Human Activity Recognition using CNN-BiLSTM

# 1. Problem Statement

The project aims to build a deep learning–based model to classify six human physical activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) using smartphone sensor data. Accurate recognition of human activities can enable fitness tracking, health monitoring, and context-aware applications.

# 2. Dataset Description

• Source: UCI Human Activity Recognition (HAR) dataset from smartphones.  
• Size: Training samples: 7,352; Testing samples: 2,947  
• Features: Accelerometer and gyroscope signals from smartphones (X, Y, Z axes).  
• 9 signals in total: body\_acc (x, y, z), body\_gyro (x, y, z), total\_acc (x, y, z).  
• Sampling: 128 timesteps per window, each window labeled with one of six activities.  
• Preprocessing: Data loaded from text files, normalized and reshaped, labels adjusted from 1–6 to 0–5.

# 3. Methodology

A hybrid CNN + BiLSTM model is used:  
- CNN front-end extracts spatial features from sensor signals.  
- BiLSTM captures temporal dependencies bidirectionally.  
- Layer Normalization and Dropout improve stability and reduce overfitting.  
- Fully connected layer outputs activity class probabilities.  
  
**Rationale**: CNN learns local patterns in sensor signals, while BiLSTM models long-term temporal dependencies and leveraging reverse sequence information via bidirectional LSTM, Stacked LSTM to learn hierarchal representations of the learned temporal patterns leading to robust classification.

# 4. Experimental Details

## 4.1 Architecture Description

|  |  |  |
| --- | --- | --- |
| Layer | Description | Output Shape |
| Conv1D | 9 → 64 channels, kernel=5, ReLU | (batch, 64, 128) |
| Conv1D | 64 → 64 channels, kernel=3, ReLU | (batch, 64, 128) |
| BiLSTM | hidden=64, 2 layers, bidirectional | (batch, 128, 128) |
| LayerNorm | Normalization | (batch, 128) |
| Dropout | p=0.25 | (batch, 128) |
| Linear | 128 → 6 classes | (batch, 6) |

## 4.2 Evaluation Metrics

• Loss Function: Cross-Entropy Loss  
• Metrics: Accuracy, Precision, Recall, F1 score

## 4.3 Dataset Split

• Training: ~70%  
• Testing: ~30%  
• Validation: from training batches

## 4.4 Implementation Details

• Optimizer: AdamW  
• Learning Rate: 0.0025 (with warmup + cosine annealing schedule)  
• Batch Size: 1500  
• Epochs: 300  
• Gradient Clipping: 5.0  
• Dropout: 0.25  
• Early stopping by saving best test accuracy

# 5. Results and Discussion

## 5.1 Results

Testing Accuracy: 91.89%

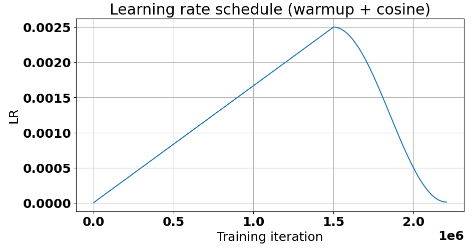
Precision: 92.03%

Recall: 91.89%

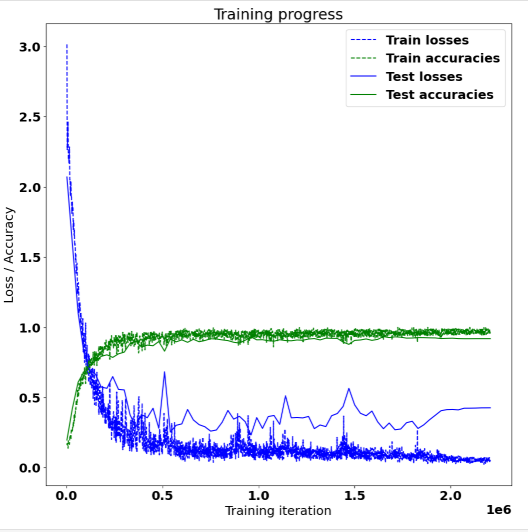
F1 score: 91.86%

## 5.2 Visualizations

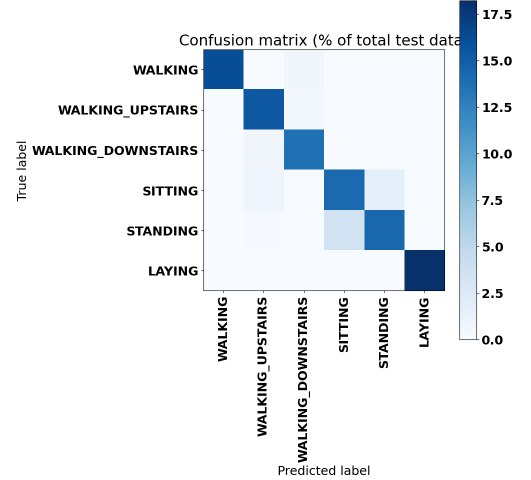
**Learning rate schedule visualization.**



**Training vs Validation Accuracy and Loss curves are shown below:**



**Confusion matrix heatmap (class-wise performance)**



## 5.3 Analysis

• CNN+BiLSTM architecture effectively captured both local and temporal dependencies.  
• The warmup + cosine learning rate schedule improved stability.  
• Misclassifications mainly occurred between similar activities like sitting vs standing.

## 5.4 Improvements

• Introduce attention mechanism to better focus on discriminative time steps.  
• Apply data augmentation (noise injection, window slicing, misclassified samples injection)  
• Optimize for lightweight models for real-time deployment on smartphones.