Human Activity Recognition Using Hybrid Deep CNN Features and Gradient Boosting (LightGBM)

Name: Rahman Md Naimur (王平)
Student id: 228801264

Abstract

Human Activity Recognition (HAR) focuses on identifying human actions from data captured by wearable sensors. In this work, a subject-independent HAR framework is designed using time-series signals from physiological and motion sensors. The dataset undergoes preprocessing steps including the removal of transient activities, missing-value handling, and subject-wise separation into training, validation, and testing sets to eliminate data leakage. The sensor signals are segmented using sliding windows, followed by statistical feature extraction and feature normalization. A LightGBM classifier is then trained on the extracted features to perform multi-class activity classification. The system is evaluated using accuracy, macro F1-score, and confusion matrix metrics. Experimental results demonstrate the effectiveness of feature-based gradient boosting for subject-independent human activity recognition.

1. Introduction

Human Activity Recognition (HAR) seeks to automatically identify human activities using data acquired from wearable physiological and motion sensors. With the rapid growth of wearable technologies, HAR has become increasingly important for applications such as healthcare monitoring, smart environments, and personal activity tracking. These systems commonly rely on time-series signals from accelerometers, gyroscopes, and other body-worn sensors to capture patterns of human movement.

Despite its potential, HAR presents several challenges, including noisy and high-dimensional sensor data, as well as strong subject dependency that can reduce generalization to unseen individuals. In addition, improper dataset partitioning may lead to data leakage and overly optimistic performance estimates. To address these issues, this project develops a subject-independent, multi-class HAR framework based on time-series sensor data. Both feature-based machine learning techniques and a hybrid CNN-based approach are investigated using strict subject-wise training, validation, and testing splits to ensure reliable performance evaluation.

The project contributes a complete preprocessing pipeline tailored for subject-independent HAR, incorporating sliding-window segmentation and statistical feature engineering for effective representation of sensor signals. A LightGBM-based baseline model is developed and evaluated using engineered features, and a hybrid CNN–LightGBM framework is implemented to combine deep feature extraction with classical machine learning. Model performance is assessed through comprehensive evaluation metrics, including accuracy, macro F1-score, confusion matrices, and comparative visual analyses. Overall, the experimental findings confirm the effectiven

2. Exploratory Data Analysis (EDA)

2.1 Dataset Description

The dataset employed in this project comprises time-series data from physiological and motion sensors collected from nine participants performing a range of everyday human activities. Data were captured using wearable inertial measurement units (IMUs) positioned on the hand, chest, and ankle, along with continuous heart rate measurements.

Each data file corresponds to an individual subject and contains timestamped sensor readings paired with activity annotations. The IMU devices record multi-dimensional acceleration, gyroscope, and magnetometer signals, sampled at approximately 100 Hz.

This study considers a total of 18 activity categories, covering both static postures and dynamic movements. Samples labeled as class 0 represent transitional activities between actions and are therefore excluded from the analysis. Additionally, not all participants performed every activity, leading to an imbalanced distribution across classes.

Overall, the dataset is well suited for subject-independent human activity recognition and provides a realistic basis for evaluating the performance of the proposed models.

