

# An Integrated MLOps Framework for Developing, Tracking, and Deploying Ensemble Learning Models for Lower Limb Motion Classification

R. Anitha<sup>1</sup>, Geetesh N.<sup>2</sup>, Vikash Kumar<sup>1</sup>, K. Mohanavelu<sup>4</sup>

<sup>1, 4</sup>Defence Bioengineering and Electromedical Laboratory (DEBEL), DRDO, Bengaluru, India

<sup>2, 3</sup>Department of Computer Science in AI ML, Vijaya Vittala Institute of Technology, Bengaluru, India

**Abstract:** *Wearable inertial sensing has become an effective approach for lower limb motion classification, yet many existing studies primarily emphasize classification accuracy while providing limited support for reproducible model development, experiment tracking, and deployment. This paper presents a reproducible Machine Learning Operations (MLOps) framework for lower limb motion classification using biomechanically meaningful lower-limb joint-angle features obtained from wearable inertial measurement unit (IMU) measurements. The framework processes hip, knee, and ankle joint-angle signals, performs sliding-window feature extraction using statistical, derivative based, and frequency-domain descriptors, and trains multiple supervised machine learning models, including XGBoost, Extra Trees, CatBoost, Ridge Classifier, and Histogram Gradient Boosting. The complete machine learning lifecycle is integrated using Git for version control, MLflow for experiment tracking, Docker for reproducible containerized deployment, and Streamlit for interactive inference and visualization. The framework was evaluated using Xsens Awinda data collected from five subjects performing six lower limb activities, namely Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown, with 1,617 feature windows used for the reported test evaluation. Experimental results demonstrate that the Extra Trees classifier achieved the highest performance, with an accuracy of 99.59 percent and a Macro F1-score of 0.9939, outperforming the other evaluated machine learning models. The proposed framework enables reproducible model training, artifact management, deployment, and interactive inference within a unified workflow while improving the interpretability of activity recognition through joint angle-based biomechanical feature representation. The proposed MLOps framework provides a practical and scalable solution for developing reproducible wearable sensing applications and offers a foundation for future real-time, multimodal, and edge deployable lower limb motion analysis systems.*

**Keywords:** MLOps, Machine Learning Pipeline, Lower Limb Motion Classification, Joint Kinematics, IMU, MLflow, Docker, Git, Streamlit, Ensemble Learning, Experiment Tracking, Model Deployment

## 1. Introduction

Accurate lower limb motion classification has become increasingly important for wearable robotics, rehabilitation engineering, gait analysis, elderly mobility monitoring, sports biomechanics, and human activity recognition. Reliable recognition of human locomotion enables intelligent assistive technologies to understand user movement patterns, support clinical decision-making, improve rehabilitation outcomes, and enhance the performance of wearable robotic systems. With the growing adoption of wearable healthcare devices and intelligent monitoring systems, there is an increasing demand for motion classification techniques that are accurate, reliable, computationally efficient, and suitable for real-world deployment. Wearable inertial sensing has emerged as an effective technology for these applications due to its portability, low cost, non-invasive nature, and ability to capture human movement outside laboratory environments. Compared with vision-based motion capture systems, wearable inertial measurement units (IMUs) provide continuous motion monitoring while preserving user privacy, minimizing infrastructure requirements, and enabling real-time operation in both indoor and outdoor environments [1], [2], [7], [21], [22], [27]. Consequently, IMU-based activity recognition has become an active area of research for developing intelligent healthcare systems, rehabilitation technologies, wearable robotic platforms, and next generation assistive devices.

Recent advances in artificial intelligence (AI) and machine learning have significantly improved the performance of wearable sensor-based lower limb motion classification systems. Conventional machine learning algorithms, including decision trees, random forests, gradient boosting, and ensemble learning methods, have demonstrated high classification accuracy when combined with carefully engineered statistical and frequency domain features extracted from IMU sensor data [3]–[6], [10], [22], [26]. These approaches have been widely adopted because of their relatively low computational complexity, robustness, and suitability for structured sensor data. More recently, deep learning approaches have attracted considerable attention because of their ability to automatically learn hierarchical feature representations from wearable sensor signals, thereby reducing the dependence on manual feature engineering and improving recognition performance for complex locomotion activities [2], [9], [15], [25], [26], [28]. Although deep learning methods often provide excellent classification performance, they generally require larger datasets, higher computational resources, and increased training time, which may limit their applicability in resource-constrained wearable and edge-computing environments. Consequently, engineered biomechanical features derived from joint kinematics continue to remain attractive for practical wearable sensing applications because they provide improved interpretability, lower computational complexity, and direct correspondence with human movement patterns. Joint-angle-based feature representation therefore offers an effective and practically deployable solution for developing accurate and

Volume 15 Issue 10, October 2026

Fully Refereed | Open Access | Double Blind Peer Reviewed Journal

[www.ijsr.net](http://www.ijsr.net)

computationally efficient lower limb motion classification systems.

Despite these advancements, existing wearable motion classification studies primarily focus on improving recognition accuracy while placing comparatively less emphasis on the reproducibility, traceability, and deployment of machine learning models. In many reported studies, data preprocessing, feature engineering, model training, and evaluation are performed using isolated scripts with limited support for systematic experiment tracking, artifact management, or version control [16], [17], [19], [20], [25]. Furthermore, trained models, preprocessing pipelines, feature selection strategies, software dependencies, and experimental configurations are often not preserved in a structured and reusable manner, making it difficult to reproduce experimental results, compare model performance across different studies, or ensure consistent deployment across heterogeneous computing environments [18]–[20], [25]. As machine learning models become increasingly complex, these challenges significantly reduce research reproducibility, hinder collaboration among researchers, and complicate the transition of machine learning solutions from laboratory research to practical deployment. Consequently, there is a growing need for an integrated machine learning lifecycle that systematically manages data preprocessing, experiment tracking, artifact storage, model versioning, deployment, and maintenance within a unified framework.

Machine Learning Operations (MLOps) has recently emerged as an effective paradigm for managing the complete lifecycle of machine learning systems by integrating machine learning methodologies with modern software engineering practices [16], [17], [19], [20], [25]. MLOps enables systematic experiment tracking, version control, artifact management, reproducible execution, deployment automation, and continuous model maintenance, thereby improving transparency, reliability, and scalability throughout the machine learning development process. These capabilities are particularly important for wearable sensing applications, where multiple preprocessing stages, feature engineering techniques, machine learning models, and deployment environments must remain consistent to ensure reliable and repeatable performance. Despite the growing adoption of MLOps across various artificial intelligence applications, relatively few studies have explored its integration into wearable lower limb motion classification systems, particularly those employing biomechanically meaningful joint kinematic representations for activity recognition.

To address these challenges, this paper presents a reproducible Machine Learning Operations (MLOps) framework for lower limb motion classification using biomechanically meaningful joint-angle features obtained from an Xsens Awinda wearable inertial sensing system. The proposed framework integrates Xsens-derived joint-angle data processing, sliding window feature engineering, comparative machine learning model evaluation, experiment tracking, artifact management, and deployment within a unified end-to-end workflow. Hip, knee, and ankle joint angles exported by the Xsens system are used directly as biomechanical inputs, followed by statistical, derivative-based, and frequency-domain feature extraction to

generate discriminative representations of lower limb movement.

