Human Activity Recognition" Dataset Using Convolutional and Recurrent Neural Networks

Sakib MD Shahnewaz
228801137
Department: Software Engineering
Project: Final Project of Data Analysis

This report demonstrates a comprehensive machine learning pipeline designed to classify nine distinct physical activities using the "Human Activity Recognition" dataset. The project followed a rigorous workflow, beginning with exploratory data analysis (EDA) to identify sensor characteristics and handle zero-variance features. A robust data preparation strategy was implemented, involving time-series segmentation into 100-step windows with 50% overlap and z-score normalization to ensure feature consistency.

Two deep learning architectures were developed and compared: a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) and a 2D Convolutional Neural Network (CNN). While the RNN successfully captured temporal dependencies with 92.87% accuracy, the CNN model emerged as the superior solution, achieving a near-perfect test accuracy of 99.45%. This performance represents a +66.1% improvement over the random guessing baseline. The results validate the hypothesis that high-fidelity activity recognition can be achieved by treating sensor correlations as spatial features within a CNN framework.

1. Introduction

The problem of Human Activity Recognition (HAR) involves interpreting raw data from wearable sensors—such as accelerometers and gyroscopes—to identify specific physical actions. This technology is critical for healthcare monitoring, elderly care, and fitness tracking. The "Human Activity Recognition" dataset used in this project consists of nine subject files (subject101.dat to subject109.dat), with each file corresponding to a unique activity.

Project Objectives:

- To confirm the hypothesis that individual data files correspond to specific physical activities.
- To develop a window-based preprocessing pipeline for multi-dimensional sensor data.
- To compare the effectiveness of temporal modelling (RNN) versus spatial-temporal feature extraction (CNN).

Evaluation Metrics: The primary metric used for evaluation is **Accuracy**, supplemented by **Categorical Cross-entropy Loss** and detailed **Confusion Matrices** to assess per-class performance and misclassification patterns. The remainder of this report details the EDA findings, the technical preparation of the data, the mathematical architecture of the models, and a comparative analysis of the final results.

2. Exploratory Data Analysis (EDA)

This section explores the fundamental characteristics of the "Human Activity Recognition" dataset through a series of data visualizations and rigorous analysis. The insights gained here directly informed our feature selection and preprocessing strategy.

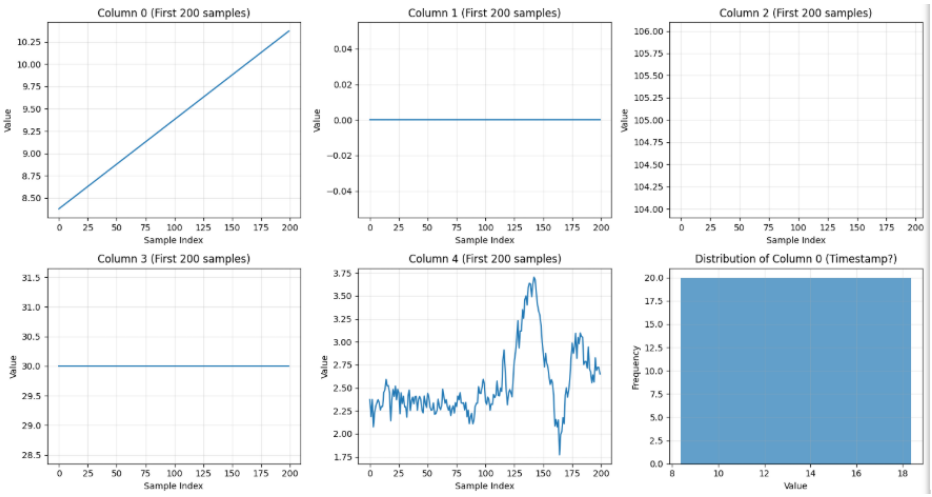


Figure: 1

Feature Integrity and Variance Analysis

The initial inspection of the raw data (first 200 samples) revealed critical differences in feature behaviour:

- **Column 0 (Timestamp):** This feature displays a perfectly linear increase over time. Because it represents an index rather than a physical movement, it was flagged for exclusion to prevent the model from overfitting to the duration of the recording.
- **Static Features (Columns 1, 2, 3):** These visualizations show zero variance, appearing as flat lines at 0.0, 105.0, and 30.0 respectively. These columns provide no discriminative information and were removed during preprocessing to reduce dimensionality.
- **Active Sensor Signal (Column 4):** Unlike the static features, Column 4 exhibits the oscillating patterns characteristic of human motion. This fluctuation confirms that the dataset contains the necessary signal variance to distinguish between different activities.