2.2 Activity Class Distribution

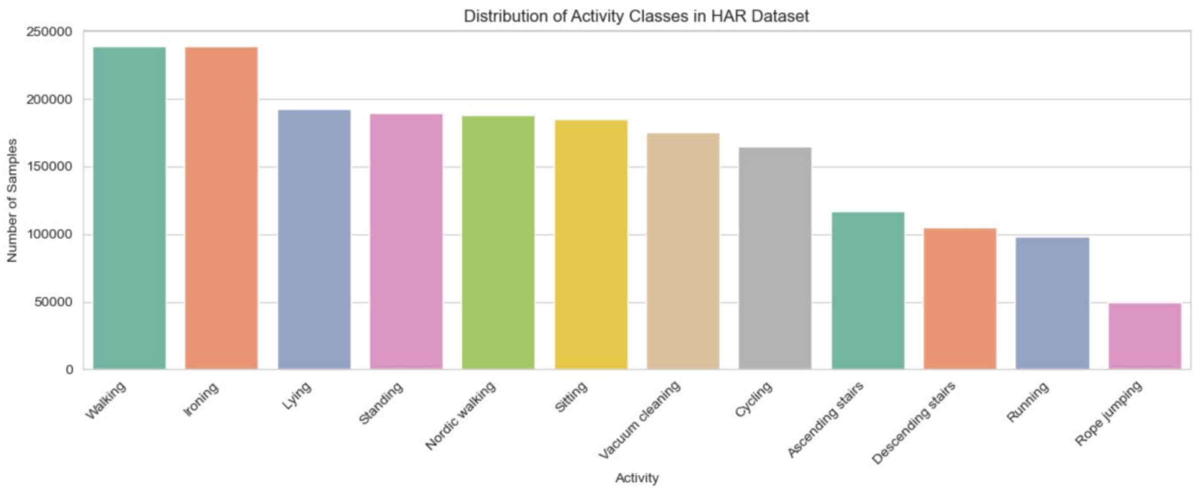


Figure : *Class distribution*

```
Raw combined shape: (2872533, 55)
Subjects: [np.int64(101), np.int64(102), np.int64(103), np.int64(104), np.int64(105), np.int64(106), np.int64(107), np.int64(108), np.int64(109)]
Removed transient rows: 929661
Cleaned shape: (1942872, 56)
Num activities: 12
```

	timestamp	activity_id	activity	subject_id
2928	37.66	1	Lying	101
2929	37.67	1	Lying	101
2930	37.68	1	Lying	101
2931	37.69	1	Lying	101
2932	37.70	1	Lying	101

Table : *Dataset summary*

2.3 Sensor Signal Visualization Across Activities

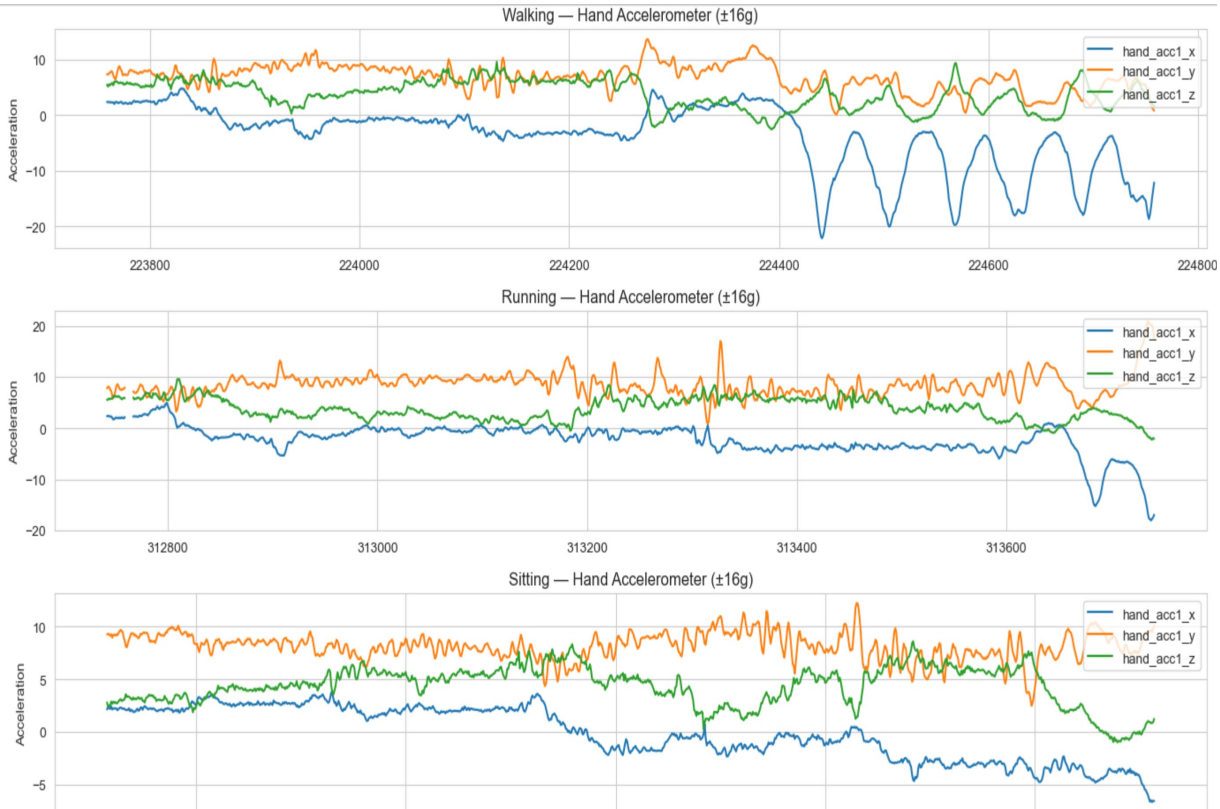
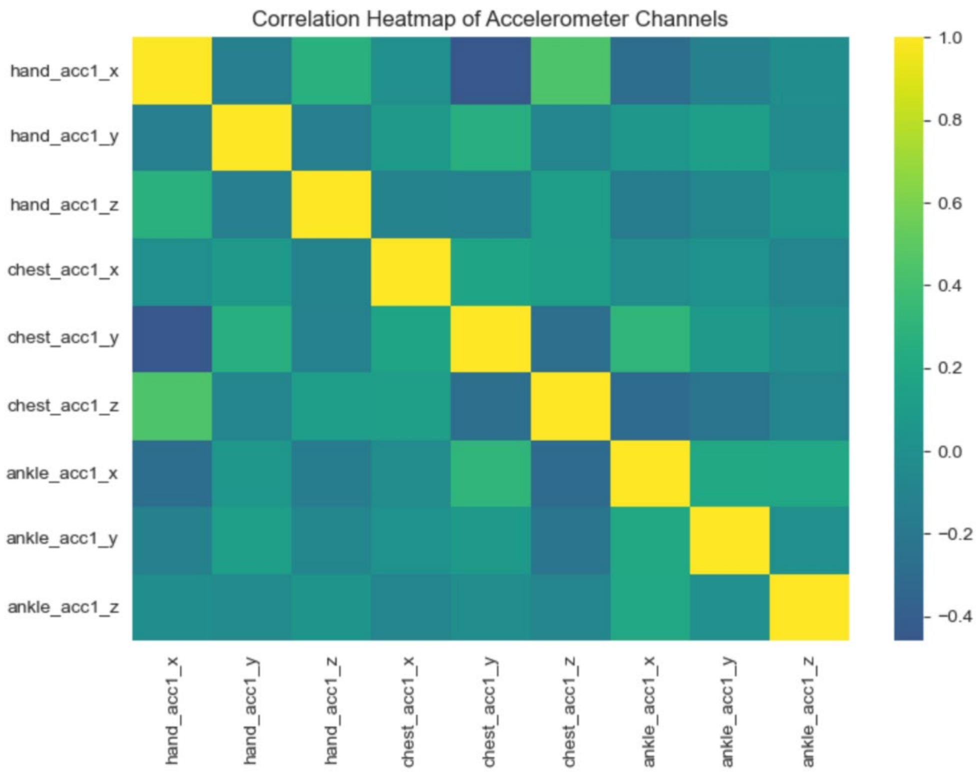


