Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.37909

Research on Human Activity Recognition Based on Smartphone Sensors

Ziyan Tao

Shoreline Community College, Shanghai, China
rumi080712@outlook.com

Abstract. To address the problem that current smart bands and fitness applications require users to manually select the exercise mode, which often leads to incorrect activity records, this paper studies an automatic human activity recognition method based on the built-in sensors of smartphones. The method is evaluated on the public UCI HAR dataset, which contains sensor records of six daily activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) collected from 30 volunteers, with 561 time-domain and frequency-domain features extracted from the signals of a three-axis accelerometer and a three-axis gyroscope for each record. Three classification models are compared: the decision tree, the K-Nearest Neighbor based on Euclidean distance, and KNN after PCA dimensionality reduction. The experimental results show that KNN achieves the best performance with an accuracy of 90.16%; the decision tree reaches 86.29%; after the features are compressed from 561 to 30 dimensions by PCA, KNN still maintains an accuracy of 87.41%, which significantly reduces the computational cost. The confusion matrix analysis shows that sitting and standing are the main sources of confusion (136 misclassifications in total), because the sensor signals of these two low-intensity static activities have extremely low amplitude and highly similar statistical features. In addition, a signal-level simulation platform is built to verify the learning curves, the hyperparameter sensitivity, and the noise robustness. The results prove the feasibility of high-accuracy activity recognition locally on mobile devices and provide a reference for the automatic activity recording of smart wearable devices.

Keywords: Human activity recognition, decision tree, dimensionality reduction

1. Introduction

With the widespread use of wearable devices such as smart wristbands, daily exercise monitoring has become an important means for public health management. However, the existing commercial fitness wristbands and their accompanying applications generally require users to manually select the exercise mode before starting the activity. If users forget to switch, the data recorded by the device will not match the actual exercise status, resulting in distorted key indicators such as energy consumption statistics and step count.

This problem is more common in consumer products: most mainstream wristbands and fitness applications on the market rely on manual mode selection. The fundamental reason is that some

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.37909

daily activities are highly similar at the sensor signal level. In both sitting and standing states, the signal amplitude measured by the built-in accelerometer of the device is close to zero. Traditional methods such as threshold determination are difficult to distinguish between the two, so manufacturers generally leave the mode selection to the users.

Therefore, the research question of this paper is: can the built-in accelerometer and gyroscope of smartphones be used to automatically recognize the six most common daily activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) through machine learning algorithms, so that users no longer need to switch modes manually and exercise records can be automated and accurate. Considerable research has been conducted on this question. Reyes-Ortiz et al. released the public UCI HAR dataset in 2013, which provides a unified benchmark for activity recognition based on smartphone sensors [1]. Kwapisz et al. proved in 2011 that recognizing daily activities from mobile phone accelerometer data is feasible [2]. Shoaib et al. systematically compared decision trees, KNN, Bayesian networks, and support vector machines on sensor-fusion-based activity recognition in 2014, and pointed out that different algorithms perform differently on dynamic and static activities [3]. In recent years, deep learning methods have achieved higher accuracy on HAR tasks: Ignatov realized real-time activity recognition from accelerometer data with convolutional neural networks, and Ramanujam and Zhang et al. reviewed the advances of deep learning methods from the perspectives of smartphone sensors and wearable sensors respectively [4-6].

However, deep models usually require considerable computational and storage resources, which conflicts with the lightweight requirement of on-device inference, and lightweight models such as TinyHAR proposed by Zhou et al. provide a feasible direction for this problem [7]. Nevertheless, most existing research focuses on improving the overall accuracy, while the analysis of which activity pairs are most easily confused and why remains insufficient. To address these shortcomings, this paper takes the UCI HAR dataset as the research object, builds and compares three activity recognition models — the decision tree, KNN, and PCA combined with KNN — to explore a lightweight recognition solution deployable on mobile devices. In addition to comparing the overall performance, this paper quantitatively analyzes the distribution and causes of recognition errors based on the confusion matrix, in order to provide clear directions for further improvement.

2. Dataset and sensor characteristics

This paper uses the public UCI HAR dataset which was jointly collected by the research teams of the University of Oviedo (Spain) and the University of Passau (Germany) [1]. The dataset records the movement data of 30 volunteers (aged 19 to 48) wearing a Samsung Galaxy S2 smartphone fixed at the waist.