Multiple supervised machine learning algorithms, including XGBoost, Extra Trees, CatBoost, Ridge Classifier, and Histogram Gradient Boosting, are systematically trained, evaluated, and compared within an MLflow-based experiment management environment. Git is employed for source-code version control, MLflow manages experimental metadata and trained model artifacts, Docker provides a reproducible containerized execution environment, and Streamlit enables interactive inference, visualization, and artifact reuse. Unlike existing wearable activity recognition studies that primarily emphasize classification accuracy, the proposed framework integrates biomechanical joint-angle representation with an end-to-end MLOps lifecycle encompassing experiment tracking, artifact management, reproducible deployment, and interactive inference within a unified workflow. The resulting framework bridges the gap between machine learning research and deployment-ready wearable sensing applications while improving reproducibility, transparency, and practical usability.

The remainder of this paper is organized as follows. Section II reviews the related literature. Section III presents the proposed MLOps framework and methodology. Section IV describes the experimental setup and implementation details. Section V presents the experimental results and discusses the deployment framework. Finally, Section VI concludes the paper and outlines future research directions.

## 2. Related Work

Wearable inertial sensing has been widely adopted for lower limb motion analysis and human activity recognition because of its portability, low cost, and ability to continuously monitor human movement outside laboratory environments [1], [2], [7], [21], [22], [29], [30]. Public benchmark datasets, such as the smartphone-based human activity recognition dataset reported by Anguita et al. [8], have also supported the development and evaluation of wearable activity recognition algorithms. Existing studies have employed statistical, temporal, and frequency domain features extracted from IMU signals together with conventional machine learning algorithms such as decision trees, random forests, gradient boosting, and ensemble learning methods for activity classification [3]–[6], [10], [22], [26]. More recently, deep learning approaches have demonstrated improved recognition performance by automatically learning feature representations from wearable sensor data [2], [9], [15], [24]–[26]. Although these methods have achieved high classification accuracy, they primarily focus on algorithm development and activity recognition performance.

Machine Learning Operations (MLOps) has recently emerged as an important paradigm for improving the reproducibility, traceability, deployment, and maintenance of machine learning systems [16], [17], [19], [20], [25]. Technologies such as MLflow, Docker, and Git provide systematic support for experiment tracking, artifact management, software version control, and containerized deployment [11]–[13]. While these technologies have been successfully applied in general machine learning applications, their adoption within

wearable sensing and lower limb motion classification remains relatively limited. Most existing wearable activity recognition studies do not integrate experiment tracking, model versioning, artifact management, and deployment within a unified machine learning lifecycle.

Therefore, there remains a need for an end-to-end framework that combines biomechanically meaningful joint kinematic feature extraction with reproducible machine learning development, systematic experiment tracking, artifact management, and deployment. The proposed MLOps framework addresses these challenges by integrating joint-angle data processing, feature engineering, comparative model evaluation, MLflow-based experiment tracking, Docker-supported deployment, and interactive Streamlit-based inference within a unified workflow.

### 3. Proposed MLOps Framework

This section presents the proposed end-to-end Machine Learning Operations (MLOps) framework for lower limb motion classification using wearable IMU sensor data. The framework integrates joint-angle data processing, feature engineering, comparative machine learning model development, experiment tracking, artifact management, and deployment within a unified workflow. The overall architecture is designed to provide a systematic and reproducible machine learning lifecycle, enabling seamless transition from offline model development to deployment-ready inference. Figure 2 illustrates the complete workflow of the proposed framework.

### 4. Methods

This section presents the methodology adopted for developing the proposed Machine Learning Operations (MLOps) framework for lower limb motion classification. The methodology consists of eight major stages: data acquisition, joint-angle data processing, feature engineering, supervised machine learning model development, experiment tracking, model artifact management, deployment, and interactive inference. Figure 2 illustrates the overall workflow adopted in this study.

#### a) Experimental Study Design

The experimental framework was developed using an Xsens Awinda wearable inertial motion-capture system. Data were collected from five human participants while performing six lower-limb activities: Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown. The Xsens system directly provides joint-angle outputs, and the hip, knee, and ankle joint-angle channels were therefore used as the biomechanical input representation without independently recomputing the joint angles from segment orientations. Sliding-window segmentation was employed to capture temporal motion characteristics, followed by comparative evaluation of

multiple supervised learning algorithms. Figure 1 shows the sensor placement and representative data-collection procedure.



**Figure 1:** Experimental data collection: (a) IMU sensor placement; (b) IMU based activity data collection.

#### b) Sensor System and Data Acquisition

Lower-limb motion data were acquired using the Xsens Awinda wireless inertial motion-capture system at a sampling frequency of 60 Hz. Data were collected from five participants performing six activities: Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown. The Xsens processing pipeline directly provides joint-angle outputs in three rotational components for the monitored joints. Therefore, the machine-learning pipeline used the exported hip, knee, and ankle joint-angle signals rather than independently reconstructing joint angles from raw segment orientations.

#### c) Training and Testing Protocol

The extracted joint-angle feature dataset was divided into training and testing subsets using an 80:20 hold-out evaluation strategy. The training subset was used for model learning and parameter tuning, whereas the testing subset was reserved exclusively for final performance evaluation. Feature scaling, feature selection, and label encoding were fitted using only the training data and subsequently applied to the testing data to avoid preprocessing leakage. The reported test set consisted of 1,617 feature windows distributed across the six evaluated activity classes: Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown. The complete data collection involved five subjects. Model performance was assessed using Accuracy, Precision, Recall, Macro F1-score, and confusion-matrix analysis.

The 80:20 hold-out split was performed at the feature window level. Therefore, the reported results represent holdout window-level classification performance and should not be interpreted as subject-independent generalization.

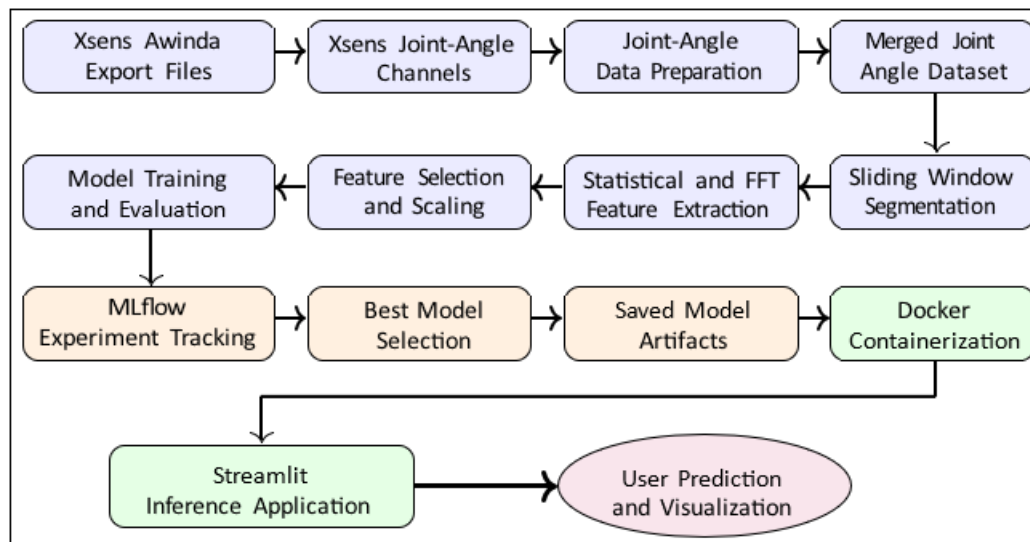


**Figure 2:** End-to-end pipeline of the proposed MLOps-driven lower limb motion classification framework. The pipeline summarizes data ingestion, joint-angle data processing, feature engineering, model development, experiment tracking, deployment, and inference

#### d) Overall Framework and Data Processing

The project is organized as a reproducible machine learning workflow. The workflow begins with raw IMU sensor files and ends with an interactive Streamlit inference interface. The complete end-to-end workflow is illustrated in Fig. 2. The implemented workflow comprises data preparation, feature engineering, model development, MLflow-based experiment tracking, artifact management, Docker-based packaging, and Streamlit-based deployment.