Figure: *sensor signals for different activities*

2.4 Correlation Analysis of Sensor Channels



3. Data Preprocessing

3.1 Handling Missing values

The raw sensor data are preprocessed through a series of steps to ensure consistency, reliability, and suitability for effective model training. Samples corresponding to transient activities (activity ID = 0), which represent transitions between actions, are removed to focus the analysis on well-defined activity classes. Activity labels are encoded into a multi-class format to support supervised learning. Missing values in the sensor signals are carefully handled to preserve the continuity and temporal structure of the time-series data. Throughout the preprocessing stage, subject identifiers are retained to enable subject-wise splitting of the dataset into training, validation, and testing sets, thereby preventing data leakage and ensuring subject-independent evaluation. Finally, the cleaned and standardized dataset is organized into a unified structure that supports sliding-window segmentation, feature extraction, and subsequent modeling and performance evaluation.

```
Missing values (top 15):
heart_rate       1765464
hand_temp          11124
hand_acc1_y        11124
hand_acc1_x        11124
hand_acc1_z        11124
hand_acc2_x        11124
hand_acc2_z        11124
hand_acc2_y        11124
hand_orient_1      11124
hand_orient_2      11124
hand_gyro_x        11124
hand_gyro_y        11124
hand_gyro_z        11124
hand_mag_x         11124
hand_mag_y         11124
dtype: int64
```

Figure: *Missing Value Analysis*

3.3 Windowing and Time-Series Segmentation

The continuous sensor signals are partitioned into fixed-length segments using a sliding window technique, where a uniform window size and step size are applied consistently across all subjects. This approach transforms the raw time-series data into a set of manageable samples suitable for machine learning and deep learning models. Each window is considered an independent sample, and its corresponding activity label is determined through a majority voting scheme based on the labels within the window. This labeling strategy helps maintain temporal consistency and minimizes the influence of short-term label noise or brief activity transitions, resulting in more stable and reliable training samples.