The dataset contains six types of daily activities, as defined in Table 1. The researchers recorded the signals of the three-axis accelerometer and the three-axis gyroscope simultaneously at a sampling frequency of 50 Hz [8]. The continuous time-series signals were segmented into windows of 2.56 seconds (128 samples per window), and time-domain statistics and frequency-domain indicators were calculated within each window, forming a 561-dimensional feature vector. The time-domain features (mean, standard deviation, maximum, minimum, etc.) and the frequency-domain features (energy and entropy extracted after the fast Fourier transform) each account for approximately half of the total. The dataset has been pre-divided into a training set of 7,352 records and a test set of 2,947 records; using the official division avoids data leakage and ensures the comparability of different algorithms.

DOI: 10.54254/2755-2721/2026.37909

Table 1. Definition of dataset activity categories

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

3. Methods and system architecture

3.1. Overall concept

The workflow of the entire system can be divided into four steps: Firstly, the built-in accelerometer and gyroscope in the mobile phone collect motion data at a fixed frequency in real time; Secondly, 561-dimensional statistical features are extracted from the raw signals; Thirdly, the trained classification model is used to complete activity discrimination locally; Finally, the recognition results are automatically recorded in the application. The system architecture is shown in Figure 1. It should be emphasized that all calculations are completed locally on the mobile phone, without uploading to the cloud. This not only ensures response speed but also protects user privacy, embodying the design concept of edge computing.

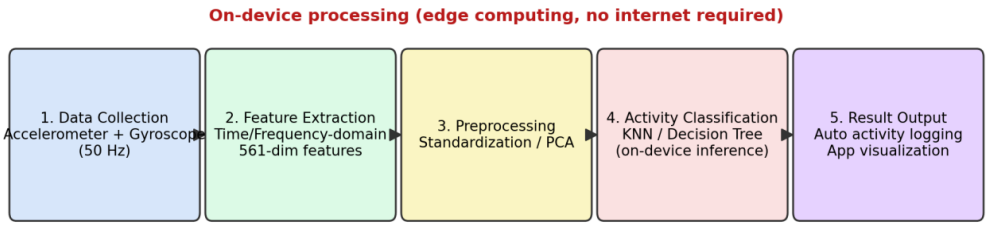

Figure 1. Overall system architecture. (picture credit : original)

3.2. K-Nearest Neighbor algorithm (KNN)

KNN, proposed by Cover and Hart in 1967, is the best-performing algorithm in the experiments of this paper [9]. Its basic idea is as follows: for a sample to be classified, search the training set for the k samples most similar to it, and use the majority category among these k samples as the prediction result. In this paper, $k = 5$ is chosen, that is, the prediction is made by majority voting over the 5 nearest neighbors.

The similarity between samples is measured by Euclidean distance. Let the sample to be classified $x = (x1, x2, ..., xd)$ and the training sample $xi = (xi1, xi2, ..., xid)$, with the feature dimension d = 561, the distance between the two is defined as:

$$d(x, xi) = \sqrt{(\Sigma(j=1 \rightarrow d) (xj - xij)^2)} \quad (1)$$

DOI: 10.54254/2755-2721/2026.37909

$$\hat{y} = \text{argmax}(c)\ \Sigma(xi \in Nk(x))\ I\ (yi = c) \tag{2}$$

Where $Nk(x)$ denotes the set of k nearest training samples to the sample x, yi is the true activity label of the neighbor sample xi, and c is a candidate category (in this paper, c ranges from 1 to 6, corresponding to the six activities in Table 1). $I(\cdot)$ is an indicator function that equals 1 when the condition inside holds and 0 otherwise; in other words, Equation (2) counts the occurrences of each category in the neighbor set and outputs the most frequent one. KNN does not require an explicit training process and directly uses the training set as the model; its limitations are the high computational cost of distance calculation in high-dimensional spaces and its sensitivity to noisy features.

3.3. Decision tree

The decision tree adopts a different modeling approach from KNN: instead of saving all the training samples, it recursively learns a series of partitioning rules from the data, such as dividing the samples level by level based on a certain feature threshold until the leaf nodes output the category. Therefore, it has strong interpretability.