#### e) System Architecture

As illustrated in Fig. 3, the proposed MLOps framework is organized into two interconnected components: an offline training module and an online inference module. This modular architecture enables the separation of computationally intensive model development tasks from real-time prediction, thereby improving scalability, reproducibility, and deployment efficiency.

The offline training module performs data preparation, joint-angle data processing, feature engineering, supervised machine learning model development, and experiment tracking. Xsens Awinda export files provide hip, knee, and ankle joint-angle channels, which are first prepared and subsequently segmented using a sliding-window approach for statistical, derivative based, and frequency-domain feature extraction. The extracted features are used to train and evaluate multiple machine learning models, while MLflow records model configurations, hyperparameters, evaluation metrics, comparison plots, and trained model artifacts. Following comparative evaluation, the best-performing model together with the associated preprocessing artifacts, including the scaler, feature selector, label encoder, and feature definitions, are stored for deployment.

The online inference module utilizes the saved artifacts to perform activity recognition on previously unseen IMU sensor data. Users upload sensor datasets through a Streamlit based interface, where the same preprocessing pipeline used during training is automatically applied before prediction. The generated activity labels, joint-angle visualizations, and

downloadable prediction results are presented through an interactive interface, ensuring consistency between online prediction and online deployment/inference. By separating model development from inference while maintaining a common preprocessing and artifact management pipeline, the proposed architecture establishes a systematic, reproducible, and deployment ready framework for lower limb motion classification.

#### f) Reproducible Training and Deployment Workflow

Figure 4 illustrates the reproducible training and deployment workflow adopted in the proposed MLOps framework. The workflow is designed to support the complete machine learning lifecycle, including model development, experiment tracking, artifact management, reproducible execution, and deployment ready inference. By integrating software engineering practices with machine learning development, the framework ensures that every stage of model training and deployment can be systematically reproduced and consistently executed across different computing environments [16], [17], [19].

During model development, Git is employed to maintain source-code version control, enabling systematic tracking of code modifications and facilitating collaborative development [13]. MLflow serves as the experiment management platform, recording model configurations, hyperparameters, evaluation metrics, confusion matrices, comparison plots, and trained model artifacts for each experimental run [11]. This enables efficient comparison of multiple machine learning models and ensures complete traceability of experimental results.

Following model selection, Docker is used to package the trained model together with all required software dependencies into a containerized runtime environment, thereby eliminating platform-specific dependency issues and ensuring consistent execution across different systems [12]. Finally, Streamlit provides an interactive web-based interface through which users can upload IMU sensor datasets, perform activity prediction, visualize joint-angle signals and classification results, and download prediction outputs. The

integration of these components establishes a unified and reproducible workflow that bridges offline model development with deployment-ready wearable motion classification applications.

#### g) Joint-Angle Representation

The proposed framework utilizes biomechanically meaningful hip, knee, and ankle joint-angle signals as the primary representation for lower-limb motion classification. Joint-angle representation describes relative movement at anatomical joints and therefore provides a direct biomechanical basis for distinguishing locomotion patterns.

In this study, the Xsens Awinda system was used for motion acquisition. The Xsens processing software directly provides joint-angle outputs in three rotational components, and these exported joint-angle channels were used directly in the machine-learning pipeline. Consequently, the framework does not independently recompute hip, knee, and ankle joint angles from segment-orientation matrices.

For the machine-learning stage, the joint-angle channels from the left and right lower limbs are processed consistently across all six activities: Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown. Statistical, derivative-based, and frequency-domain descriptors are subsequently extracted from these signals. The reported classification metrics represent predictions produced from the *combined engineered feature*

*representation*; they are not separate accuracy values for each individual joint or rotational axis.

#### h) Feature Engineering

Following joint-angle data processing, the continuous joint angle signals are segmented into overlapping temporal windows to capture dynamic movement characteristics, which is a widely adopted approach in wearable human activity recognition [1], [7], [9], [21], [29], [30]. Window-based segmentation enables the extraction of representative statistical and spectral descriptors while preserving the temporal continuity of lower limb motion. For a window length  $W$  and stride length  $S$ , the  $i^{th}$  analysis window is represented as

$$X_i = \{x_i, x_{i+1}, \dots, x_{i+W-1}\} \quad (1)$$

In this study, a window length of  $W = 100$  frames and a stride length of  $S = 50$  frames are employed. Consequently, consecutive windows overlap by 50 frames, corresponding to a 50% overlap. This segmentation increases the number of feature windows while preserving local temporal continuity. Each segmented window is independently processed to extract discriminative features describing the statistical distribution, temporal variation, and frequency characteristics of lower limb motion. Statistical descriptors are computed to characterize the central tendency and variability of the joint-angle signals.

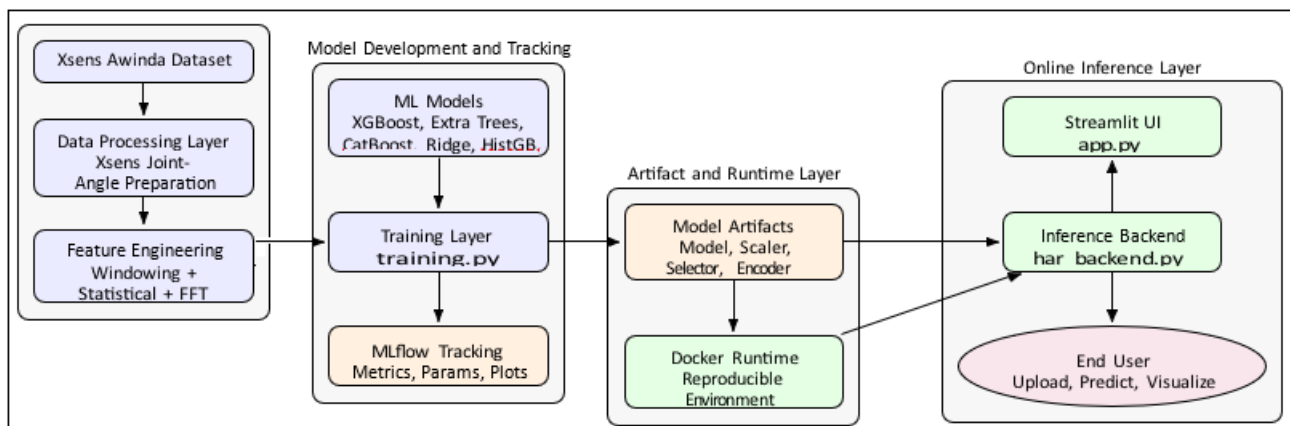


**Figure 3:** System architecture of the proposed lower limb motion classification platform showing offline data preparation, model development, artifact management, Docker packaging, and online inference components.