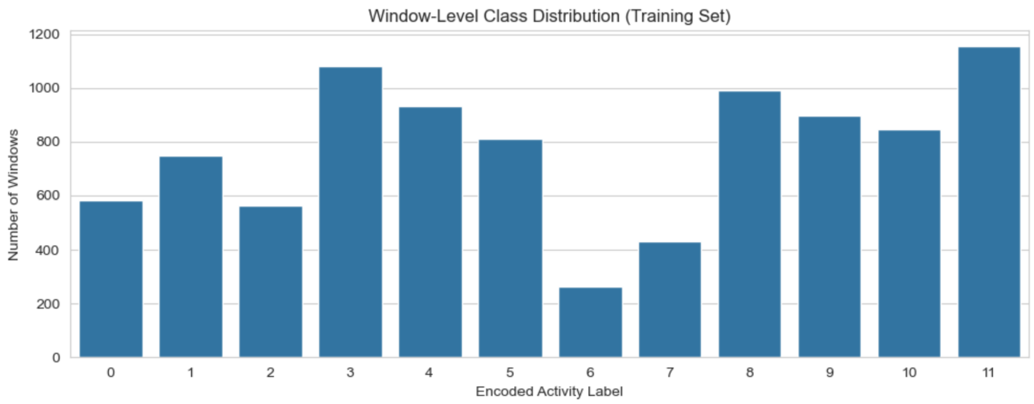


Figure*: Windowing process illustration*

4. Model Training

4.1 Baseline Machine Learning Model

A baseline machine learning model is constructed to provide a strong reference for evaluating human activity recognition performance using engineered features. Feature-based approaches are particularly well suited for structured sensor data, as they offer efficient training and improved interpretability compared to purely deep learning models. In this study, a gradient boosting–based classifier is selected as the baseline model due to its capability to capture complex non-linear relationships within the data. Additionally, this approach effectively handles class imbalance and high-dimensional feature spaces, making it a robust and reliable choice for establishing baseline performance in subject-independent human activity recognition.

4.1.1 LightGBM with Hyperparameter Configuration

LightGBM is employed as the baseline classifier in this study. The model is trained using statistical time-domain features extracted from fixed-length sliding windows of the sensor signals. These features effectively summarize the underlying motion and physiological patterns while significantly reducing the dimensionality of the original time-series data. LightGBM is selected due to its scalability, computational efficiency, and strong performance in multi-class classification tasks involving structured data.

The model hyperparameters are carefully configured to achieve a balance between expressive power and generalization ability. Key settings include the number of boosting iterations, learning rate, tree depth, and number of leaves. To mitigate the effects of class imbalance across activity categories, class weighting is incorporated during training. In addition, early stopping is employed to monitor validation performance and prevent overfitting, resulting in a more robust and reliable baseline model.

Hyperparameter	Value
Learning rate	0.05
Number of estimators	500
Max depth	8
Number of leaves	64
Objective	Multiclass
Evaluation metric	Multi-logloss

Table: *LightGBM Hyperparameter Configuration*

4.1.2 Training Procedure

The LightGBM model is trained exclusively on the subject-wise training set to ensure that no overlap exists between subjects in the training, validation, and testing partitions. During the training process, model performance is continuously monitored on the validation set, and early stopping is applied to identify the optimal number of boosting iterations and to prevent overfitting. Upon completion of training, the finalized model is evaluated on the unseen test set using standard performance metrics, including accuracy, macro F1-score, and confusion matrix analysis, providing a comprehensive assessment of classification performance.

Validation Monitoring

LGBMClassifier

▼ Parameters

Parameter	Value
boosting_type	'gbdt'
num_leaves	128
max_depth	-1
learning_rate	0.03
n_estimators	1200
subsample_for_bin	200000
objective	'multiclass'
class_weight	{np.int64(0): 1.360074074074074, np.int64(1): 1.0317487075747358, np.int64(2): 1.4076203618521925, np.int64(3): 0.7129931655793724, ...}
min_split_gain	0.0
min_child_weight	0.001
min_child_samples	20
subsample	0.9
subsample_freq	0
colsample_bytree	0.9
reg_alpha	0.0
reg_lambda	0.0
random_state	42
n_jobs	-1
importance_type	'split'
num_class	12

Table: *LightGBM Model Training Parameters*

Model Evaluation

```
warnings.warn(
LightGBM Test Accuracy: 0.9941972920696325
LightGBM Macro F1     : 0.9920588952131922
```

Table: *Baseline Model Result*

4.2 Deep Learning Feature Learning

To explore representation learning from raw sensor signals, a convolutional neural network (CNN) is developed as a deep feature extractor. CNNs are well suited for time-series data because they can capture local temporal patterns and invariant features across sensor channels. In this study, the CNN is used primarily for feature learning rather than as a standalone classification model.