The node partitioning criterion uses Gini impurity, which measures the degree of category heterogeneity in a dataset: the more evenly the category distribution is, the higher the impurity is; the purer the samples are, the lower the impurity is. For a dataset S with m categories, it is defined as:

$$\text{Gini}(S) = 1 - \Sigma(i=1 \to m)\ (pi)^2 \tag{3}$$

Where pi represents the proportion of category i in the dataset S, and the smaller the Gini value, the purer the dataset. The algorithm selects the feature and threshold that leads to the greatest reduction in weighted Gini impurity for each partition. To prevent overfitting, this paper limits the maximum depth of the decision tree to 15 layers.

3.4. PCA dimensionality reduction + KNN

Although the 561-dimensional features contain rich information, there are redundancies and noises, and the calculation cost of high-dimensional distances is large [10]. To verify the impact of dimensionality reduction on recognition performance, principal component analysis (PCA) is introduced for feature compression: find the orthogonal projection direction with the largest variance among 561 feature directions, project the data onto a small number of new directions, and retain the main information while suppressing noise.

This paper compresses the 561-dimensional features to 30 dimensions. This value has experimental basis: the first 30 principal components can already explain more than 90% of the variance of the original data, that is, while discarding about 95% of the dimensions, the information loss is very small, and it can significantly reduce the computational cost in the prediction stage, which is of great significance for the deployment on mobile devices with limited computing power.

Let the standardized data matrix be $X \in R(n \times d)$, its covariance matrix $\Sigma = X^T X/(n-1)$, after feature decomposition $\Sigma = W \Lambda W^T$, take the feature vectors corresponding to the r largest eigenvalues as the projection matrix Wr, and the dimensionality reduction is expressed as:

$$Z = X \cdot Wr \tag{4}$$

DOI: 10.54254/2755-2721/2026.37909

4. Experimental results and analysis

4.1. Model convergence verification

Before conducting the performance analysis of the prediction, it is necessary to verify that the model obtained from the training is convergent and usable. Figure 2 presents the learning curves of KNN (k=5) and decision tree, which show the trend of training accuracy and 5-fold cross-validation accuracy as the size of the training set changes from small to large. It can be seen that as the number of training samples increases, the cross-validation accuracy rises rapidly and stabilizes, and the gap between the training accuracy and the cross-validation accuracy is very small, indicating that the model does not have overfitting or underfitting and has stable generalization ability, which can be used for subsequent prediction analysis; the curve enters a plateau after the sample size reaches approximately 60% of the total training set, indicating that the current data size is sufficient for model training.

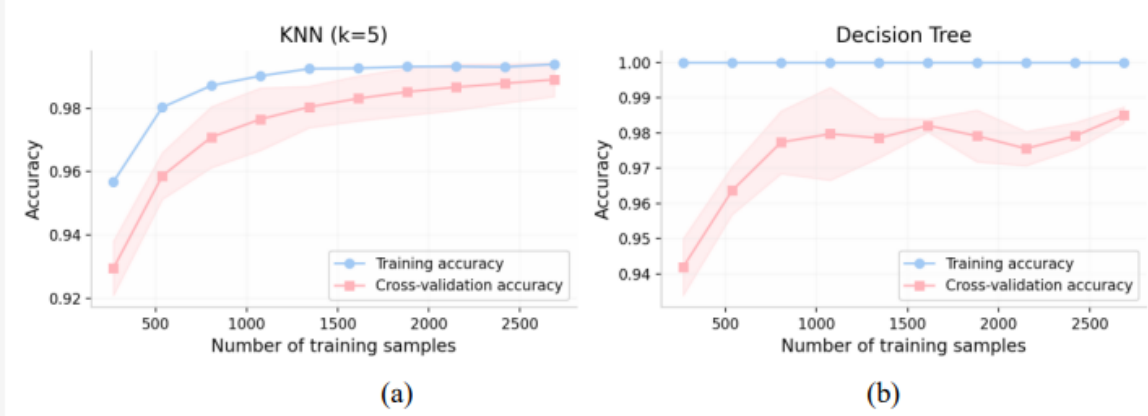

Figure 2. Model learning curve (5-fold cross-validation,(a) KNN(k=5); (b) decision tree (picture credit: original)

