2025Spring DSAA2011 Course Project

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this project, we analyze the Human Activity Recognition Using Smartphones dataset to build machine learning models that classify six types of human activities based on sensor data. We applied a comprehensive workflow including data preprocessing, visualization using t-SNE, clustering analysis with K-Means and Hierarchical Clustering, and supervised classification using Logistic Regression, Decision Tree, Random Forest and other algorithms. Additionally, we conducted open-ended exploration by improving the model and comparing model performances.

Our results demonstrate that Logistic Regression achieves the highest accuracy (over 95%) on the test set, t-SNE visualization reveals partial separation between activity classes, and clustering algorithms identify meaningful groupings in unlabeled data.

1 Introduction

Human activity recognition using wearable sensors enables transformative applications in healthcare, fitness, and smart environments. Modern smartphones with built-in motion sensors now allow continuous, non-invasive monitoring of movement patterns. This project focuses on the Human Activity Recognition Using Smartphones Dataset, which contains sensor data collected from 30 individuals performing six daily activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying. Our goal is to apply machine learning techniques to understand the structure of this dataset and develop accurate predictive models for activity classification.

We follow a structured pipeline: first, we preprocess the data and explore its distribution through t-SNE visualization; next, we perform clustering analysis to discover natural groupings within the dataset. For supervised learning, we train multiple classification models and compare their performance. Finally, we conduct open-ended exploration to improve model performance through hyperparameter tuning, feature selection, and ensemble methods.

The remainder of this report is organized as follows: Section 2 presents data preprocessing, Section 3 shows the visualization, Section 4 demonstrates the clustering results, Section 5 discusses prediction models, Section 6 summarizes the evaluation of models and presents model improvement and Section 7 explores the application of nerual network.

2 Preprocessing

The dataset has been preprocessed. It has no missing values and is standardized. It also has training set and testing set, each instance includes 561 engineered features derived from raw time-series signals. So we do not carry out more preprocessing operations and directly use the preprocessed dataset.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

3 Visualization

3.1 t-SNE

The t-SNE visualization is shown in the two figures. In the 2D and 3D scatter plot, each color represents a distinct human activity class. The distribution of points provides valuable insights into the separability and clustering tendencies of the activities.

