International Journal of New Developments in Engineering and Society
ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

Classification and comparison of human activities by machine learning

Tiandao Luo[1,*], Jingkai Zhang[2]

[1]*School of Electrical and Control Engineering, Henan Urban Construction University, Pingdingshan, China, 467000*
[2]*School of Information Engineering, Zhengzhou Technology and Business University, Pingdingshan, China, 451400*
[*]*Corresponding author: ltd190615173@163.com*

Abstract: *The current study of human activity recognition and classification has been an important part of promoting the development of science and technology in society. Human activity recognition and classification are in several fields, such as competitive sports, criminal investigation field, etc. As the field of micro-electromechanics continues to evolve, more accurate human recognition is becoming possible, with wearable multi-axis inertial sensors that allow us to visually detect the desired data. In this paper, the data of 19 human activities for 8 testers are feature extracted and normalized. The data are divided into training and test sets by machine learning models: support vector machine (SVR) classification, XGBoost classification, and logistic regression. The experiment was repeated 10 times to take the average value. The models were then scored, and by comparing integrated machine learning with traditional machine learning, it was found that integrated learning improved by 5%−29% in terms of accuracy compared to traditional machine learning.*

Keywords: *Feature Extraction; Principal Component Analysis; Inertial sensors*

1. Introduction

Nowadays, research on the recognition and classification of human activities relies more on cameras, and other devices to capture human activities in motion, for example, in the field of film and television, games and criminal investigation have a wide range of motion capture [3,4]. The three-dimensional data can be a more accurate identification of human activities, but this method for the selection of the machine, the use of environmental, dead-end loss of information problems, and other conditions have requirements, and three-dimensional data is not easy for long-term storage costs are also very high, while the three-axis inertial sensor can be three-dimensional data through a one-dimensional expression, accounting for very little memory and one-dimensional data to facilitate subsequent processing, for the identification of human activities and classification of universal access has great advantages. Three-axis inertial sensor, as shown in Figure 1.

When it comes to the choice of sensors, wearable sensors are the obvious choice for recording human activities. In recent years inertial sensors have been developed and applied in-car navigation, intelligent robot state analysis, and medical and other applications. The development of microelectromechanical technology in recent years has led to a continuous decline in the cost of inertial sensors, of which high-precision low-cost three-axis inertial sensors are increasingly being used by the general public, and inertial accelerometers load a collection of microsensors such as accelerometers, gyroscopes, magnetometers, etc [1] into wearable device species to identify data such as angular velocity for us. This makes it easy to identify human activities using multiple inertial sensors and can be extended to many aspects such as competitive sports and medical care.

In the data obtained through inertial sensors, what machine learning models are used to classify the data after feature extraction, and which one can make the test data more accurate is what we want to explore in this paper, so this paper uses several machine learning models including XGBoost classification, support vector machine classification, and so on. Because of the consideration of arithmetic power and efficiency, we also set up 2 kinds of data after PCA dimensionality reduction and before PCA dimensionality reduction to see if the accuracy rate is similar in the case of reducing computing time to increase efficiency.

ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

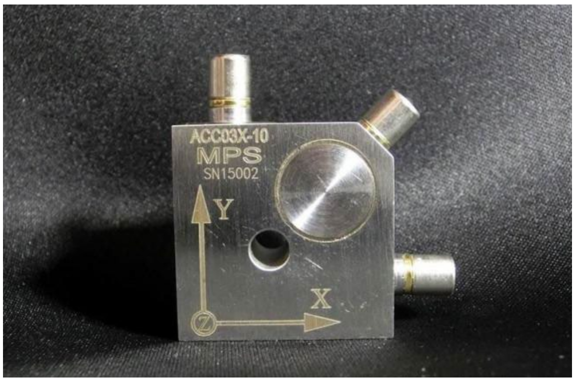

Figure 1: Three-axis inertial sensor

2. Feature Extraction

The human activities used in this paper were 19 movements performed by eight test subjects 4 male and 4 female: sitting (A1); standing (A2); lying flat (A3); lying down on the right side (A4); climbing up stairs (A5); descending stairs (A6); standing in a moving elevator (A7); walking around in an elevator (A8); walking in a parking lot (A9); moving on a treadmill at 4 km/h and remain flat (A10); moving on a treadmill at 4 km/h and maintaining a 15° incline (A11); running on a treadmill at 8 km/h (A12); using a stepper for exercises (A13); using a cross trainer for exercises (A14); riding an exercise bike in a horizontal position (A15); exercising a bike in a vertical position (A16); rowing (A17); jumping jacks (A18); and playing basketball (A19) [2]. Each person wears five inertial sensors each of which performs a 5-minute spontaneous activity sampled at 25Hz, resulting in a 5s segment divided into 60 signal segments. Each sensor is composed of three three-axis devices that collect data from the x, y, and z axes to ensure the reliability of the collected data. Finally, the data is collected into 9120 (19 movements x 8 persons x 60 segments).

After getting the original data, we can extract a total of 23 features from the discrete-time series with 11 categories in the time domain and 12 categories in the frequency domain, as shown in Table 1. After extracting the features, the amount of data is extremely large and confusing for subsequent processing, so this paper converts these features into one line, and generates a column name of length 1035, with a total of 1035 (45cols×23) features, and finally merges 60 feature files into one file.

Table 1: Formula in frequency domain and time domain

Domain	Function	Formula
Time domain	Peak-to-peak	$F_1 = \max(x(i)) - \min(x(i))$
	Average value	$F_2 = \bar{x} = \frac{1}{N_s}\sum_{i=1}^{N_s} x(i)$
	Square root amplitude	$F_3 = \left(\frac{1}{N_s}\sqrt{\sum_{i=1}^{N_s} \lvert x(i)\rvert}\right)^2$
	Variance	$F_4 = \frac{1}{N_s-1}\sum_{i=1}^{N_s}(x(i)-\bar{x})^2$
	Standard deviation	$F_5 = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(x(i)-\bar{x})^2}$
	Kurtosis	$F_6 = \frac{\sum_{n=1}^{N_s}[x(i)-F_2]^3}{(N_s-1)F_5^3}$
	RMS	$F_7 = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(x(i))^2}$
	Skewness	$F_8 = \frac{N_s F_7}{\sum_{n=1}^{N_s}\lvert x(i)\rvert}$
	Waveform factor	$F_9 = \frac{\max\lvert x(i)\rvert}{F_7}$
	Peak factor	$F_{10} = \frac{N_s \max\lvert x(i)\rvert}{\sum_{n=1}^{N_s}\lvert x(i)\rvert}$
	Clearance factor	$F_{11} = \frac{\max\lvert x(i)\rvert}{F_7}$
Frequency domain	Spectrum mean	$F_{12} = \frac{1}{K}\sum_{k=1}^{K} s(k)$
	Root-mean-square value	$F_{13} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}[s(k)-F_{12}]^2}$
	F_{14}	$F_{14} = \frac{\sum_{k=1}^{K}[s(k)-F_{12}]^3}{(K-1)F_{13}^3}$
	F_{15}	$F_{15} = \frac{\sum_{k=1}^{K}[s(k)-F_{12}]^4}{(K-1)F_{13}^4}$
	Frequency center of gravity	$F_{16} = \frac{\sum_{k=1}^{K} f_k \cdot s(k)}{\sum_{k=1}^{K} s(k)}$
	F_{17}	$F_{17} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(f_k - F_{16})^2 \cdot s(k)}$
	Root mean square frequency	$F_{18} = \sqrt{\frac{\sum_{k=1}^{K} f_k^2 \cdot s(k)}{\sum_{k=1}^{K} s(k)}}$
	F_{19}	$F_{19} = \sqrt{\frac{\sum_{k=1}^{K} f_k^4 \cdot s(k)}{\sum_{k=1}^{K} f_k^2 \cdot s(k)}}$
	F_{20}	$F_{20} = \frac{\sum_{k=1}^{K} f_k^2 \cdot s(k)}{\sqrt{\sum_{k=1}^{K} s(k)\sum_{k=1}^{K} f_k^4 \cdot s(k)}}$
	F_{21}	$F_{21} = \frac{F_{17}}{F_{16}}$
	F_{22}	$F_{22} = \frac{\sum_{k=1}^{K}(f_k - F_{16})^3 \cdot s(k)}{(K-1)F_{17}^3}$
	Standard deviation frequency	$F_{23} = \frac{\sum_{k=1}^{K}(f_k - F_{16})^{1/2} \cdot s(k)}{(K-1)F_{17}^{1/2}}$

ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

3. Division of the data set

To perform machine learning on the extracted features the dataset must first be partitioned, generally using two methods leave-out and cross-validation methods.