4.2. Hyperparameter sensitivity analysis

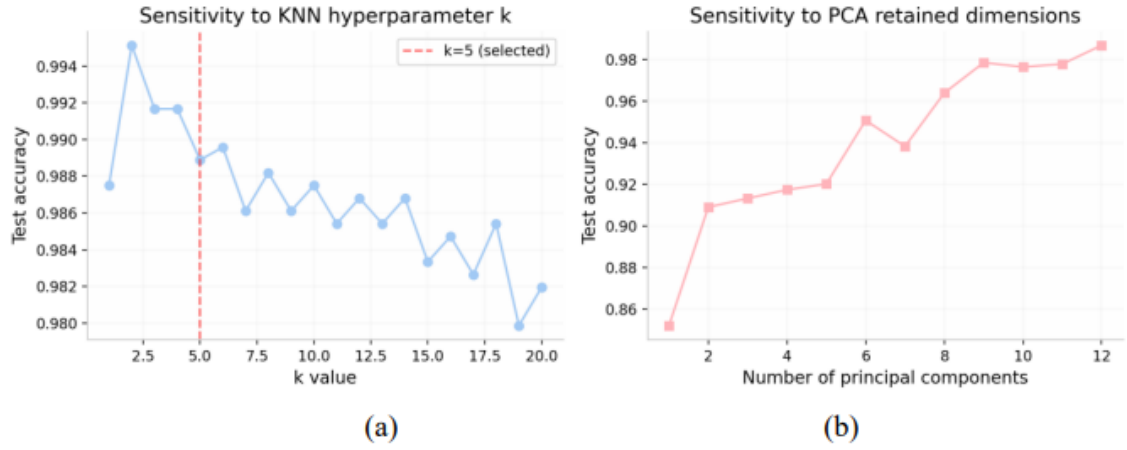

Figure 3. Hyperparameter sensitivity analysis: (a) the k value of KNN; (b) the retained dimension of PCA (picture credit: original)

DOI: 10.54254/2755-2721/2026.37909

The number of neighbors k in KNN and the retained dimension in PCA are two key hyperparameters. Figure 3(a) shows the change in the test accuracy of KNN when k ranges from 1 to 20: The accuracy is higher within a smaller range, and k = 5 is in a stable high-accuracy interval, balancing the smoothing ability for noise and the resolution ability of the decision boundary. Therefore, in this paper, k = 5 is selected. Figure 3(b) shows the test accuracy of KNN when the retained dimension of PCA ranges from 1 to all features: The accuracy increases rapidly with the increase in dimension and tends to saturate after approximately 10 dimensions, which is consistent with the analysis conclusion that the first 30 principal components can explain more than 90% of the variance. This verifies the rationality of the 561 → 30 dimension compression scheme.