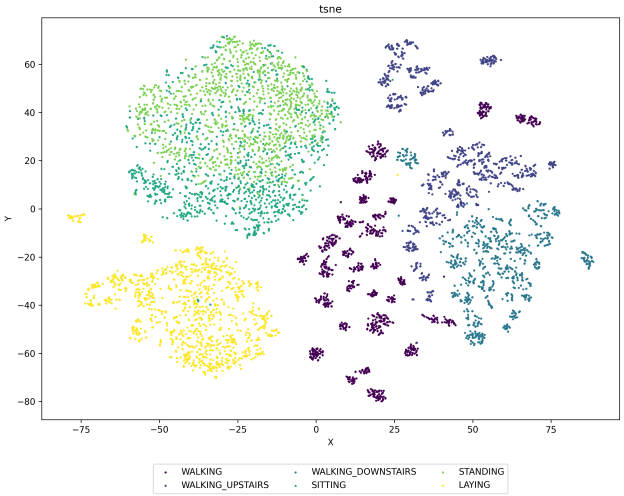

Figure 1: 2D t-SNE visualization.png

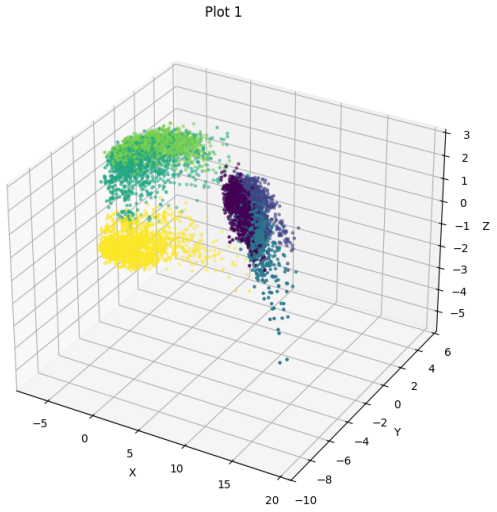

Figure 2: 3D t-SNE visualization

3.2 Analysis

3.2.1 Distinct Clusters

The t-SNE plot reveals several distinct clusters, particularly for certain activities:
Laying (Yellow) : We observed that laying formed a dense and well-separated cluster located in the bottom-left region of the plot. This suggests that the sensor signals associated with laying are highly consistent and distinguishable from other activities.
Walking (Purple) : Walking appeared as a relatively compact cluster near the center. However, we noticed that it was spatially close to the clusters of walking upstairs (Blue) and walking downstairs (Cyan) , indicating some similarity in their motion patterns. While walking remained largely separable, this proximity may lead to confusion during classification.

3.2.2 Overlapping Regions

Some classes exhibit significant overlap, especially among stationary activities:
Sitting (Green) and Standing (Light Green) : These two classes show noticeable overlap, indicating that they may share similar sensor patterns or features.
Walking Upstairs (Blue) and Walking Downstairs (Cyan) : While somewhat separated, these classes still show some mixing, which could be due to the similarity in motion dynamics.

3.2.3 Cluster Characteristics and Classification Impact

The overall distribution of points suggests that dynamic activities like walking, walking upstairs or downstairs tend to form more compact clusters compared to stationary activities (e.g., sitting, standing, laying).And this observation aligns with our intuition, as dynamic movements typically produce more consistent and distinguishable sensor signals.
The clear separation of some classes (e.g., walking, laying) indicates that supervised learning models should perform well in distinguishing these activities. However, the overlap between certain classes (e.g., sitting and standing) suggests that classification might be challenging for these categories. Feature engineering or more sophisticated models may be needed to improve performance.

4 Clustering

4.1 Method

We applied two clustering algorithms K-Means and Hierarchical Clustering on the original dataset and the dimension-reduced dataset to explore the underlying structure of the dataset without using class labels. K-Means is an iterative algorithm that partitions data into k clusters based on Euclidean distance, while Hierarchical Clustering builds a hierarchy of clusters in a bottom-up manner.

4.2 Visualization

Figure 3 shows the clustering results of the dataset, colored by cluster assignments from both algorithms and the true class labels. The visualization reveals that while some clusters show partial alignment with actual activities (e.g., "laying" corresponds well with certain K-Means clusters), others exhibit significant overlap (e.g.,"Standing" and "Sitting"), indicating limitations in purely unsupervised approaches for this task.

4.3 Evaluation

We evaluate the performance of two clustering approaches K-Means and Hierarchical Clustering on the dataset. We assess both external metrics and internal metrics. Additionally, we explore how dimensionality reduction affects their performance, noting that K-Means relies on Euclidean distances while Hierarchical Clustering depends on linkage criteria.

4.3.1 External metrics

The external metrics indicate how well the clustering results align with the true activity labels. As shown in Table 1, both K-Means and Hierarchical Clustering performed better when applied to