4.2.1 CNN Feature Extractor Architecture

Model: "CNN_Feature_Extractor"

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 256, 41)	0
conv1d_6 (Conv1D)	(None, 256, 64)	18,432
batch_normalization_6 (BatchNormalization)	(None, 256, 64)	256
re_lu_3 (ReLU)	(None, 256, 64)	0
max_pooling1d_4 (MaxPooling1D)	(None, 128, 64)	0
conv1d_7 (Conv1D)	(None, 128, 128)	41,088
batch_normalization_7 (BatchNormalization)	(None, 128, 128)	512
re_lu_4 (ReLU)	(None, 128, 128)	0
max_pooling1d_5 (MaxPooling1D)	(None, 64, 128)	0
conv1d_8 (Conv1D)	(None, 64, 256)	98,560
batch_normalization_8 (BatchNormalization)	(None, 64, 256)	1,024
re_lu_5 (ReLU)	(None, 64, 256)	0
global_average_pooling1d (GlobalAveragePooling1D)	(None, 256)	0
dropout_7 (Dropout)	(None, 256)	0

Total params: 159,872 (624.50 KB)

Trainable params: 158,976 (621.00 KB)

Non-trainable params: 896 (3.50 KB)

Figure: *Architecture of the CNN Feature Extractor*

4.3 Hybrid Model: CNN → LightGBM

4.3.1 Hybrid Model Motivation

To leverage the strengths of both deep learning and traditional machine learning, a hybrid modeling framework is proposed. In this framework, a convolutional neural network (CNN) is employed to automatically learn informative feature representations from raw sensor data segments, while LightGBM is used as the final classifier. This combination is designed to improve generalization to unseen subjects while maintaining high classification accuracy and computational efficiency.

4.3.2 CNN-Based Feature Extraction

After the CNN is trained, the final softmax classification layer is removed, transforming the network into a feature extractor. The activations from the penultimate fully connected layer are taken as compact feature representations for each sensor window. These learned embeddings encode temporal dynamics and inter-channel relationships and are then used as input features to train the LightGBM classifier.

```
144/144 ━━━━━━━━━━━━━━━━━━━━ 3s 19ms/step
59/59 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step
33/33 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step
CNN Feature Shapes:
(18361, 256) (7438, 256) (4136, 256)
```

Figure: *CNN-Based Feature Extraction and Hybrid Feature Fusion Process*

4.3.3 LightGBM Training on Learned Features

The feature representations extracted by the CNN are used to train a LightGBM classifier under the same subject-wise data partitioning strategy. This enables the classifier to leverage high-level temporal features learned by the CNN while benefiting from the strong decision-making ability and generalization performance of gradient boosting. The hybrid model is evaluated using accuracy, macro F1-score, and confusion matrix analysis.

4.3.4 Hybrid LightGBM Model Classifier

LGBMClassifier

▼Parameters

boosting_type	'gbdt'		
num_leaves	128		
max_depth	-1	subsample	0.8
learning_rate	0.03	subsample_freq	0
n_estimators	1200	colsample_bytree	0.8
subsample_for_bin	200000	reg_alpha	0.0
objective	'multiclass'	reg_lambda	0.0
class_weight	'balanced'	random_state	42
min_split_gain	0.0	n_jobs	-1
min_child_weight	0.001	importance_type	'split'
min_child_samples	20	num_class	12

Table: *Hybrid Model Training Parameters*

4.3.5 Hybrid Model Evaluation

```
Hybrid Test Accuracy : 0.9927466150870407
Hybrid Macro F1      : 0.9909846666489771
```

Table: *Hybrid Model Result*

5. Mathematical Representation of the Proposed Model

Let the dataset be represented as

where $x_i \in \mathbb{R}^d$ denotes a feature vector extracted from a time-series window and $y_i \in \{1,2,\ldots,C\}$ denotes the corresponding activity class.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \quad \text{... (i)}$$

5.1 LightGBM Classifier

LightGBM is a gradient boosting decision tree (GBDT) model that builds an ensemble of decision trees in a stage-wise manner. At iteration t, the model prediction is given by:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \quad \text{... (ii)}$$

where f_k represents the k-th decision tree.