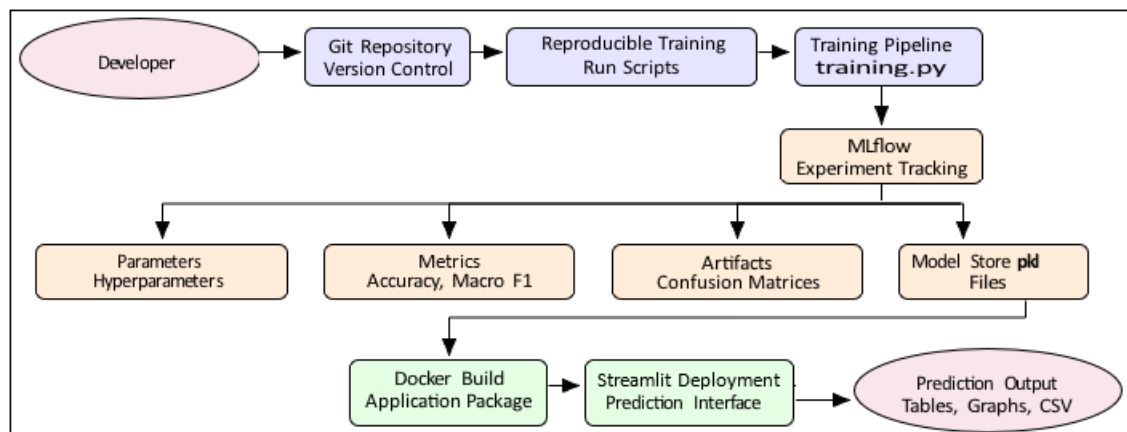


**Figure 4:** MLOps-oriented architecture used in this work. The workflow follows Git-based version control, offline training, MLflow experiment tracking, model artifact storage, Docker packaging, and Streamlit-based inference.

The mean value is calculated as

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (2)$$

while the standard deviation is obtained using

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (3)$$

The root mean square (RMS), which reflects the signal magnitude, is computed as

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (4)$$

To capture frequency-domain characteristics associated with different locomotion patterns, the Fast Fourier Transform (FFT) is applied to each analysis window

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (5)$$

The final feature vector consists of statistical, derivative based, and spectral descriptors, including mean, standard deviation, minimum, maximum, median, variance, percentiles, RMS, signal energy, skewness, kurtosis, signal range, first-order derivative features, FFT mean, FFT standard deviation, and FFT maximum. These complementary features provide a comprehensive representation of lower limb joint kinematics and serve as the input to the supervised machine learning models.

#### i) Machine Learning Model Development

The engineered joint-angle feature vectors were used to train and evaluate five supervised machine learning algorithms representing different classification strategies. The selected models include XGBoost, Extra Trees, CatBoost, Ridge Classifier, and Histogram Gradient Boosting. These algorithms were chosen because they provide a balance between predictive performance, computational efficiency, model interpretability, and deployment suitability for structured wearable sensor data. Ensemble learning models are capable of capturing complex nonlinear relationships within high-dimensional feature spaces, whereas Ridge Classifier serves as a linear baseline for evaluating the discriminative capability of the engineered joint angle features. Furthermore, all models were managed using the proposed MLflow-based experiment tracking framework, enabling systematic comparison of hyperparameters, evaluation metrics, trained models, and generated artifacts.

- **XGBoost:** XGBoost was selected to evaluate optimized gradient boosting on nonlinear joint-angle descriptors [4], [22]. The algorithm combines multiple weak learners through gradient boosting while incorporating regularization to reduce model complexity and improve generalization. Since coordinated lower limb motion involves nonlinear interactions among hip, knee, and ankle kinematics, XGBoost provides a strong baseline for learning complex decision boundaries.

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (6)$$

where  $l(\cdot)$  denotes the prediction loss function and  $\Omega(f_k)$  represents the regularization term controlling model complexity.

- **Extra Trees:** Extra Trees was selected because the engineered feature space contains numerous statistical and spectral descriptors extracted from multiple lower limb joints. The algorithm employs randomized split selection to improve model diversity while reducing overfitting [3], [22]. Owing to its robustness and computational efficiency for structured tabular data, Extra Trees was expected to perform well for lower limb motion classification and ultimately achieved the highest recognition performance in this study.

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_M(x)\} \quad (7)$$

where  $T_i(x)$  represents the prediction generated by the  $i^{\text{th}}$  randomized decision tree.

- **CatBoost:** CatBoost was included as an additional gradient boosting algorithm designed for structured tabular datasets [5], [26]. The model was evaluated to investigate whether an alternative boosting strategy could improve recognition of dynamic lower limb locomotion activities while maintaining strong generalization performance.

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (8)$$

where  $h_m(x)$  denotes the weak learner added at iteration  $m$ , and  $\eta$  is the learning rate.

- **Ridge Classifier:** Ridge Classifier was employed as a linear baseline to evaluate the separability of the engineered joint-angle feature space [10], [14], [26]. Although linear in nature, Ridge Classification provides an effective benchmark for assessing whether the extracted features are sufficiently discriminative without requiring nonlinear decision boundaries.

$$J(w) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \|w\|_2^2 \quad (9)$$

where  $\lambda$  denotes the L2 regularization parameter controlling model complexity.

- **Histogram Gradient Boosting:** Histogram Gradient Boosting was evaluated as an efficient tree-based ensemble learning algorithm optimized for structured tabular data [6], [10], [26]. By discretizing continuous feature values into histogram bins, the algorithm significantly reduces computational complexity while maintaining competitive classification performance, making it attractive for deployment-oriented wearable sensing applications.

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (10)$$

where  $h_m(x)$  denotes the weak learner added during iteration  $m$ , and  $\eta$  represents the learning rate.

#### j) Experiment Tracking and Training Workflow

The reproducible training workflow adopted in the proposed framework is illustrated in Fig. 5. The workflow integrates software version control, automated model training, experiment tracking, and artifact management within a unified MLOps pipeline. Source-code modifications are first maintained using Git to ensure version consistency and facilitate collaborative development [13]. The joint-angle data processing and feature engineering scripts are subsequently executed before training multiple supervised machine learning models. MLflow serves as the experiment management platform by recording hyperparameters,

evaluation metrics, confusion matrices, comparison plots, trained model artifacts, and execution metadata for every training run [11], [20]. This enables systematic comparison of different machine learning algorithms while preserving complete experiment traceability and

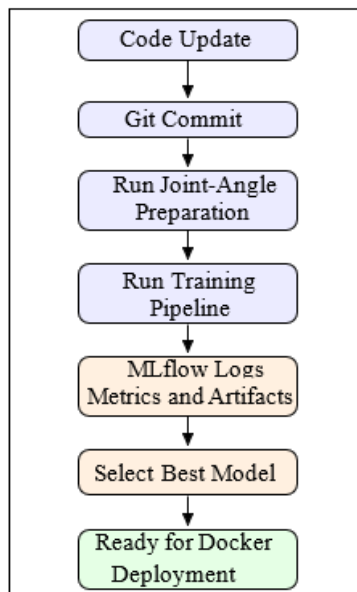


**Figure 5:** Reproducible model development workflow using Git, Python scripts, and MLflow

reproducibility. Following comparative evaluation, the best performing model together with the associated preprocessing artifacts is selected for deployment. The resulting workflow establishes a consistent transition from model development to deployment-ready inference while maintaining full reproducibility throughout the machine learning lifecycle [16]–[20].

#### k) Deployment Workflow

The deployment workflow adopted in the proposed framework is illustrated in Fig. 6. Following model selection, the trained classifier together with all preprocessing artifacts, including the scaler, feature selector, label encoder, and feature definitions, are packaged for deployment. Docker is employed to encapsulate the application and all software dependencies within a containerized runtime environment, thereby ensuring consistent execution across different operating systems and hardware platforms [12], [20].