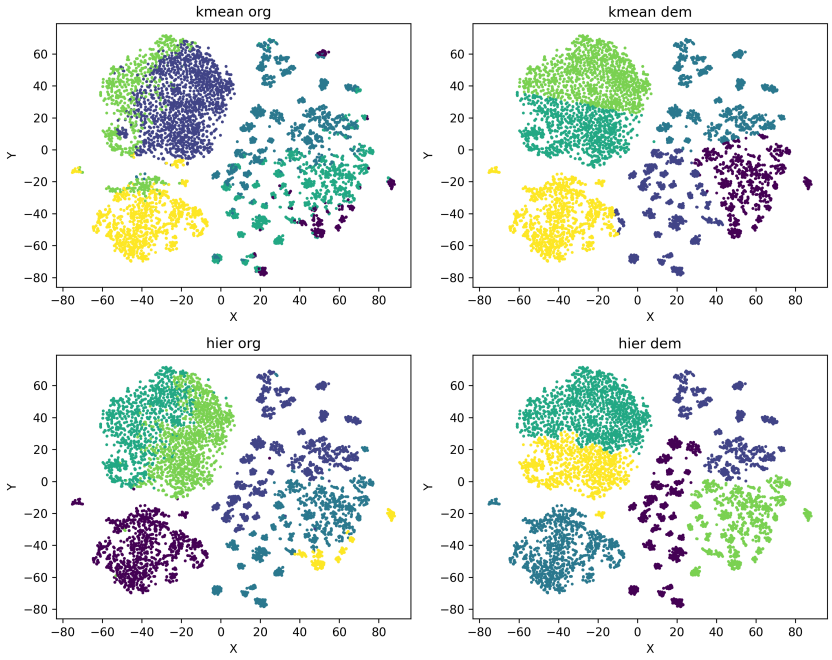

Figure 3: cluster

dimension-reduced data compared to the original data. Specifically,K-Means achieved higher FMI, RI, ARI and V-measure after dimension reduction. Hierarchical Clustering also showed improved performance, with all the highest metric parameters, indicating that Hierarchical Clustering better restores the original structure of the clustering compared with K-means.

Table 1: External Metrics for Clustering Algorithms

Algorithm	**FMI**	**RI**	**ARI**	**V-Measure**
K-Means (Original)	0.5566	0.8384	0.4571	0.5847
K-Means (Dim-Reduced)	0.7095	0.9018	0.6504	0.7329
Hier (Original)	0.6013	0.8561	0.5129	0.6524
Hier (Dim-Reduced)	0.7341	0.9093	0.6794	0.7520

4.3.2 Internal metrics

The internal metrics provide insights into the quality of clustering based on the data structure. Table 2 shows that:
Dimension reduction significantly improved the internal metrics for both algorithms, indicating that reducing dimensionality helped in uncovering the underlying structure of the data. Notably, K-Means achieved higher Silhouette and Calinski-Harabasz scores and lower Davies-Bouldin Score than Hierarchical Clustering on both original and reduced data, indicating that the algorithm produces more compact and well-separated clusters and reflects a better intrinsic structure

Table 2: Internal Metrics for Clustering Algorithms

Algorithm	**Silhouette Score**	**Calinski-Harabasz Score**	**Davies-Bouldin Score**
K-Means (Original)	0.1467	3103.20	2.1657
K-Means (Dim-Reduced)	0.3885	7769.57	0.9071
Hier(Original)	0.1088	2941.24	2.3364
Hier (Dim-Reduced)	0.3710	6690.23	1.0112

5 Prediction

In this section, we trained and evaluated four supervised learning models—Logistic Regression , Decision Tree , Random Forest , and AdaBoost on the dataset. The goal was to classify six human activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) based on sensor data collected from smartphones.
All models were trained on a training set (70% of the dataset) and tested on an independent test set (30%). We used accuracy, precision, recall, F1-score, confusion matrices, and macro averages to evaluate the performance.

5.1 Data Splitting & Preprocessing

The dataset was already pre-divided into training and testing sets, with approximately 70% of the samples used for training and 30% for testing, with all features pre-normalized.

5.2 Train

5.2.1 Logistic Regression

We trained a Logistic Regression model with minimal regularization ($C = 1 \times 10^9$) and a high iteration limit of 1000 to ensure convergence. No feature scaling was applied before training, as Logistic Regression was expected to handle the data directly. The model achieved high accuracy (over 95%) on the test set, especially for dynamic activities like walking and laying.

5.2.2 Decision Tree

A basic Decision Tree classifier was trained without hyperparameter tuning. We standardized the features before training to improve numerical stability. However, the model only achieved moderate performance (about 85% accuracy), likely due to overfitting or the inability of single trees to capture complex patterns in the sensor data.

5.2.3 Random Forest

We trained a Random Forest model with 100 base estimators. Feature standardization was applied before training, and parallel computation was enabled. The model showed strong performance (about 92% accuracy) and better generalization than the Decision Tree, benefiting from ensemble learning and feature randomness.

