoring systems to handle unknown behaviour patterns in real-world applications.

E. Limitations and Further Work

Although this study had achieved certain results in classification accuracy, comparison in different model and open set classification, it still has the following limitations.

1) Limited number of movement types and resolution of weight: Only two relatively similar movement types were considered as main movement types in this study. In the real world application, it limits the generalizability of the models to more diverse and complex human activities. Future research should include a wider range of activities with more distinct motion characteristics to fully evaluate model performance. The classification of object weight was discretized into a limited number of class, which may not capture finer grained variations in weight. In real world applications, object weights may vary continuously rather than in fixed intervals. This discretization could lead to reduced sensitivity and limit the model's ability to generalize across subtle weight differences. Future studies may explore regression based methods to improve resolution and realism.

2) Single subject limitation: This study was conducted using data from only one participant. As a result, the model's performance may be overfitted to the movement patterns associated with that individual. This limits the generalizability of the results to a broader population. Future research should include data from multiple subjects with diverse physical conditions to ensure the robustness and applicability of the proposed methods.

3) Unclear decision boundaries in open set classification: When using CPN for open set classification, the model's ability to distinguish unknown classes degraded as the number and diversity of known classes increased. This indicates that under conditions of high intra-class similarity, prototype boundaries become less distinguishable. Future work may explore more robust open set classification strategies.

In addition, future studies could explore increasing the number of IMU sensors. For example, placing one on the upper arm and another on the forearm to capture more comprehensive movement information. This multi position sensing strategy may enhance the model's ability to improve classification performance.

VI. Conclusion

This study provides an explorative AI-based solution for classification of handheld object weight during human motion by using IMU data. By comparing traditional machine learning approaches (SVM and KNN with GA-based feature selection) and deep learning models (1D-CNN-LSTM and Wavelet-CNN), the study highlights the clear advantages of deep architectures in handling complex, non linear motion patterns. Among all tested methods, the 1D-CNN-LSTM model achieved the highest classification accuracy, while the Convolutional Prototype Network (CPN) introduced open set recognition capability.

To sum up, this study successfully met the proposed hypothesis, showing effective classification of motion patterns across different movement types and object weights by using only IMU data and AI method, as well as reasonable performance unknown class rejection function. These results provide a solid base for designing an IMU based workload recording system for real world applications to prevent strain injury or to be integrated into wearable exoskeletons.

References

[1] B. R. da Costa and E. R. Vieira, "Risk factors for work-related musculoskeletal disorders: A systematic review of recent longitudinal studies," *American Journal of Industrial Medicine*, vol. 53, no. 3, pp. 285–323, 2010. DOI: 10.1002/ajim.20750. [Online]. Available: https://doi.org/10.1002/ajim.20750.

[2] G. Li, B. Wan, K. Su, J. Huo, C. Jiang, and F. Wang, "Semg and imu data-based hand gesture recognition method using multistream cnn with a fine-tuning transfer framework," *IEEE Sensors Journal*, vol. 23, no. 24, pp. 31 414–31 424, 2023. DOI: 10.1109/JSEN.2023.3327999.

[3] H. Li, S. Guo, D. Bu, H. Wang, and M. Kawanishi, "Subject-independent estimation of continuous movements using cnn-lstm for a home-based upper limb rehabilitation system," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6403–6410, Oct. 2023. DOI: 10.1109/LRA.2023.3303701.

[4] H. Li, S. Guo, D. Bu, and H. Wang, "A two-stage ga-based semg feature selection method for user-independent continuous estimation of elbow angles," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, no. Art no. 6502909, pp. 1–9, 2023. DOI: 10.1109/TIM.2023.3276522.

[5] C. Crema, A. Depari, A. Flammini, E. Sisinni, T. Haslwanter, and S. Salzmann, "Imu-based solution for automatic detection and classification of exercises in the fitness scenario," in *2017 IEEE Sensors Applications Symposium (SAS)*, 2017, pp. 1–6. DOI: 10.1109/SAS.2017.7894068.

[6] P. Hausberger, A. Fernbach, and W. Kastner, "Imu-based smart fitness devices for weight training," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 5182–5189. DOI: 10.1109/IECON.2016.7793510.

[7] M. Ravizza, L. Giani, F. J. Sheiban, A. Pedrocchi, J. DeWitt, and G. Ferrigno, "Imu-based classification of resistive exercises for real-time training monitoring on board the international space station with potential telemedicine spin-off," *PLOS ONE*, vol. 18, no. 8, e0289777, 2023. DOI: 10.1371/journal.pone.0289777. [Online]. Available: https://doi.org/10.1371/journal.pone.0289777.

[8] C. Crema, A. Depari, M. Flammini, E. Sisinni, and A. Vezzoli, "The wearphone: Changing smartphones into multichannel vital signs monitors," in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, IEEE, 2016, pp. 1–6. DOI: 10.1109/MeMeA.2016.7533759.

[9] R. Romijnders and E. Warmerdam, "Validation of imu-based gait event detection during curved walking and turning in older adults and parkinson's disease patients," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 28, 2021. DOI: 10.1186/s12984-021-00828-0. [Online]. Available: https://doi.org/10.1186/s12984-021-00828-0.

[10] A. Gouda and J. Andrysek, "Rules-based real-time gait event detection algorithm for lower-limb prosthesis users during level-ground and ramp walking," *Sensors (Basel, Switzerland)*, vol. 22, no. 22, p. 8888, 2022. DOI: 10.3390/s22228888. [Online]. Available: https://doi.org/10.3390/s22228888.

[11] S. Briouza, H. Gritli, N. Khraief, S. Belghith, and D. Singh, "Emg signal classification for human hand rehabilitation via two machine learning techniques: Knn and svm," in *2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, Hammamet, Tunisia: IEEE, 2022, pp. 412–417. DOI: 10.1109/IC_ASET53395.2022.9765856.

[12] F. Cruciani, A. Vafeiadis, C. Nugent, *et al.*, "Feature learning for human activity recognition using convolutional neural networks: A case study for inertial measurement unit and audio data," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, Jan. 2020. DOI: 10.1007/s42486-020-00026-2.

[13] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 2018, pp. 3474–3482. DOI: 10.1109/CVPR.2018.00366. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00366.

[14] M. P. Ford, R. C. Wagenaar, and K. M. Newell, "Arm constraint and walking in healthy adults," *Gait & Posture*, vol. 26, no. 1, pp. 135–141, 2007. DOI: 10.1016/j.gaitpost.2006.08.008.

[15] WitMotion, *Wt9011dcl-rf imu wireless accelerometer sensor, 32slave cascade gyro 2.4g radio frequency tilt sensor for human movement tracking*, https://witmotion-sensor.com/products/wt9011dcl-rf-imu-wireless-accelerometer-sensor-32slave-cascade-gyro-2-4gradio-frequency-tilt-sensor-for-human-movement-tracking.

[16] A. O. Souza, J. Grenier, F. Charpillet, P. Maurice, and S. Ivaldi, "Towards data-driven predictive control of active upper-body exoskeletons for load carrying," in *2023 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*, Berlin, Germany: IEEE, 2023, pp. 59–64. DOI: 10.1109/ARSO56563.2023.10187548.

[17] A. Belkhou, A. Jbari, and L. Belarbi, "A continuous wavelet based technique for the analysis of electromyography signals," Nov. 2017, pp. 1–5. DOI: 10.1109/EITech.2017.8255232.