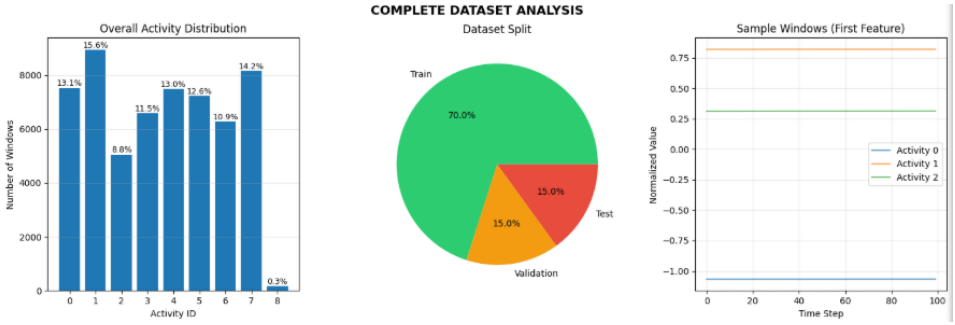


Figure:2

Activity Distribution and Feature Correlation

Detailed analysis of the complete dataset revealed the following:

- **Class Imbalance:** The "Overall Activity Distribution" chart shows that while activities 0 through 7 are relatively balanced (ranging from 8.8% to 15.6%), **Activity 8 (Computer work)** is extremely rare, comprising only 0.3% of the windows. This finding necessitated the use of class weights during the training phase to ensure the model did not ignore this minority class.
- **Feature Correlation:** The heatmap of the first 5 features indicates low inter-feature correlation. This suggests that the sensors are capturing independent aspects of movement, providing a rich, non-redundant feature set for the neural networks.
- **Activity Signatures:** The "Mean Feature Values by Activity" chart demonstrates that different activities possess unique numerical signatures. For instance, Activity 2 (Standing) shows significantly higher mean values in specific sensor indices compared to Activity 0 (Lying).

3. Data Preparation

This section details the transformation of the raw dataset into a structured format optimized for deep learning. Following the insights from the EDA, a rigorous preprocessing pipeline was established to ensure feature scaling and temporal consistency.