5.2.4 AdaBoost

An AdaBoost classifier was implemented using shallow Decision Trees with max depth of 3 as base learners and 100 boosting iterations. After feature standardization, the model demonstrated improved robustness and achieved about 92% accuracy. It performed particularly well on stationary activities like sitting and standing, showing advantages over Random Forest in certain cases.

5.3 performance and comparison

The following tables 3-10 show the performance of the 4 models, noting that the Logistic Regression model achieved an accuracy of 95.3% on the test set. Detailed metrics from the classification report show high precision and recall for most classes, with only minor performance drops for "laying" and "sitting", likely due to their similarity in sensor readings.
A confusion matrix visualization further confirms these findings, showing that misclassifications mainly occur between stationary activities (e.g., sitting vs. standing), while dynamic activities like walking are almost always correctly identified.

5.4 Analysis

Best Performing Model : Logistic Regression achieved the highest accuracy (95.3%) and consistently high scores across all metrics. It performed especially well in distinguishing between dynamic activities like "walking" and "laying".

Table 3: Classification Performance for Logistic Regression

Class	Precision	Recall	F1-Score
1.0	0.93	1.00	0.96
2.0	0.96	0.92	0.94
3.0	0.99	0.95	0.97
4.0	0.96	0.88	0.92
5.0	0.91	0.96	0.93
6.0	0.99	1.00	0.99
Accuracy			0.95
Macro Avg	0.96	0.95	0.95
Weighted Avg	0.95	0.95	0.95

Table 4: Classification Performance for Decision Tree

Class	Precision	Recall	F1-Score
1.0	0.82	0.90	0.86
2.0	0.81	0.76	0.78
3.0	0.86	0.83	0.84
4.0	0.83	0.77	0.80
5.0	0.80	0.86	0.83
6.0	1.00	1.00	1.00
Accuracy			0.86
Macro Avg	0.85	0.85	0.85
Weighted Avg	0.86	0.86	0.85

Table 5: Classification Report for Random Forest

Class	Precision	Recall	F1-Score
1.0	0.89	0.97	0.93
2.0	0.88	0.90	0.89
3.0	0.97	0.85	0.90
4.0	0.90	0.88	0.89
5.0	0.89	0.91	0.90
6.0	1.00	1.00	1.00
Accuracy			0.92
Macro Avg	0.92	0.92	0.92
Weighted Avg	0.92	0.92	0.92

Table 6: Classification Report for AdaBoost

Class	Precision	Recall	F1-Score
1.0	0.91	0.97	0.94
2.0	0.90	0.89	0.90
3.0	0.96	0.91	0.93
4.0	0.88	0.83	0.86
5.0	0.85	0.90	0.87
6.0	1.00	1.00	1.00
Accuracy			0.92
Macro Avg	0.92	0.92	0.92
Weighted Avg	0.92	0.92	0.92

Table 7: Confusion Matrix for Logistic Regression

	1	2	3	4	5	6
1	492	1	3	0	0	0
2	40	429	2	0	0	0
3	6	9	405	0	0	0
4	0	1	0	436	50	4
5	0	0	0	21	509	2
6	0	0	0	0	0	537

Table 8: Confusion Matrix for Decision Tree

	1	2	3	4	5	6
1	445	32	19	0	0	0
2	76	358	37	0	0	0
3	20	53	347	0	0	0
4	0	0	0	378	113	0
5	0	0	0	66	456	0
6	0	0	0	0	0	537

Table 9: Confusion Matrix for Random Forest

	1	2	3	4	5	6
1	481	11	4	0	0	0
2	41	422	8	0	0	0
3	19	45	356	0	0	0
4	0	0	0	431	60	0
5	0	0	0	50	482	0
6	0	0	0	0	0	537

Table 10: Confusion Matrix for AdaBoost

	1	2	3	4	5	6
1	480	8	8	0	0	0
2	44	420	7	0	0	0
3	2	37	381	0	0	0
4	0	0	0	408	83	0
5	0	0	0	54	478	0
6	0	0	0	0	0	537