4.3. Simulation experiment

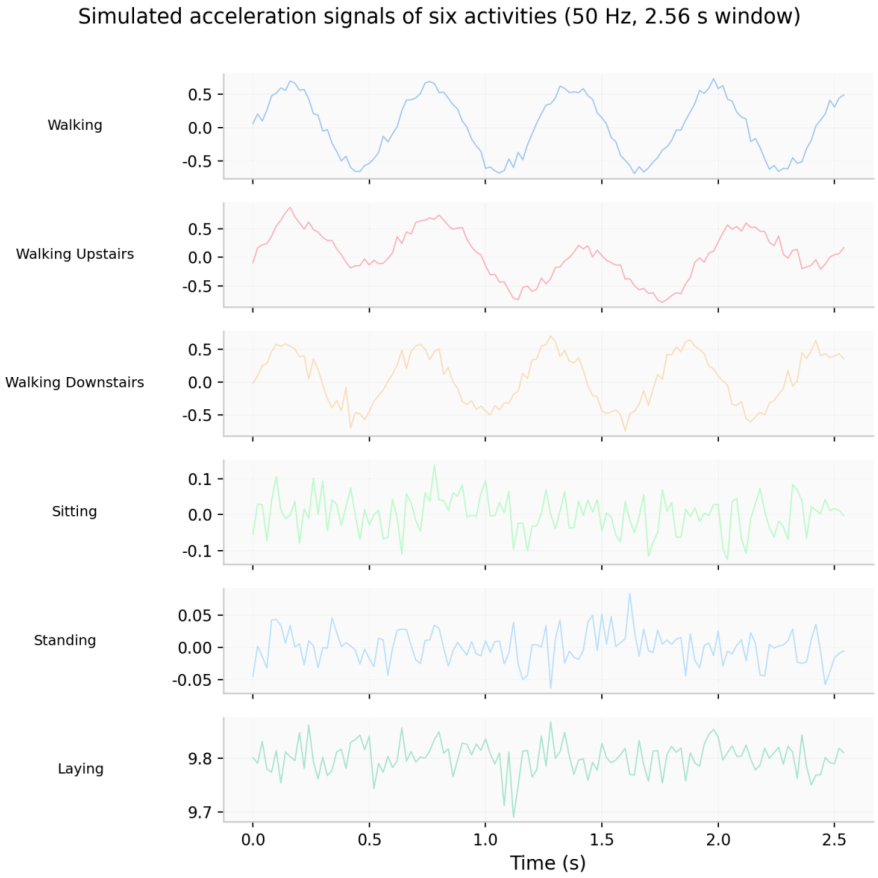

Figure 4. Example of synthesized acceleration signals for six types of activities (picture credit: original)

To verify the feasibility of the end-to-end process of the designed system and to analyze extreme working conditions that cannot be covered by real data, this paper constructs a signal-level simulation platform. The simulation strictly replicates the acquisition and processing conditions of real data: a 2.56-second (128-point) window of three-axis acceleration signal is generated at a 50 Hz sampling frequency. The signal models for six types of activities are as follows: walking uses a periodic signal near the frequency of 1.8 Hz with noise superimposed; going up and down stairs add 0.5 Hz low-frequency components and random impact components on the basis of the walking signal, and the step frequency is respectively shifted to 1.6 Hz and 1.9 Hz; sitting and standing use

DOI: 10.54254/2755-2721/2026.37909

noise signals with decreasing amplitudes and nearly zero mean values; the lying-down signal adds a 9.8 m/s² gravitational bias. Examples of the synthesized signals for each type of activity are shown in Figure 4. Each type of activity generates 800 samples, and the mean, standard deviation, range, root mean square, spectral energy, and main frequency are extracted as statistical quantities to form feature vectors, and the preprocessing and classification process exactly the same as the real data experiment is reused.

The recognition results on the simulation dataset are as follows: Decision Tree 98.2%, KNN (k=5) 98.9%, PCA + KNN 97.8%. These results are consistent with the relative pattern that KNN outperforms the decision tree on the real dataset, verifying the rationality of the selected modeling process. Figure 5(a) examines the impact of sensor noise intensity (1 to 10 times the baseline noise) on the accuracy of KNN: The accuracy remains above 90% even at 10 times the noise level, indicating that the system has strong robustness to sensor noise in actual use. Figure 5(b) shows the simulation confusion matrix, which indicates that the main residual errors are concentrated between going upstairs and going downstairs, consistent with the confusion phenomenon in the real data; while in the simulation, due to the artificially set distinguishable noise amplitude levels, the confusion between sitting and standing is very rare - this provides an indirect evidence that the root cause of the confusion between sitting and standing in the real data lies in the extremely low amplitudes of the two types of signals and their highly similar statistical characteristics.