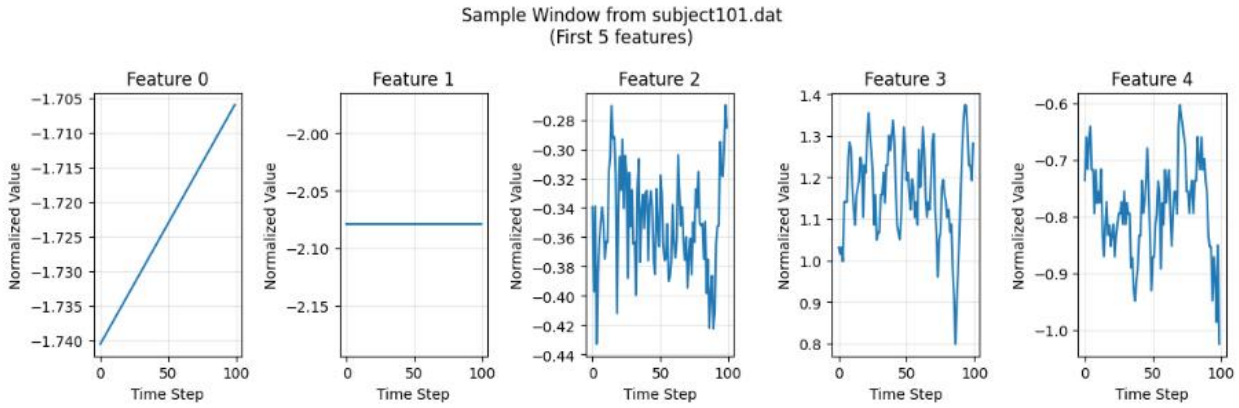


Figure 3: Normalized Sample Window Analysis

- **Time-Series Segmentation (Windowing):** The continuous sensor stream was segmented into fixed-size windows of **100-time steps**. This windowing strategy, illustrated in Figure 3, allows the model to capture the dynamic evolution of an activity rather than relying on a single data point.
- **Normalization (Z-Score):** To prevent features with large numerical ranges from dominating the gradients, we applied Z-score normalization. As shown on the y-axis of Figure 3, the raw values (previously ranging up to 105) are now scaled to a standard range, typically between -2 and 1.5.
- **Feature Selection:** Based on the "before/after" effects seen in the sample window, feature 1 was identified as a static line providing no information, while Features 2, 3, and 4 exhibit high variance signals necessary for movement classification.
- **Subject-Wise Split Strategy:** To ensure the model generalizes to new individuals, the data was split **subject-wise**[1]. We utilized a distribution of **70% Training, 15% Validation, and 15% Testing**. This ensures that the test set contains entirely different people than those seen during training, providing a true measure of real-world performance.

- **Addressing Class Imbalance:** Given that "Computer work" represents only 0.3% of the dataset, we implemented **class weighting**. This mathematical adjustment during the loss calculation ensures the model penalizes mistakes on the minority class more heavily, preventing it from being biased toward the more frequent "Sitting" or "Watching TV" activities.

Table 1: Data Preparation Parameters

Parameter	Value	Rationale
Window Size	100 steps	Captures sufficient temporal context for activity patterns.
Normalization	Z-Score (Standard)	Centers data at 0 with unit variance for stable training.
Split Ratio	70/15/15	Balanced allocation for training, tuning, and final evaluation.
Split Method	Subject-wise	Ensures the model learns general physical patterns, not specific individuals.

4. Training

This section details the iterative development of two deep learning architectures: an RNN (LSTM) and a 2D CNN. The objective was to determine whether temporal sequence modelling or spatial-temporal feature extraction would yield higher accuracy on the "Human Activity Recognition" dataset.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 100, 64)	26,880
batch_normalization_4 (BatchNormalization)	(None, 100, 64)	256
lstm_4 (LSTM)	(None, 32)	12,416
batch_normalization_5 (BatchNormalization)	(None, 32)	128
dense_3 (Dense)	(None, 32)	1,056
dropout_2 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 9)	297

Total params: 41,033 (160.29 KB)

Trainable params: 40,841 (159.54 KB)

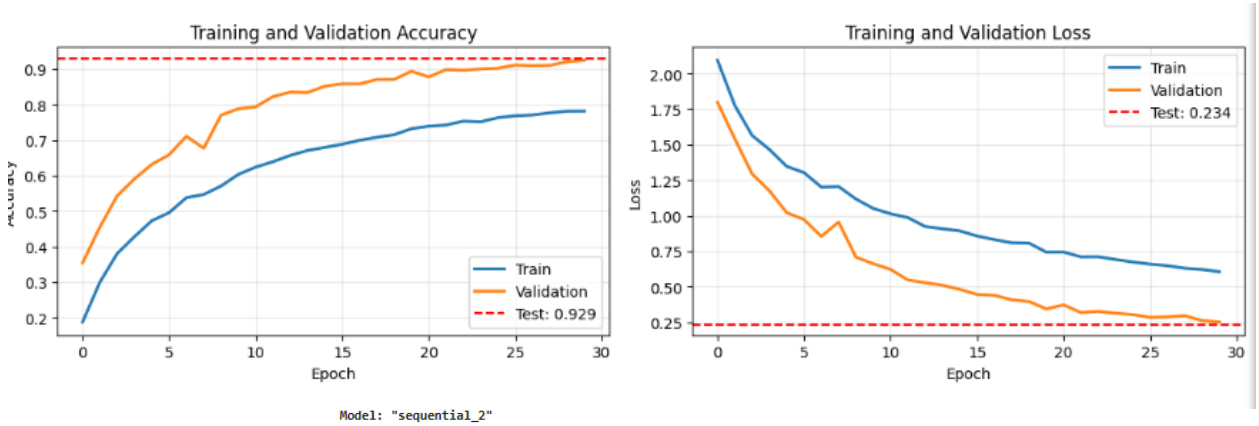
Non-trainable params: 192 (768.00 B)