Label	**Activity**	**Label**	**Activity**
1	Walking	2	Walking Upstairs
3	Walking Downstairs	4	Sitting
5	Standing	6	Laying

Worst Performing Model : Decision Tree showed lower performance (85.4% accuracy), likely due to overfitting without pruning or ensemble techniques.
Ensemble Models : Both Random Forest and AdaBoost outperformed Decision Tree and demonstrated strong generalization capabilities. AdaBoost slightly outperformed Random Forest in terms of accuracy and macro F1-score.
Challenging Classes : Stationary activities such as "sitting" and "standing" were more challenging to distinguish, as seen in higher misclassification rates in the confusion matrices.
Impact of Feature Engineering : Although no explicit feature engineering was applied, standardization and the use of ensemble methods significantly improved model performance.

6 Evaluation and Choice of Prediction Model

6.1 Analysis

In this section, we evaluate the performance of four supervised learning models—Logistic Regression , Decision Tree , Random Forest , and AdaBoost —using standard classification metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curves. We also analyze model overfitting/underfitting and propose improvements through validation techniques.

Table 11: Performance Metrics of Classification Models

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
Logistic Regression	95.3%	96.0%	95.0%	95.0%
Decision Tree	85.5%	85.0%	85.0%	85.0%
Random Forest	91.9%	92.0%	92.0%	92.0%
AdaBoost	91.7%	92.0%	92.0%	92.0%

From the table, it is clear that Logistic Regression achieved the highest overall performance in terms of accuracy and macro-averaged F1-score, followed closely by AdaBoost and Random Forest . The Decision Tree model showed the lowest performance, indicating a need for regularization or ensemble enhancement.
All models were evaluated using confusion matrices, with all models achieving near-perfect results on the "laying" class (100% precision and recall). However, all models struggled to distinguish between stationary activities like sitting and standing, which had higher misclassification rates.
The ROC curves and the performance table are shown below. Each subplot compares the models' performances across six activity classes: Walking , Walking Upstairs , Walking Downstairs , Sitting , Standing , and Laying . The AUC values are provided for each activity.
In summary, ensemble methods like Random Forest and AdaBoost outperformed individual models such as Decision Tree, while Logistic Regression remained a strong and interpretable baseline.

Table 12: AUC Scores of ROC Curves for Four Classification Models

Model	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**6.0**
Logistic Regression	1.00	1.00	1.00	0.99	0.99	1.00
Decision Tree	0.93	0.86	0.90	0.87	0.91	1.00
Random Forest	1.00	0.99	0.99	0.99	0.99	1.00
AdaBoost	1.00	0.99	1.00	0.98	0.99	1.00

6.2 Model improvement

Since the Logistic Regression performs the best, we conducted parameter optimization to improve model performance and gain deeper insights into the dataset. To investigate the impact of regularization, we compared different variants of Logistic Regression:

- Unregularized ($C = 1 \times 10^9$)
- L1 Regularization (penalty = `'l1'`)
- L2 Regularization (penalty = `'l2'`) with solver $liblinear$