The deployed application is implemented using Streamlit, which provides an interactive web-based interface for lower limb motion classification. Users upload IMU sensor datasets, after which the saved preprocessing pipeline is automatically applied before activity prediction. The application further provides visualization of joint-angle signals, predicted activities, downloadable prediction outputs, and optional animation generation. By preserving consistency between offline model development and online inference, the deployment workflow enables reliable and reproducible execution of the proposed lower limb motion classification framework.

## 5. Results and Discussion

The proposed MLOps framework was experimentally evaluated using lower limb joint-angle data obtained from wearable IMU sensors. The evaluation followed the end-to-end workflow presented in Fig. 2, encompassing joint-angle data processing, feature engineering, supervised machine learning model development, experiment tracking, artifact management, and deployment. Five supervised machine learning algorithms, namely XGBoost, Extra Trees, CatBoost, Ridge Classifier, and Histogram Gradient Boosting, were trained and systematically compared using the proposed MLflow based experiment management framework. Classification performance was evaluated using Accuracy and Macro F1-score, while confusion matrix analysis and feature importance were employed to investigate model behaviour and interpretability.

#### A. Comparative Model Performance

Table I summarizes the classification performance of all evaluated machine learning models. The experiments demonstrate that ensemble learning algorithms consistently outperform the linear baseline, indicating that lower limb motion classification is characterized by nonlinear relationships among the engineered joint-angle features.

**Table I:** ML flow Experiment Tracking Summary with Precision and Recall

| Model                | Accuracy (%) | Precision | Recall | Macro F1 Score |
|----------------------|--------------|-----------|--------|----------------|
| XGBoost              | 81.95        | 0.82      | 0.82   | 0.8674         |
| Extra Trees          | 99.59        | 0.98      | 0.98   | 0.9939         |
| CatBoost             | 94.40        | 0.95      | 0.95   | 0.9419         |
| Ridge Classifier     | 96.27        | 0.96      | 0.97   | 0.9739         |
| HistGradientBoosting | 72.61        | 0.73      | 0.72   | 0.7639         |

Among the evaluated classifiers, Extra Trees achieved the highest performance with an accuracy of 99.59%, weighted

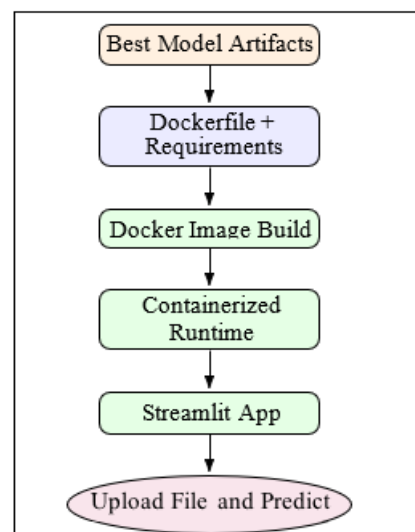


**Figure 6:** Model packaging and Streamlit deployment workflow.

precision of 0.98, weighted recall of 0.98, and a Macro F1score of 0.9939. Its superior performance can be attributed to the randomized tree construction strategy, which effectively captures complex nonlinear relationships among

statistical and frequency-domain joint-angle descriptors while reducing overfitting. Ridge Classifier also achieved competitive performance, suggesting that the engineered joint-angle features possess strong discriminative capability even under linear decision boundaries. CatBoost demonstrated robust recognition accuracy for structured feature vectors, whereas XGBoost and Histogram Gradient Boosting exhibited comparatively lower performance for the evaluated dataset. The Extra Trees classifier achieved the highest accuracy and Macro F1-score. It was therefore selected as the best model and saved along with the scaler, feature selector, label encoder, and feature names.

The values in Table I are overall activity-classification metrics obtained from the combined feature vector constructed from the available lower-limb joint-angle channels. They should not be interpreted as identical accuracy values for each hip, knee, ankle, or rotational component. Separate per joint/per-axis accuracies were not reported in the present evaluation.

### B. Feature Importance Analysis

To improve the interpretability of the proposed framework, feature importance analysis was performed using the Extra Trees classifier. Unlike overall accuracy metrics, feature importance provides insight into the contribution of individual biomechanical descriptors toward activity discrimination. Since the extracted feature space contains statistical, derivative-based, and frequency-domain descriptors from multiple lower limb joints, feature importance enables identification of the most influential movement characteristics.

The analysis indicates that descriptors related to signal variance, movement range, RMS, energy, and FFT-based features contribute most significantly to activity recognition. Hip, knee, and ankle joint kinematics collectively contribute to classification performance, demonstrating that coordinated lower limb movement provides a highly discriminative representation for locomotion analysis. Dynamic activities such as Walking, Stairs, SlopeUp, and SlopeDown are primarily characterized by variations in joint movement amplitude and frequency content, whereas the more static Sitting and Standing activities are characterized by lower signal variability and reduced movement range.

Figure 7 presents the relative importance of the most discriminative joint-angle features obtained from the Extra Trees classifier. Knee range emerged as the most influential feature, followed by hip variance, ankle RMS, knee FFT maximum, and hip energy. These findings indicate that both temporal and frequency-domain characteristics of hip, knee, and ankle motion contribute substantially to activity discrimination. The feature importance analysis demonstrates that the proposed joint-angle representation not only improves classification performance but also enhances model interpretability by directly

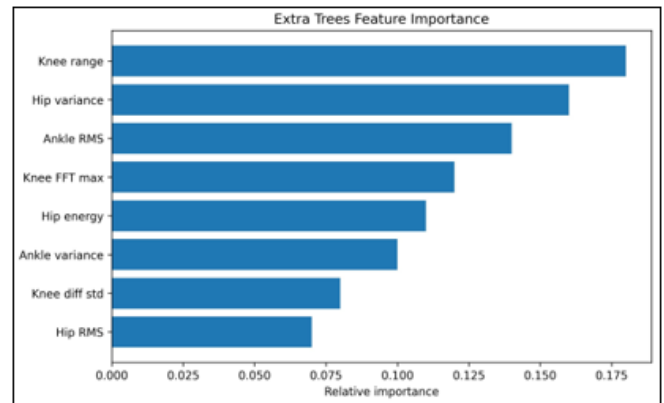


**Figure 7:** Feature importance obtained using the Extra Trees classifier. The most influential features include knee range, hip variance, ankle RMS, knee FFT maximum, and hip energy, indicating that biomechanically meaningful joint-angle descriptors play a significant role in lower limb motion classification relating important features to biomechanical movement characteristics.

### C. Confusion Matrix Analysis

Figures 8–12 present the confusion matrices obtained for the evaluated machine learning models, including XGBoost in Fig. 8, CatBoost in Fig. 9, Extra Trees in Fig. 10, Ridge Classifier in Fig. 11, and Histogram Gradient Boosting in Fig. 12. The updated confusion matrices include all six evaluated activity labels—Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown—using the class order defined by the label encoder. These visualizations provide a detailed understanding of class-specific prediction performance beyond overall classification accuracy.

The Extra Trees classifier exhibits nearly perfect classification across all locomotion activities, with negligible misclassification between activity classes. The few remaining errors primarily occur between *SlopeUp* and *SlopeDown* activities because both movements involve similar lower limb kinematic patterns and overlapping joint-angle trajectories. In contrast, XGBoost and Histogram Gradient Boosting exhibit comparatively larger confusion among dynamic locomotion classes, indicating that these models are less effective in capturing subtle biomechanical differences between similar movement patterns. Ridge Classifier and CatBoost demonstrate intermediate performance, correctly recognizing most activities while exhibiting occasional misclassification for biomechanically similar motions.