The objective function optimized by LightGBM is:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k) \quad \text{.................................. (iii)}$$

where $\ell(\cdot)$ is the multi-class classification loss and $\Omega(\cdot)$ is a regularization term that controls model complexity.

5.2 CNN-Based Feature Extraction

For the hybrid model, a convolutional neural network (CNN) is used to extract discriminative temporal features from raw sensor windows. Given an input window $X \in \mathbb{R}^{T \times M}$, a one-dimensional convolution operation is defined as:

$$h_j = \sigma\left(\sum_{k=1}^{K} w_k \cdot X_{j+k-1} + b\right) \quad \text{.................................. (iv)}$$

where w_k and b are learnable parameters, K is the kernel size, and $\sigma(\cdot)$ denotes a nonlinear activation function.

The output of the final CNN feature layer is used as an input feature vector for the LightGBM classifier, forming the hybrid CNN–LightGBM model.

6. Experimental Results

The baseline LightGBM model trained on handcrafted statistical features exhibits strong performance on the subject-independent test set**. It attains an accuracy of 99.4% and a macro F1-score of 99.2%,** demonstrating its ability to effectively capture discriminative patterns from the extracted sensor features. These findings confirm that feature-based gradient boosting serves as a highly robust and reliable baseline for subject-independent human activity recognition.

6.1 Baseline LightGBM Results

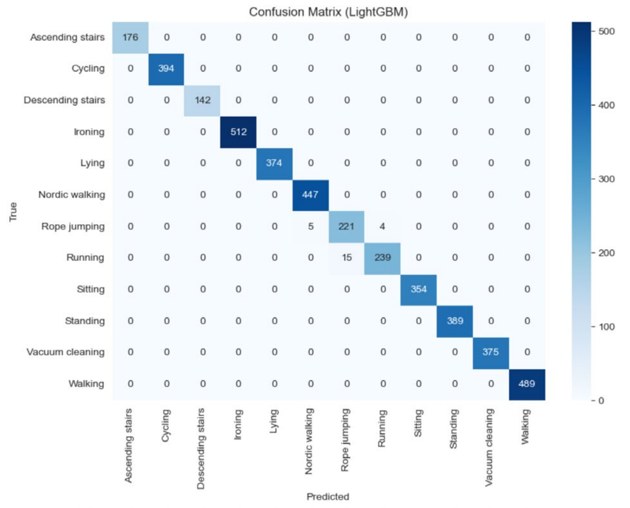


Figure: *Confusion Matrix (LightGBM)*

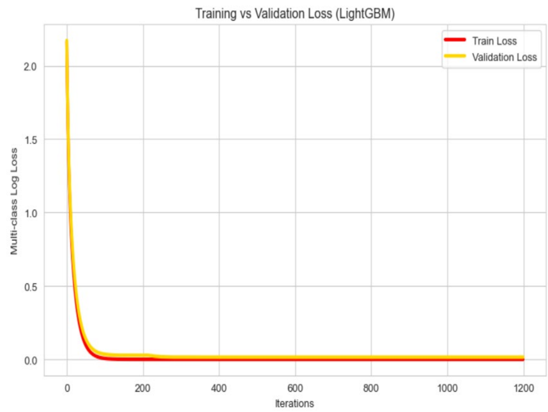


Figure: *Training vs Validation Loss*

Classification Report:

	precision	recall	f1-score	support
Ascending stairs	1.00	1.00	1.00	176
Cycling	1.00	1.00	1.00	394
Descending stairs	1.00	1.00	1.00	142
Ironing	1.00	1.00	1.00	512
Lying	0.98	0.99	0.98	374
Nordic walking	0.98	1.00	0.99	447
Rope jumping	1.00	0.89	0.94	230
Running	0.97	0.99	0.98	254
Sitting	1.00	1.00	1.00	354
Standing	1.00	1.00	1.00	389
Vacuum cleaning	1.00	1.00	1.00	375
Walking	1.00	1.00	1.00	489
accuracy			0.99	4136
macro avg	0.99	0.99	0.99	4136
weighted avg	0.99	0.99	0.99	4136

Table: *Classification Report*

6.2 Hybrid CNN → LightGBM Results

The hybrid CNN–LightGBM model further enhances classification performance by utilizing deep feature representations learned by the CNN**. It achieves an accuracy of 99.27% and a macro F1-score of 99.10%,** demonstrating the effectiveness of integrating deep temporal feature extraction with gradient boosting–based classification. These results underscore the advantages of the proposed hybrid framework for subject-independent human activity recognition.