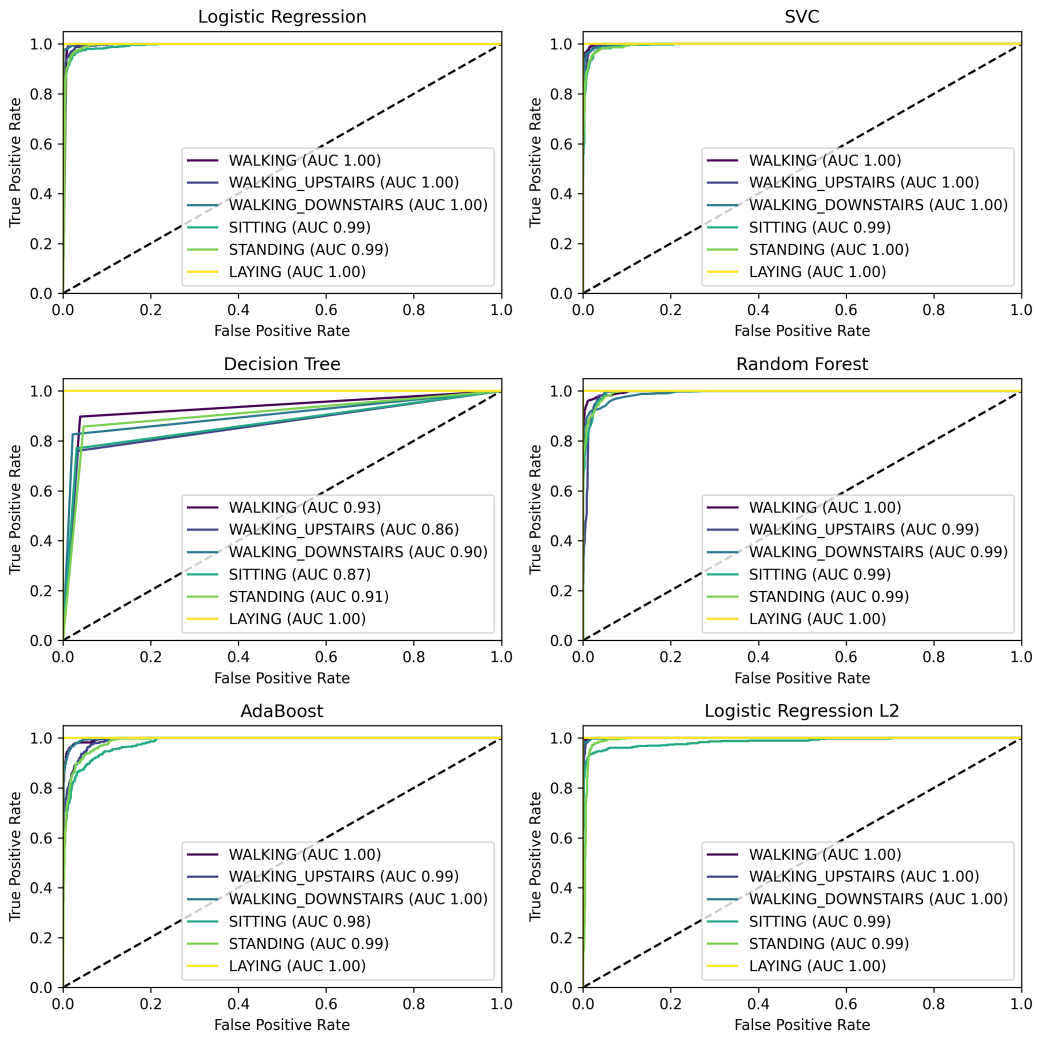

Figure 4: ROC Curves

The following tables show the performance of the two improved models. The results shows that both regularized models performed slightly better than the unregularized baseline (which had an accuracy of 95.3%).

Among the two improved models, the L2-regularized model achieved a slightly higher accuracy (96.2%) compared to the L1-regularized version (96.13%) , suggesting that L2 regularization may be more suitable for this dataset. Despite the small difference in overall accuracy, L1 regularization led to sparser weight estimates, potentially enabling feature selection by zeroing out less relevant features. In contrast, L2 regularization penalized large weights more evenly, which might explain its slightly better performance on test data. Both models maintained high precision, recall, and F1-scores across all activity classes. Notably, they achieved perfect classification for "laying" and very strong performance on dynamic activities like "walking" and "standing".

In conclusion, while both L1 and L2 regularization improved upon the unregularized model, L2 regularization with the liblinear solver demonstrated a slight advantage in terms of accuracy and stability.