Overall, the confusion matrix analysis confirms that the proposed joint-angle representation provides highly discriminative features for lower limb motion classification and that the Extra Trees classifier most effectively exploits these features for robust activity recognition.

### D. Performance Visualization

Figure 13 shows that Extra Trees achieved the highest performance, followed by Ridge Classifier and CatBoost.

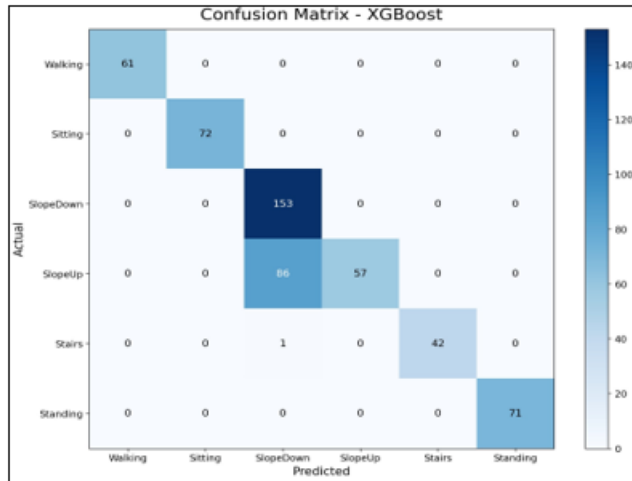


Figure 8: Confusion matrix of XGBoost

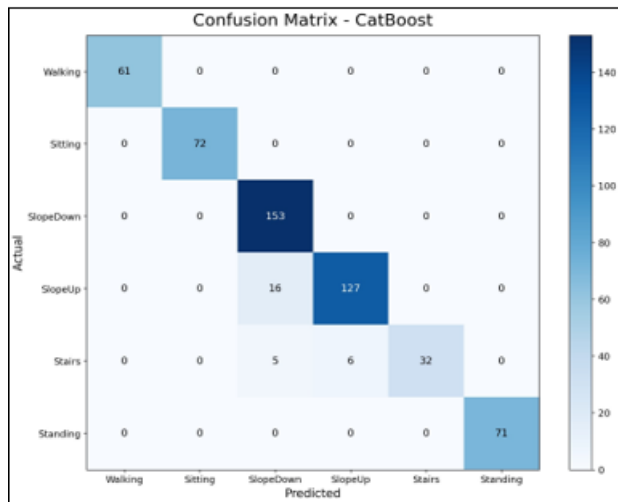


Figure 9: Confusion matrix of CatBoost

### E. Comparison with Existing Studies

Table II compares the proposed framework with representative wearable activity recognition and MLOps-related studies. Existing wearable sensing studies commonly focus on activity recognition performance, while MLOps studies focus on lifecycle automation and reproducible deployment. In contrast, the proposed work combines biomechanically meaningful joint-angle features, ensemble learning, MLflow-based experiment tracking, artifact management, Docker packaging, and Streamlit-based inference within a unified workflow.

### F. Deployment Validation

Figure 14 illustrates the developed Streamlit-based deployment interface used for lower limb motion classification. The application allows users to upload IMU datasets, visualize hip,

Table II: Comparison of the Proposed Framework with Existing Studies

| Study                   | Sensor/Input                     | Method                          | Main Focus                                      | MLOps Support | Deployment port | Sup- | Reported Performance                  |
|-------------------------|----------------------------------|---------------------------------|-------------------------------------------------|---------------|-----------------|------|---------------------------------------|
| Lara and Labrador [1]   | Wearable sensors                 | Conventional HAR methods        | Survey of wearable HAR                          | No            | No              |      | Not applicable                        |
| Wang et al. [2]         | Wearable sensor signals          | Deep learning models            | Sensor-based activity recognition               | No            | Limited         |      | Not directly comparable               |
| Ordoñez and Roggen [9]  | Multimodal wearable sensors      | CNN-LSTM                        | Deep wearable activity recognition              | No            | No              |      | High recognition performance reported |
| Kreuzberger et al. [17] | ML lifecycle components          | MLOps architecture              | MLOps concepts and lifecycle                    | Yes           | Yes             |      | Not activity-classification focused   |
| Proposed framework      | Xsens Awinda joint-angle outputs | Extra Trees with MLOps workflow | Lower limb motion classification and deployment | Yes           | Yes             |      | 99.59% accuracy                       |