3.1 Leave-Out Method

The leave-out method is straightforward to divide the data set using stratified sampling. If there is too much data, most of the data is used as the training set and a small portion of the data is used as the test set. This method is easy to operate and computationally small, but it is divided only once and cannot exclude the occurrence of chance results.

3.2 Cross-validation method

There are two ways of cross-validation methods: K-fold cross-validation and leave-one-out method.

K this cross-validation is to split the initial data set into K non-overlapping sub-datasets and then do K training sessions. Each time, one sub-dataset is used to validate the model and the other K-1 sub-datasets are used to train the model. The errors of these K training sessions are then averaged. This method can reduce the occurrence of chance events to improve the accuracy but may miss data resulting in not learning some data with very low accuracy in testing [6]. The leave-one-out method of dividing only once is a special case of K-fold cross-validation, with more accurate results but too high of a computational cost to be put into daily use.

3.3 Method selection

Considering that the cross-validation method may have the problem of data traversal, the leave out method is based on the testers as the classification object to effectively avoid data traversal. The cross-validation method has a great advantage in terms of computational cost. We finally chose the leave-out method as the classification method for our data set partitioning and the leave-out method is based on the testers as the classification object.

4. Classification Technology Discussion

4.1 Integrated Machine Learning

XGBoost is an efficient implementation of the Gradient Boosting Decision Tree, which is a weak classifier that sums the results directly to the predicted value and then fits the next weak classifier to the error of the previous predicted value. Unlike GBDT, XGBoost adds a regularization term to the loss function, significantly improving computational speed and efficiency where the X stands for Extreme.

The essence of XGBoost is to continuously add prediction trees and split the original feature trees to generate new feature trees, to fit the last error, when the number of trees is large enough, the results are predicted more accurately, the prediction tree scores are added up to the prediction value. The formula is as follows:

$$\begin{aligned} Obj &= \sum_{i=1}^{N} L\left[F_m(x_i), y_i\right] + \sum_{j=1}^{m} \Omega\left(f_j\right) \\ &= \sum_{i=1}^{N} L\left[F_{m-1}(x_i) + f_m(x_i), y_i\right] + \sum_{j=1}^{m} \Omega\left(f_j\right) \end{aligned} \tag{1}$$

4.2 Traditional machine learning

4.2.1 Logistic regression

Logistic regression is an extension of linear regression, where the probability of an event is determined by a logistic function that can classify the data. The formula is as follows:

ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

$$J(\theta) = -\frac{1}{m}\left[\sum_{t=1}^{m} y^{(i)} \log h_\theta(x^i) + \sum_{t=1}^{m}(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \quad (2)$$

4.2.2 Linear supportable vector machine classification (SVM)

SVM is a linear classifier built on the basis of VC dimensional theory in the field of statistics to binary classification of data, and its decision boundary is the maximum margin hyperplane solved for the learning sample.

The basic idea of SVM algorithm is data input space mapping data high-dimensional feature space by nonlinear mapping, and then do linear regression in this space, SVM solve problems such as significant advantage small samples, nonlinear and high-dimensional pattern recognition. Using SVM to solve the regression problem has obvious advantages such as strong generalization ability and global best. The formula is as follows:

$$E(w,b) = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, f(x_i)\right) + \alpha R(w) \quad (3)$$

4.2.3 Decision Trees Classification

Decision tree classification is the process of splitting a complex problem into nodes, each of which is a simple problem, and dividing the branches of the tree according to the hierarchy of the problem: a test for an attribute of the instance is specified, it splits the samples arriving at that node according to a particular attribute (e.g., dynamic static), and then populates the branches following that node with the possible values of that node. The plurality of the output variables in the samples contained in the leaf nodes of the classification decision tree is the classification result [5]. The decision tree classification model is formulated as follows:

$$G(Q_m,\theta) = \frac{n_m^{\text{left}}}{n_m} H(Q_m^{\text{left}}(\theta)) + \frac{n_m^{\text{right}}}{n_m} H(Q_m^{\text{right}}(\theta)) \quad (4)$$

5. Experiment

In this paper, the classification effects of each of the four machine learning are compared and the impact of the PCA algorithm in this problem is verified.