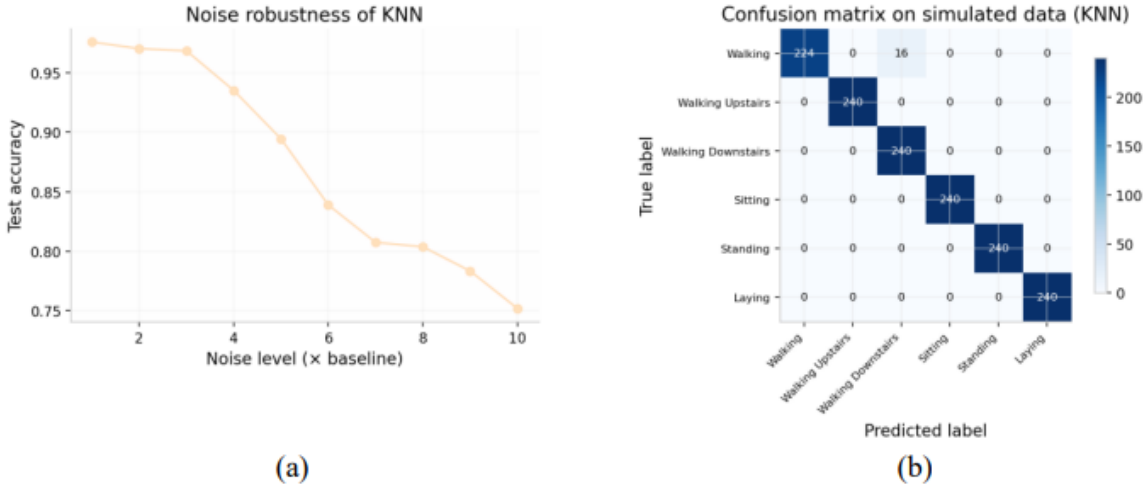

Figure 5. Simulation results: (a) noise robustness; (b) KNN confusion matrix (picture credit: original)

4.4. Accuracy comparison

Table 2 presents the overall performance of the three methods on the test set. The accuracy of the decision tree is 86.29%, although its rules are interpretable, its discriminative ability in 561-dimensional sensor features is limited; the accuracy of KNN based on Euclidean distance reaches 90.16%, which is approximately 3.9 percentage points higher than that of the decision tree. This indicates that for high-dimensional statistical features, similarity matching based on distance metrics has more discriminative power than axis-parallel rule partitioning.

DOI: 10.54254/2755-2721/2026.37909

Table 2. Recognition accuracy rates of each model on the UCI HAR test set

Algorithm	Accuracy Rate (%)
Decision Tree	86.29
KNN (k=5)	90.16
PCA + KNN (30 dimensions)	87.41

The results of PCA + KNN further illustrate the value of dimensionality reduction: after compressing 561-dimensional features to 30 dimensions, the accuracy rate decreased from 90.16% to 87.41%, with only a loss of 2.75 percentage points, while the computational cost of distance calculation decreased by approximately 95%. This means that when deployed on mobile devices such as smartphones, using 30-dimensional features can maintain a high recognition accuracy rate while significantly reducing computational and power consumption costs, which is a reasonable trade-off between accuracy and efficiency.

4.5. Confusion matrix analysis

The overall accuracy may hide the systematic errors of the model on specific activity pairs, so a class-by-class analysis of the confusion matrix is necessary. Figure 6 shows the confusion matrix of KNN on the test set, where the off-diagonal elements reveal the concentrated areas of misclassification.

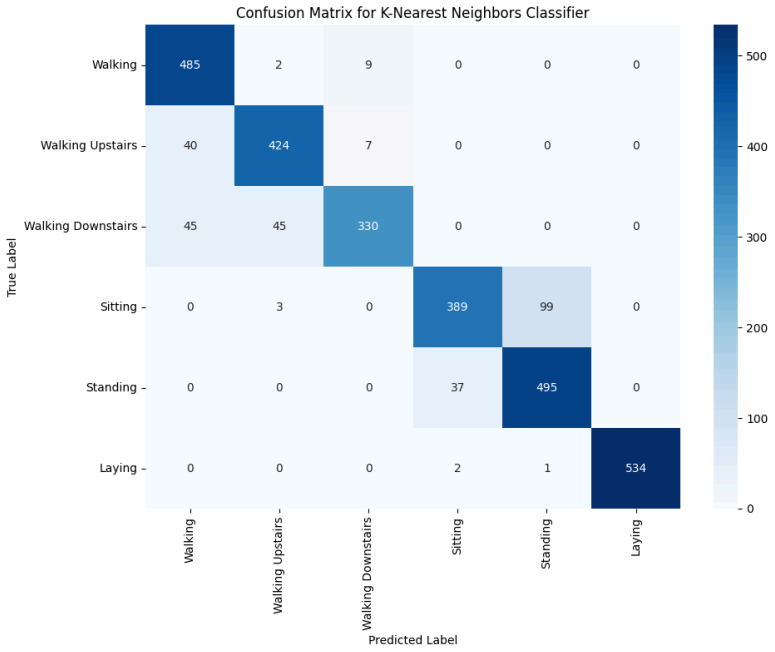

Figure 6. Confusion matrix of KNN on the UCI HAR test set (picture credit: original)