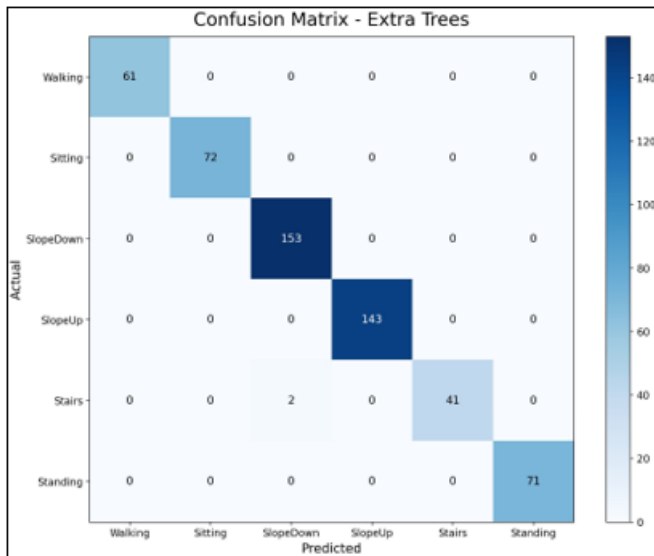


Figure 10: Confusion matrix of Extra Trees

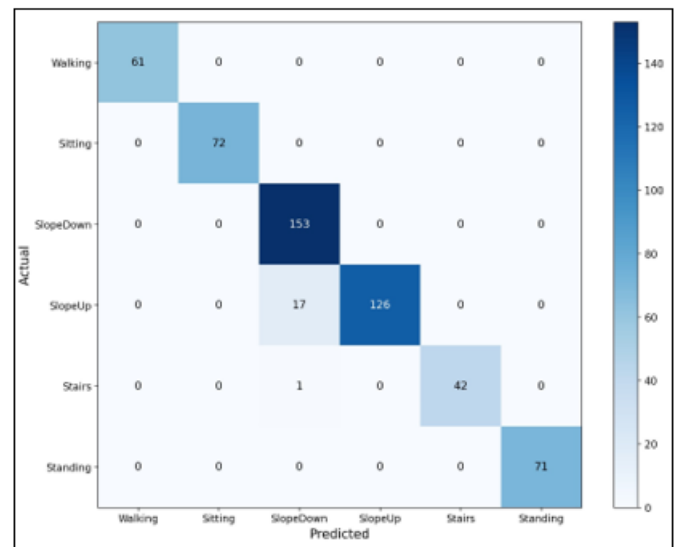


Figure 12: Confusion matrix of Ridge Classifier

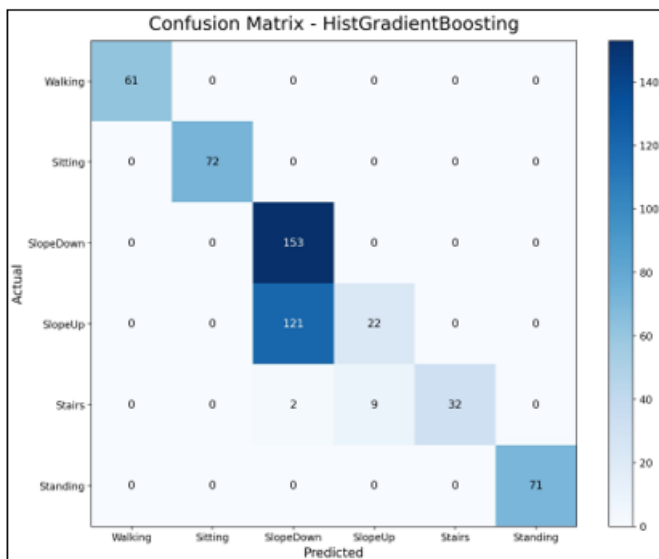


Figure 11: Confusion matrix of Hist Gradient Boosting

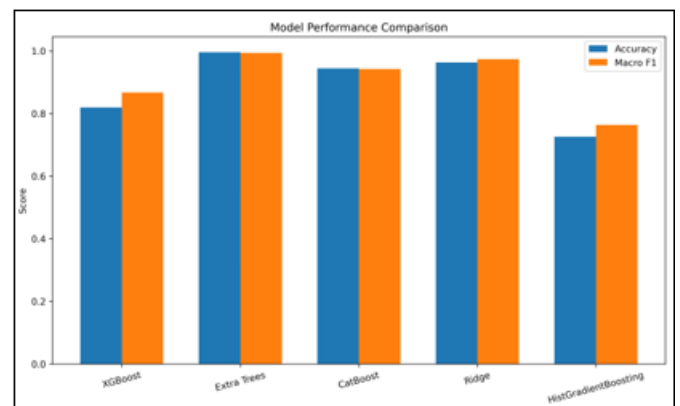
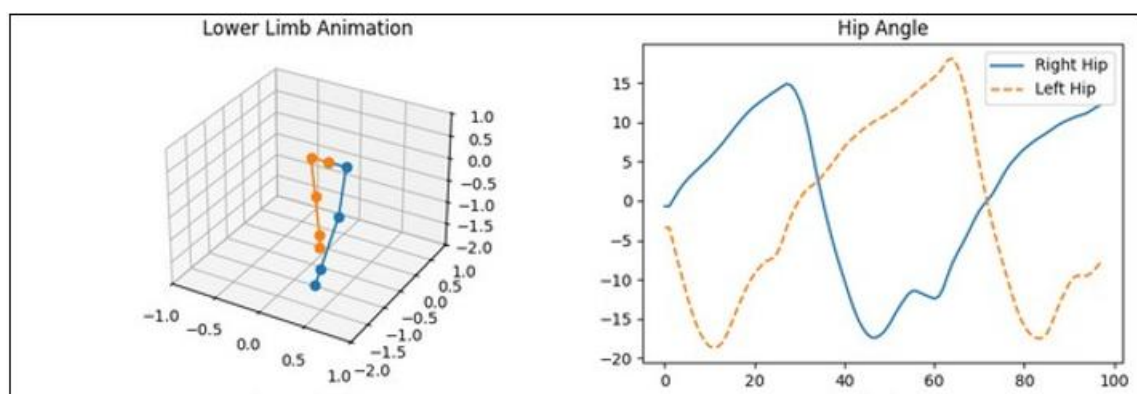


Figure 13: Combined Accuracy and Macro F1-score comparison of all evaluated models.



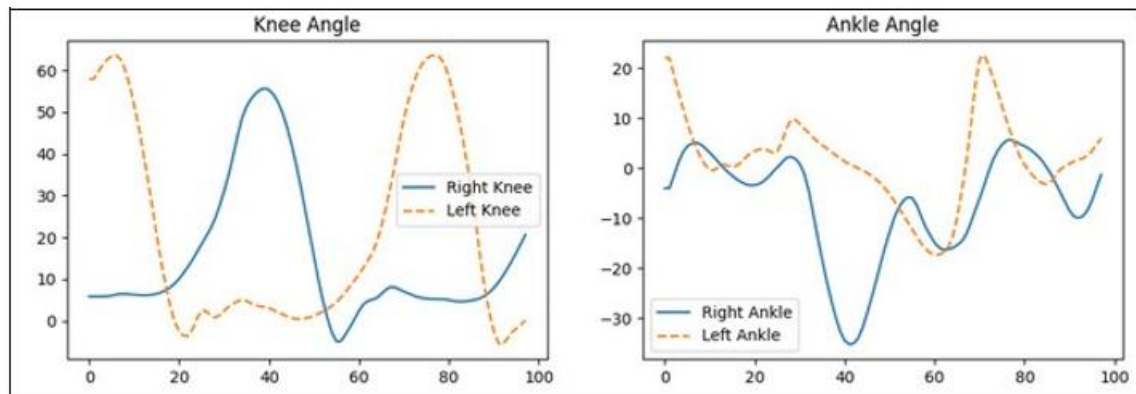


**Figure 14:** Streamlit-based deployment interface developed for the proposed MLOps framework

The application enables visualization of lower limb kinematics, 3D animation, activity prediction, and joint-angle plots for inference. knee, and ankle joint-angle trajectories, inspect lower limb animations, perform activity prediction, and download the generated results. The same preprocessing pipeline, trained model, scaler, feature selector, and label encoder employed during training are reused during inference, ensuring reproducibility between offline model development and online deployment.

Beyond model accuracy, the proposed framework demonstrates the practical advantages of integrating Machine Learning Operations (MLOps) within wearable sensing applications. MLflow was successfully employed to record hyperparameters, evaluation metrics, confusion matrices, comparison plots, and trained model artifacts for every experimental run, enabling systematic experiment management and reproducibility. Following comparative evaluation, the best-performing model together with the associated preprocessing artifacts, including the scaler, feature selector, label encoder, and feature definitions, was automatically prepared for deployment.

Docker was subsequently used to package the complete application and software dependencies into a containerized runtime environment, while Streamlit provided an interactive user interface for model inference. Users can upload IMU sensor datasets, visualize joint-angle signals, perform activity prediction, and download prediction results without requiring additional preprocessing. The deployment workflow therefore establishes a seamless transition from offline model development to deployment-ready inference while maintaining consistency between training and deployment environments.

### G. Discussion

The experimental results demonstrate that the proposed MLOps framework extends beyond conventional activity recognition pipelines by integrating reproducible model development, experiment tracking, artifact management, and deployment within a unified machine learning lifecycle. Unlike traditional wearable sensing studies that primarily emphasize classification accuracy, the proposed framework enables systematic comparison of multiple machine learning models while ensuring reproducibility throughout the complete development and deployment process.

The superior performance of the Extra Trees classifier indicates that randomized ensemble learning is well suited to the structured joint-angle feature representation used in this study. Furthermore, the high performance achieved by the Ridge Classifier suggests that the engineered biomechanical descriptors provide strong class separability even under linear decision boundaries. Recent studies have also demonstrated that optimized IMU configurations, lightweight deep learning architectures, synthetic data generation, and exoskeleton-oriented activity recognition can further improve the robustness and real-time applicability of wearable human activity recognition systems [22], [24]–[30].

Although the present study demonstrates high hold-out classification performance, several limitations remain. The current framework has been evaluated using wearable IMU data under controlled experimental conditions and does not yet incorporate multimodal sensing or embedded hardware deployment. In addition, the reported 80:20 evaluation must be interpreted according to the actual split unit used in the implementation. Overlapping windows from the same recording may be correlated; therefore, window-level random splitting does not establish subject-independent generalization. Participant- or trial-independent validation should be used in future work when sufficient independent recordings are available. Future work will therefore investigate subject independent validation, multimodal IMU–EMG fusion, optimized sensor placement, lightweight deep learning models, synthetic data augmentation, real-time locomotion transition prediction, and deployment on resource-constrained edge computing platforms, following recent advances reported in the literature [24]–[30].