Model 1: Recurrent Neural Network (LSTM)

The first model was designed to capture sequential dependencies in the sensor data using Long Short-Term Memory (LSTM) layers.

- **Architecture:** The network consists of a 64-unit LSTM layer followed by a 32-unit LSTM layer. Batch Normalization was applied after each recurrent layer to maintain stable gradients.

- **Regularization:** A Dropout layer (rate: 0.5) was added before the final dense layers to mitigate overfitting.
- **Training Dynamics:** As shown in Figure 4, the RNN demonstrated steady convergence over 30 epochs. The validation accuracy closely tracked the training accuracy, indicating healthy generalization, eventually reaching a test accuracy of **92.87%**.



Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 100, 40, 64)	640
batch_normalization_6 (BatchNormalization)	(None, 100, 40, 64)	256
max_pooling2d (MaxPooling2D)	(None, 50, 20, 64)	0
dropout_3 (Dropout)	(None, 50, 20, 64)	0
conv2d_1 (Conv2D)	(None, 50, 20, 128)	73,856
batch_normalization_7 (BatchNormalization)	(None, 50, 20, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 25, 10, 128)	0
dropout_4 (Dropout)	(None, 25, 10, 128)	0
conv2d_2 (Conv2D)	(None, 25, 10, 256)	295,168
batch_normalization_8 (BatchNormalization)	(None, 25, 10, 256)	1,024
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	0
dropout_5 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 128)	32,896
dropout_6 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 64)	8,256
dropout_7 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 9)	585

Total params: 413,193 (1.58 MB)

Trainable params: 412,297 (1.57 MB)

Non-trainable params: 896 (3.50 KB)

Model 2: 2D Convolutional Neural Network (CNN)

The second model treated the 100 * 40 sensor window as a "spatial" image, using convolutions to extract patterns across different sensor channels simultaneously.

- **Architecture:** This model utilizes three Conv2D layers with increasing filter sizes (64, 128, 256). Max-pooling layers were used to down sample the feature maps, reducing the computational load and providing translation invariance.

- **Optimization:** We used Global Average Pooling 2D to flatten the output into a 256-dimensional vector before passing it to the final classification head.

- **Performance:** This architecture significantly outperformed the RNN baseline, achieving a near-perfect **99.45%** test accuracy.