Figure 6 shows the confusion matrix, indicating that the most serious misjudgment occurred in the group of sitting and standing: sitting was misjudged as standing 99 times, and standing was misjudged as sitting 37 times, totaling 136 times, accounting for approximately 66% of all misjudgments (207 times). From the perspective of the signal mechanism, the equipment in both states was approximately stationary, and the accelerometer output was only a very small

DOI: 10.54254/2755-2721/2026.37909

measurement noise with amplitude. The mean, variance, spectral energy and other characteristic statistics were highly similar. The inter-class distance between the two types of samples in the feature space was smaller than the intra-class distance, making it difficult for the KNN's neighbor search to stably distinguish. This result indicates that the recognition ability of only relying on the accelerometer and gyroscope for low-intensity static activities has an inherent upper limit.

The second group of confusions focused on the internal activities of walking: going downstairs was misjudged as walking 45 times, and going upstairs was misjudged as walking 40 times. This is because going downstairs, going upstairs and walking on flat ground share the same gait cycle structure, and their distinction mainly relies on the vertical motion component. However, when the phone is fixed on the waist, the information in this direction in the mixed features has a limited proportion and is insufficient to stably separate the three types of patterns. This suggests that in the future, introducing barometric pressure data and using height change information to enhance the discrimination of vertical motion can be considered.

The laying activity is recognized with almost no error (only 3 misclassifications out of 537 records). The reason is that the projection of the gravitational acceleration in the coordinate system when laying differs significantly from that of the other five activities, so the classes are well separable in the feature space. The above analysis shows that KNN achieves balanced recognition performance on all six activities, with per-class recall rates above 84%.

5. Conclusion

The experimental results demonstrate that local activity recognition using smartphone sensors is feasible: KNN achieved an accuracy rate of 90.16% on the UCI HAR dataset, proving that machine learning models can effectively distinguish six types of daily activities from sensor statistical features; all calculations are completed locally on the device without the need for internet connection, verifying the advantages of the edge computing solution in terms of response speed and privacy protection.

At the same time, the experiment also revealed the obvious shortcomings of the system: the combined misjudgment of sitting and standing reached 136 times, indicating that there is still considerable room for improvement in the identification of low-amplitude static activities relying solely on accelerometers and gyroscopes; the confusion between going downstairs and walking suggests that the extraction of vertical direction motion information is not yet sufficient.

Future improvements can be made in three directions: first, introduce barometric data to distinguish upstairs, downstairs, and flat walking by using height changes; second, try stronger models such as random forests, attention-based networks, and lightweight neural networks to further reduce the misclassification rate between sitting and standing; third, deploy the recognition pipeline on a real smartphone platform and verify it in richer sport scenarios such as running and ball games, instead of being limited to the six basic activities in this paper.

References

[1] Reyes-Ortiz, J. L., Oneto, L., Samà, A., et al. (2013). Human activity recognition using smartphones. *Proceedings of the 21st International European Symposium on Artificial Neural Networks (ESANN)*. Bruges, Belgium, 2013, pp. 1-6.

[2] Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter, 12*(2), 74-82.

[3] Shoaib, M., Bosch, S., Incel, O. D., et al. (2014). Fusion of smartphone motion sensors for physical activity recognition. *Sensors, 14*(6), 10146-10176.

DOI: 10.54254/2755-2721/2026.37909

[4] Ignatov, A. (2018). Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing, 62*, 915-922.
[5] Ramanujam, E., Perumal, T., & Padmavathi, S. (2021). Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. *IEEE Sensors Journal, 21*(12), 13029-13040.
[6] Zhang, S., Li, Y., Zhang, S., et al. (2022). Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors, 22*(4), 1476.
[7] Zhou, Y., Zhao, H., Huang, Y., et al. (2022). TinyHAR: A lightweight deep learning model designed for human activity recognition. *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. Cambridge, UK, Sep. 11-15, pp. 89-93.
[8] Webber, M., & Rojas, R. F. (2021). Human activity recognition with accelerometer and gyroscope: A data fusion approach. *IEEE Sensors Journal, 21*(15), 16979-16989.
[9] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27.
[10] Thakur, D., Guzzo, A., & Fortino, G. (2023). Attention-based multihead deep learning framework for online activity monitoring with smartwatch sensors. *IEEE Internet of Things Journal, 10*(20), 17746-17754.