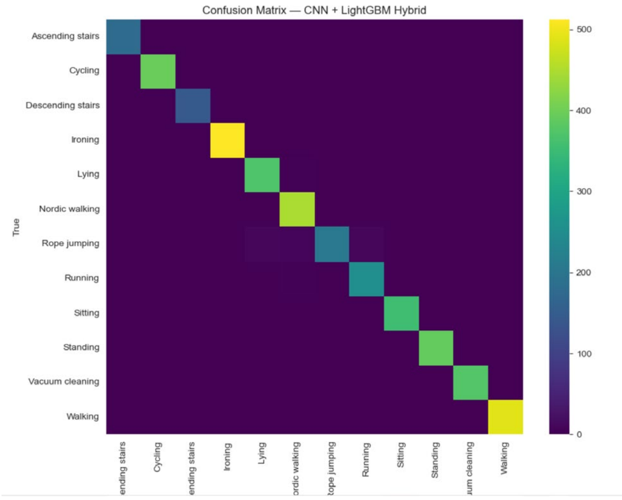


Figure: *Confusion Matrix (Hybrid Model)*

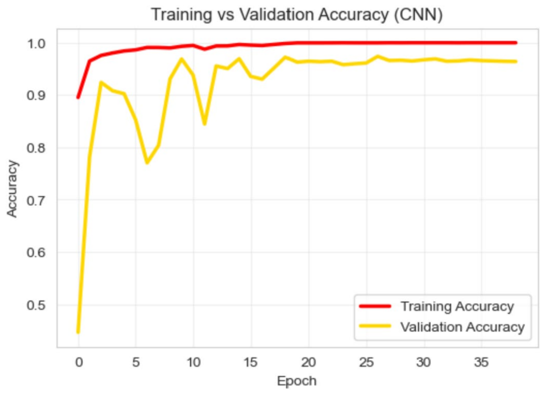


Figure: *Training vs Validation Accuracy*

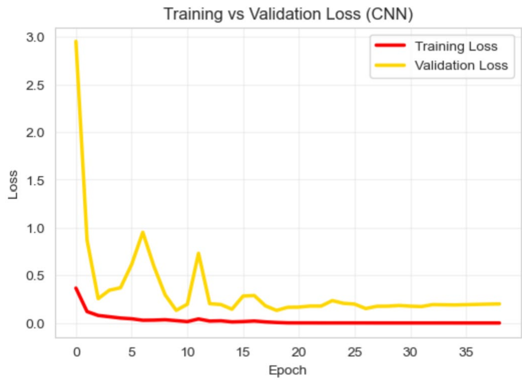


Figure: *Training vs Validation Loss*

Classification Report:

	precision	recall	f1-score	support
Ascending stairs	1.00	1.00	1.00	176
Cycling	1.00	1.00	1.00	394
Descending stairs	1.00	1.00	1.00	142
Ironing	1.00	1.00	1.00	512
Lying	0.98	0.99	0.98	374
Nordic walking	0.98	1.00	0.99	447
Rope jumping	1.00	0.89	0.94	230
Running	0.97	0.99	0.98	254
Sitting	1.00	1.00	1.00	354
Standing	1.00	1.00	1.00	389
Vacuum cleaning	1.00	1.00	1.00	375
Walking	1.00	1.00	1.00	489
accuracy			0.99	4136
macro avg	0.99	0.99	0.99	4136
weighted avg	0.99	0.99	0.99	4136

Table: *Hybrid Model Performance Metrics*

7. Comparative Analysis

7.1 Comparison of Baseline and Hybrid Models

A comparative analysis is performed to assess the performance differences between the baseline LightGBM model and the proposed hybrid CNN–LightGBM model. Both approaches are trained and evaluated using the same subject-wise experimental protocol to ensure a fair and unbiased comparison.

7.2 Accuracy and Macro F1 Comparison

The hybrid model attains higher classification accuracy and macro F1-score than the baseline model, indicating improved generalization across activity classes. The radar chart illustrates the relative improvements in both evaluation metrics and offers an intuitive visual comparison of the performance differences between the two approaches.