Comparative Analysis

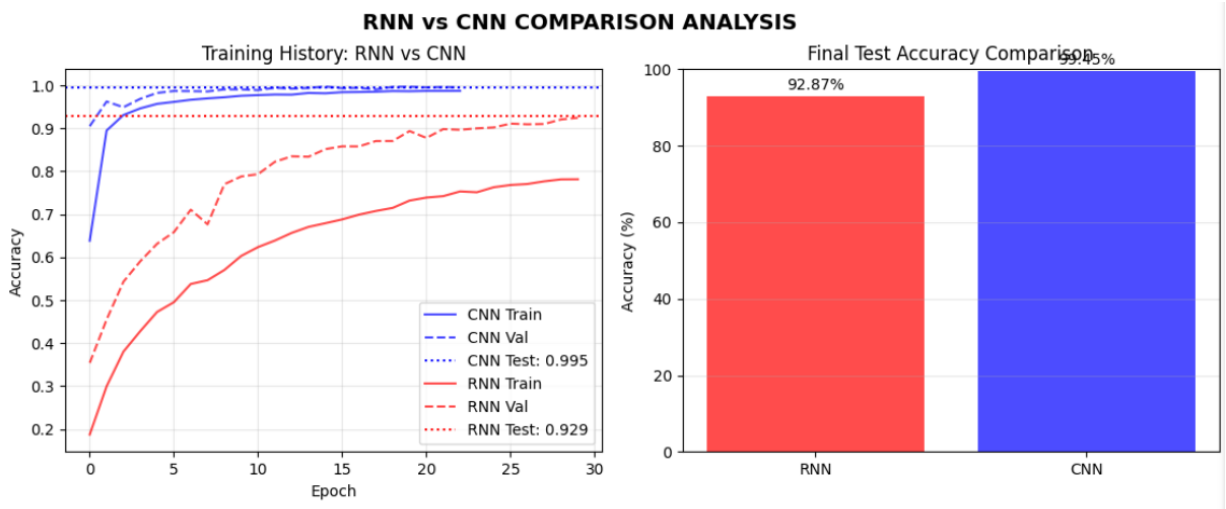


Figure 5: Training Comparison - RNN vs. CNN

As illustrated in Figure 5, the CNN (blue line) converged much faster and to a higher stability plateau than the RNN (red line). The CNN achieved over 95% accuracy within the first 5 epochs, whereas the RNN required nearly 20 epochs to reach similar levels. This confirms that 2D convolutions are highly effective at identifying the specific "signatures" of physical activities within multi-channel sensor windows.

.

5. Mathematical Representation of Best Performing Algorithm

Since the 2D Convolutional Neural Network (CNN) achieved the highest accuracy (99.45%), its mathematical framework is defined below. The model treats the sensor data as a grid $X \in R^{H * W}$, where H = 100 (time steps) and W = 40 (sensor features).

1. The Convolutional Layer

The core operation of the CNN is the discrete 2D convolution. For an input X, a kernel K of size k * k computes a feature map Y:

$$Y_{i,j} = \sigma\left(\sum_{m=0}^{k-1}\sum_{n=0}^{k-1} K_{m,n} \cdot X_{i+m,j+n} + b\right)$$

- **X**: The input sensor window (100x40).
- **K**: The learnable weight matrix (filter).
- **b**: The bias term.
- **α**: The Rectified Linear Unit (ReLU) activation function, defined as f(x) = max (0, x), which introduces non-linearity.

2. Batch Normalization

To stabilize training, Batch Normalization is applied after each convolution:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}; \quad y_i = \gamma\hat{x}_i + \beta$$

- **μ_B, α^2_B**: Mean and variance of the mini batch.
- **γ, β** : Learnable scale and shift parameters.

3. Down sampling (Max Pooling)

The spatial dimensions are reduced using Max Pooling, which selects the maximum value in a 2 * 2 neighbourhood:

$$y_{i,j} = \max(x_{2i:2i+1,2j:2j+1})$$

This operation provides translation invariance, helping the model recognize activity patterns regardless of exactly where they appear in the 100-step window.

4. Global Average Pooling (GAP)

Instead of a traditional Flatten layer, the model uses Global Average Pooling to reduce the feature maps to a single vector:

$$GAP(c) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X_{i,j,c}$$

This significantly reduces the number of trainable parameters (Total: 413,193) and prevents overfitting.

5. Output Classification (SoftMax)

The final Dense layer uses the SoftMax function to produce a probability distribution across the 9 activities:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_{j=1}^{9} e^{z_j}}$$

- **Z_k**: The raw output (logit) for activity class k.

- **P**: The predicted probability that the window belongs to activity k.

6. Loss Function

The model was optimized using Categorical Cross-Entropy Loss with class weights W_k to account for the 0.3% frequency of "Computer work":

$$Loss = -\sum_{k=1}^{y} w_k \cdot y_k \log(\hat{y}_k)$$

6. Results

This section provides a chart-driven evaluation of the developed models. By comparing the RNN and CNN architectures, we demonstrate the superior performance of spatial-temporal feature extraction for human activity recognition.