## 6. Conclusion

This paper presented a reproducible Machine Learning Operations (MLOps) framework for lower limb motion classification using biomechanically meaningful joint-angle representations derived from wearable IMU sensor data. The proposed framework integrates joint-angle data processing, feature engineering, comparative machine learning model development, experiment tracking, artifact management, and deployment within a unified end-to-end workflow. By combining wearable sensing with modern MLOps practices, the framework establishes a systematic machine learning lifecycle that supports reproducible model development, efficient experiment management, and deployment-ready activity recognition.

The proposed framework was evaluated using Xsens Awinda lower-limb motion data collected from five subjects performing six activities, namely Walking, Sitting, Standing, Stairs, SlopeUp, and SlopeDown, with 1,617 feature windows used for the reported test evaluation. Five supervised machine learning models, namely XGBoost, Extra Trees, CatBoost, Ridge Classifier, and Histogram Gradient Boosting, were systematically trained, evaluated, and compared within the MLflow-based experiment tracking environment. Among the evaluated models, the Extra Trees classifier achieved the highest performance, obtaining an accuracy of 99.59%, weighted precision of 0.98, weighted recall of 0.98, and a Macro F1-score of 0.9939. The feature importance analysis further demonstrated that biomechanically meaningful joint-angle descriptors provide highly discriminative representations for lower limb motion classification while improving the interpretability of the proposed machine learning framework.

Beyond the classification performance, the principal contribution of this work is the end-to-end reproducible MLOps lifecycle rather than a claim that Extra Trees itself is novel. The framework integrates experiment tracking, artifact management, version control, containerized deployment, and interactive inference within a single workflow. The integration of Git, MLflow, Docker, and Streamlit enables systematic experiment management, reproducible execution, and seamless transition from offline model development to deployment-ready inference. Consequently, the proposed framework bridges the gap between machine learning research and practical wearable sensing applications by providing a scalable, transparent, and reproducible solution for lower limb motion classification.

Future work will focus on subject-independent validation using a larger and more diverse participant population, integration of multimodal IMU and EMG sensing for enhanced activity recognition, optimization of the inference pipeline for real-time execution, and deployment on embedded and edge-computing platforms. In addition, future studies will investigate the robustness and generalizability of the proposed framework under diverse locomotion conditions, long-duration wearable monitoring scenarios, and real-world deployment environments.

### Acknowledgment

The authors would like to express their heartfelt gratitude to the Director of DEBEL, DRDO, for providing the opportunity to undertake this research at the laboratory and for facilitating access to the state-of-the-art research facilities, equipment, and infrastructure that made this work possible.

The authors also acknowledge the valuable guidance, encouragement, and technical support received from the scientists, researchers, engineers, and technical staff of DEBEL, DRDO, whose expertise and assistance significantly contributed to the successful completion of this study.

Finally, the authors express their sincere appreciation to the research community working in the fields of wearable robotics, biomedical signal processing, artificial intelligence, and machine learning operations (MLOps), whose pioneering

contributions have provided valuable foundations and inspiration for the advancement of this research.

### References

- [1] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition Using Wearable Sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [2] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep Learning for Sensor-Based Activity Recognition: A Survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.
- [3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [5] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems*, 2018.
- [6] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [7] A. Bulling, U. Blanke, and B. Schiele, "A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 1–33, 2014.
- [8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," in *ESANN*, 2013.
- [9] F. Ordoñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors*, vol. 16, no. 1, 2016.
- [10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [11] M. Zaharia et al., "Accelerating the Machine Learning Lifecycle with MLflow," *IEEE Data Engineering Bulletin*, vol. 41, no. 4, pp. 39–45, 2018.
- [12] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," *Linux Journal*, no. 239, 2014.
- [13] S. Chacon and B. Straub, *Pro Git*, 2nd ed., Apress, 2014.
- [14] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, 3rd ed., O'Reilly Media, 2022.
- [15] F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2022.
- [16] D. Nogare and I. F. Silveira, "MLOps for Machine Learning Model Lifecycle Automation: A Systematic Literature Review," 2024.
- [17] D. Kreuzberger, N. Kuehl, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, vol. 11, pp. 31866–31879, 2023.
- [18] S. Garg, P. Pundir, G. Rathee, P. K. Gupta, S. Garg, and S. Ahlawat, "On Reproducible Training/Model Packaging for Automated Deployment of Machine

- Learning Models Using MLOps,” in *Proc. IEEE AIKE*, 2021.
- [19] M. Testi et al., “MLOps: A Taxonomy and a Methodology,” *IEEE Access*, vol. 10, pp. 63606–63618, 2022.
- [20] S. Wazir, G. S. Kashyap, and P. Saxena, “MLOps: A Review,” 2024.
- [21] H. Nematallah and S. Rajan, “Adaptive Hierarchical Classification for Human Activity Recognition Using Inertial Measurement Unit (IMU) Time-Series Data,” *IEEE Access*, vol. 12, pp. 52127–52149, 2024, doi: 10.1109/ACCESS.2024.3386351.
- [22] S. J. Murphy and R. V. Vitali, “Assessment of IMU Configurations for Human Activity Recognition Using Ensemble Learning,” *IEEE Access*, vol. 12, pp. 111433–111442, 2024, doi: 10.1109/ACCESS.2024.3440215.
- [23] W. Guo et al., “Human Activity Recognition via Wi-Fi and Inertial Sensors with Machine Learning,” *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3360490.
- [24] X. Ma, Y. Liu, X. Zhang, L. Masia, and Q. Song, “Real-Time Continuous Locomotion Mode Recognition and Transition Prediction for Human with Lower Limb Exoskeleton,” *IEEE Journal of Biomedical and Health Informatics*, vol. 29, no. 2, pp. 1074–1086, 2025, doi: 10.1109/JBHI.2024.3462826.
- [25] C. S. Son and W. S. Kang, “Temporal and Modality Awareness-Based Lightweight Residual Network with Attention Mechanism for Human Activity Recognition Using a Lower-Limb Exoskeleton Robot,” *IEEE Access*, 2025.
- [26] P. Tokas, V. B. Semwal, and S. Jain, “An Efficient Deep Learning Framework for Lower Limb Activity Recognition from Wearable Sensors in Knee Pathologies,” *IEEE Sensors Journal*, vol. 25, no. 24, 2025, doi: 10.1109/JSEN.2025.3628388.
- [27] H. Kim, K. Shi, H. Choi, D. W. Rha, and K. Kong, “Real-Time Spatiotemporal Gait Analysis Using a Hip-Assistive Wearable Robot,” *IEEE Access*, vol. 14, 2026, doi: 10.1109/ACCESS.2026.3668322.
- [28] Z. You et al., “Optimizing Sensor and Data Selection on Lower Limbs via Deep Learning for Real-Time Human Activity Recognition,” *IEEE Transactions on Human-Machine Systems*, Early Access, 2026, doi: 10.1109/THMS.2026.3651525.
- [29] A. Al-Hourani et al., “A New Framework for Human Activity Recognition for Wearable Gait Rehabilitation Systems,” *Biomechanics*, vol. 8, no. 2, 2025.
- [30] L. Uhlenberg et al., “SynHAR: Augmenting Human Activity Recognition With Synthetic Inertial Sensor Data Generated From Human Surface Models,” *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3513477.