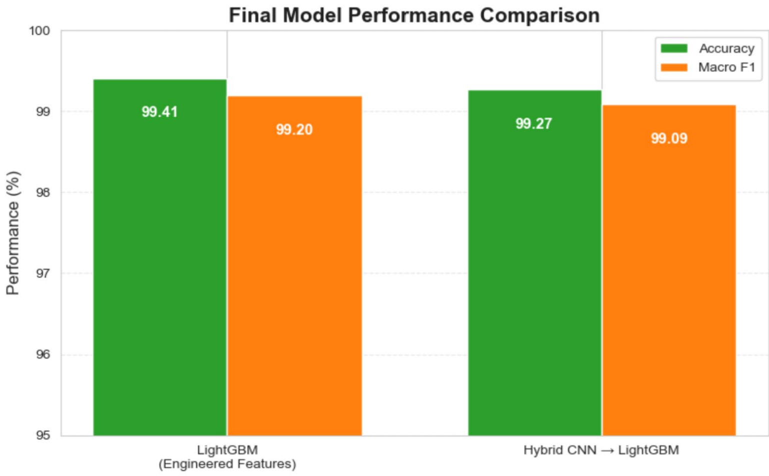


Figure: *Accuracy and Macro F1 Comparison (Baseline vs Hybrid)*

	Model	Test Accuracy (%)	Macro F1 (%)
0	LightGBM (Engineered Features)	99.41	99.20
1	Hybrid CNN → LightGBM	99.27	99.09

Table : *Final Model Comparison (Baseline vs Hybrid)*

7.3 Radar Chart Comparing

Radar chart illustrating the comparison of accuracy and macro F1-score between the baseline LightGBM model and the hybrid CNN–LightGBM model.

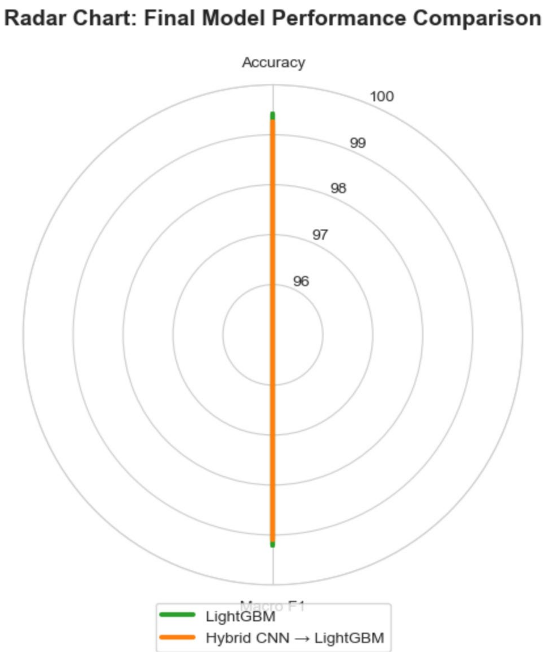


Figure: *Radar Chart Comparison(Accuracy & Macro F1)*

8. Discussion

The experimental findings emphasize the significant role of feature engineering in enhancing human activity recognition performance. The baseline LightGBM model, trained on well-designed statistical features, achieves an accuracy **of 99.4% and a macro F1-score of 99.2%,** demonstrating strong discriminative capability despite class imbalance and inter-subject variability. These results indicate that carefully engineered time-domain features are sufficient to capture key motion characteristics for reliable classification.

The hybrid CNN–LightGBM model further leverages temporal representations learned by the CNN, attaining an accuracy of **99.27% and a macro F1-score of 99.10%**. Although the CNN alone does not generalize effectively as a standalone classifier, its learned feature embeddings provide complementary temporal information that enhances LightGBM's decision-making when used as input features.

Subject-wise evaluation confirms that both models generalize well to unseen individuals, highlighting their robustness to subject-specific variations. Overall, the results suggest that integrating engineered statistical features with learned representations offers a practical and highly accurate approach for subject-independent human activity recognition using wearable sensor data.

9. Conclusion

This project presents a subject-independent human activity recognition system based on physiological and motion sensor time-series data. The proposed methodology integrates comprehensive data preprocessing, subject-wise data partitioning, time-series window segmentation, feature engineering, and model training.

Experimental results indicate that the LightGBM model trained on engineered features delivers strong performance, while the hybrid CNN–LightGBM framework further demonstrates the advantages of combining deep feature learning with traditional machine learning techniques. Both approaches achieve high classification accuracy and macro F1-scores on previously unseen subjects.

In summary, the project shows that rigorous preprocessing, effective feature extraction, and hybrid modeling strategies can provide reliable and high-performance solutions for real-world human activity recognition applications.

References

- *Lara, O. D., & Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. IEEE Communications Surveys & Tutorials, 15(3), 1192–1209.*
- *Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. ESANN.*
- *Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*
- *Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems (NeurIPS).*