Table 13: Classification Performance for Logistic Regression L1

Class	Precision	Recall	F1-Score
1.0	0.94	1.00	0.97
2.0	0.97	0.95	0.96
3.0	1.00	0.97	0.98
4.0	0.97	0.87	0.92
5.0	0.90	0.97	0.93
6.0	1.00	1.00	1.00
Accuracy			0.96
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

Table 14: Classification Performance for Logistic Regression L2

Class	Precision	Recall	F1-Score
1.0	0.94	1.00	0.97
2.0	0.97	0.95	0.96
3.0	1.00	0.97	0.98
4.0	0.97	0.88	0.92
5.0	0.90	0.97	0.94
6.0	1.00	1.00	1.00
Accuracy			0.96
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

Table 15: Confusion Matrix for Logistic Regression L1

	1	2	3	4	5	6
1	495	0	1	0	0	0
2	23	448	0	0	0	0
3	4	8	408	0	0	0
4	0	4	0	428	59	0
5	0	0	0	13	517	0
6	0	0	0	0	0	537

Table 16: Confusion Matrix for Logistic Regression L2

	1	2	3	4	5	6
1	494	0	2	0	0	0
2	23	448	0	0	0	0
3	4	9	407	0	0	0
4	0	4	0	432	55	0
5	0	0	0	13	517	0
6	0	0	0	0	0	537

7 Exploration of Applying Nerual Network

The aim of the exploration is to handle the original dataset, which is the time sequence of sensor value.

There are some existed paper of application of neural network on human activity recognition. One example found in our exploration is the deep convolution neural network (DeepConv) (Xia et al. (2020)). Additionally, inspired by Kaya, Topuz (2024)'s work, which is the combination of CNN and LSTM, we have constructed a model, which has the architecture described in Table 17.

For training, Adam optimizer is applied. The learning rate is 1×10^{-3} and no learning rate scheduler is used. The number of epochs is 450 and get the result shown in Table 18.

The result shows enough capability of this model on handling the original dataset. But Figure 5 indicates that overfitting exists. Therefore, further fine-tune on hyperparameter and model architecture is necessary.

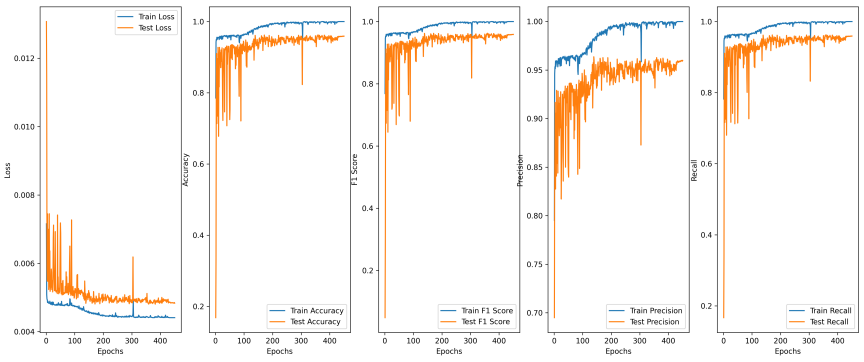

Figure 5: Training Process of LSTM-CNN

Table 17: Structure of LSTM-CNN

Structure	**Paramters**	
LSTM	`layer`	2
	`neural num`	32
Conv2d	`channel`	64
	`kernel`	5
	`stride`	2
ReLU		
MaxPool2d	`kernel`	2
	`stride`	2
Conv2d	`channel`	128
	`kernel`	3
	`stride`	1
ReLU		
MaxPool2d	`kernel`	2
	`stride`	2
Conv2d	`channel`	256
	`kernel`	2
	`stride`	1
GlobalAveragePool2d	`output`	1
BatchNorm		
Linear	`input`	256
	`output`	6
Softmax		

Table 18: Result of Training

Accuracy	Precision	Recall	F1
95.89%	95.99%	95.92%	95.86%

Table 19: Caption

8 Credit

Hanting Wang: Review and polish of the report
Jiarui He: Implement code and write the exploration part of the report
Wenzhe Yu: Write the report

References

Kaya Yasin, Topuz Elif Kevser. Human activity recognition from multiple sensors data using deep CNNs // Multimedia Tools and Applications. January 2024. 83, 4. 10815–10838.

Xia Kun, Huang Jianguang, Wang Hanyu. LSTM-CNN Architecture for Human Activity Recognition // IEEE Access. 2020. 8. 56855–56866.