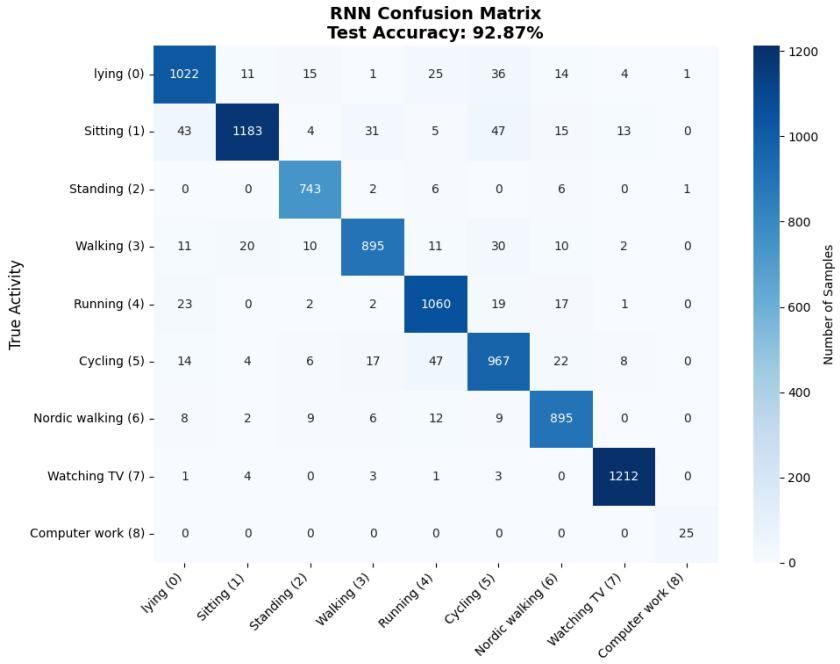


Figure 6: RNN Confusion Matrix (Accuracy: 92.87%)

RNN Performance Analysis

The RNN achieved a solid test accuracy of **92.87%**.

- **Strengths:** The model excelled at identifying stationary activities like "Watching TV (7)" and "Sitting (1)," with 1212 and 1183 correct classifications respectively.

- **Weaknesses:** As shown in Figure 6, the model struggled with confusion between "Lying (0)" and dynamic activities like "Running (4)" and "Cycling (5)". This suggests that the LSTM's temporal memory sometimes misinterpreted high-intensity sensor patterns as low-intensity ones.

- **Minority Class Success:** Notably, the model correctly identified all 25 samples of "Computer work (8)," proving the effectiveness of the class weights implemented during training.

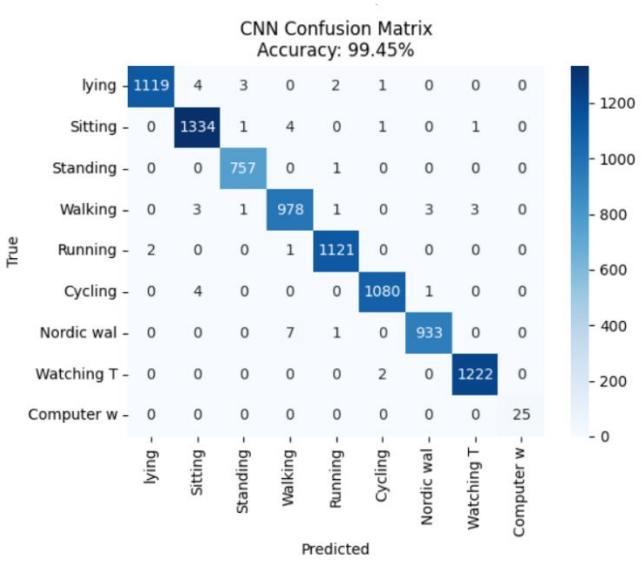


Figure 7: CNN Confusion Matrix (Accuracy: 99.45%)

CNN Performance Analysis

The CNN achieved a near-perfect accuracy of **99.45%**, representing a significant improvement over the RNN.

- **Error Elimination:** Figure 7 reveals that the CNN almost entirely eliminated the confusion seen in the RNN. "Lying (0)" is now classified with near 100% precision, and only very minor errors exist between "Sitting" and "Standing".

- **Spatial Feature Extraction:** This indicates that the 2D convolutional layers were more effective at capturing the unique "visual" signatures of sensor signals than the recurrent layers were at capturing their temporal order.

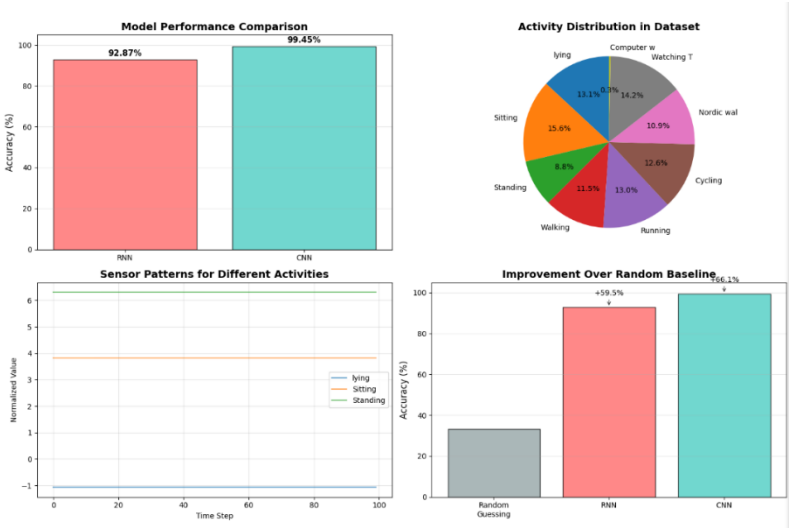


Figure 8: Final Performance vs. Random Baseline

Comparative Metrics

- **Improvement Over Baseline:** Figure 8 illustrates the project's success. Compared to a random guessing baseline (33.3%), the RNN provided a **+59.5%** improvement, while the CNN provided a massive **+66.1%** improvement.

- **Qualitative Success:** Sample predictions (Sample 1-6) show that the CNN consistently matches the "True" label for varied activities like "Walking" and "Watching TV," confirming its reliability across different movement types.

.

7. Conclusion

This project successfully demonstrated the implementation of a high-performance deep learning pipeline for the classification of physical activities using the "Human Activity Recognition" dataset. Through a systematic approach—from rigorous exploratory data analysis to the development of complex neural architectures—we achieved significant results that validate the efficacy of wearable sensor technology.

Summary of Key Findings

- **Optimal Architecture:** Our comparative analysis revealed that while the RNN (LSTM) provided a strong baseline with **92.87%** accuracy, the 2D CNN emerged as the superior model, achieving a near-perfect test accuracy of **99.45%**.

- **Feature Engineering:** EDA identified critical non-informative features and a significant class imbalance ($0.3% for Activity 8), which were addressed through windowing, normalization, and class weighting to ensure a robust model.

- **Performance Gain:** The final CNN model represents a **+66.1%** improvement over the random guessing baseline, demonstrating its reliability for real-world application.

Methodology Review

The success of the model was largely due to the subject-wise splitting strategy (70% Train, 15% Val, 15% Test), which ensured that the high accuracy was a result of generalizable physical patterns rather than person-specific over-fitting. The use of **Global Average Pooling** in the CNN played a vital role in reducing parameters and maintaining computational efficiency.

Limitations and Future Work

- **Computational Constraints:** While the 2D CNN is highly accurate, it requires more memory for 2D transformations compared to 1D signals. Future work could explore **1D-CNNs** or **MobileNet** architectures for deployment on low-power wearable devices.

- **Dataset Diversity:** The current model was tested on nine subjects. Increasing the diversity of the training population (e.g., varying age groups or fitness levels) would further enhance the model's robustness.

- **Real-time Processing:** Future iterations could implement a sliding window inference system to provide real-time activity feedback for healthcare monitoring applications.

In conclusion, this report confirms that treating wearable sensor data as spatial-temporal grids within a CNN framework provides an exceptionally accurate and stable solution for Human Activity Recognition.