5.1 PCA dimensionality reduction

Due to a large amount of extracted data, to simplify the calculation, reduce the overhead of the algorithm, and remove the redundant data, it is necessary to reduce the dimensionality of the data, PCA dimensionality reduction can effectively accomplish this task, PCA dimensionality reduction can be composed of the initial data matrix, each row of zero mean processing, resulting in a covariance matrix, calculate the eigenvalues and eigenvectors of the matrix, through the eigenvectors by finding the eigenvalues from small to large By arranging the eigenvectors in the order of finding the eigenvalues from smallest to largest into a matrix taking the first n rows to form a new matrix, it is possible to do the dimensionality reduction to n dimensions. In this paper, the data features are reduced to 95% of the data features by PCA dimensionality reduction, and the data features are reduced from 1036 dimensions to 30 dimensions.

5.2 Classification results for each model

To make the experimental results more accurate and avoid chance events as much as possible, each model in this paper was experimented with 10 times, and at the same time, to compare the computing time in integrated machine learning, 2 sets of data were set after PCA downscaling and before PCA downscaling to determine whether the computing efficiency was improved by how much with similar accuracy. The experimental results are shown in Table 2, and Table 3.

ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

Table 2: Traditional machine learning.

2*Number	Logistic Regression				SVM				DT			
	Precision	Recall	F1-score	accuracy	Precision	Recall	F1-score	accuracy	Precision	Recall	F1-score	accuracy
1	0.00	0.00	0.00		0.70	0.67	0.68		0.00	0.00	0.00	
2	0.25	0.90	0.40		0.82	0.70	0.76		0.00	0.00	0.00	
3	0.00	0.00	0.00		0.95	0.67	0.78		1.00	1.00	1.00	
4	0.95	0.90	0.92		0.95	0.65	0.77		1.00	1.00	1.00	
5	0.95	0.90	0.92		0.98	0.70	0.82		0.92	0.690	0.73	
6	0.73	1.00	0.85		1.00	0.98	0.99		0.68	0.87	0.76	
7	0.79	0.52	0.63		0.62	0.52	0.56		0.53	0.13	0.21	
8	0.69	0.30	0.42		0.39	0.80	0.52		0.23	0.78	0.36	
9	0.87	1.00	0.93		0.74	0.88	0.80		0.00	0.00	0.00	
10	0.94	0.98	0.96	0.74	0.89	0.95	0.92	0.84	0.44	0.48	0.46	0.61
11	0.98	0.77	0.86		1.00	0.70	0.82		0.67	0.03	0.06	
12	0.98	1.00	0.99		1.00	0.70	0.82		0.00	0.00	0.00	
13	0.98	1.00	0.99		1.00	1.00	1.00		0.90	0.88	0.89	
14	0.89	0.98	0.94		1.00	1.00	1.00		0.58	0.93	0.72	
15	0.92	0.98	0.95		1.00	0.95	0.97		1.00	1.00	1.00	
16	1.00	0.75	0.86		0.98	1.00	0.99		0.48	1.00	0.65	
17	0.97	1.00	0.98		0.98	1.00	0.99		1.00	1.00	1.00	
18	0.48	1.00	0.65		0.94	1.00	0.97		0.94	1.00	0.97	
19	0.98	0.98	0.98		0.98	1.00	0.99		0.45	0.90	0.60	

Table 3: Integrated Machine Learning.

2*Number	XGBoost				PCA-XGBoost			
	Precision	Recall	F1-score	accuracy	Precision	Recall	F1-score	accuracy
1	0.95	0.95	0.95		0.76	0.93	0.84	
2	0.87	0.67	0.75		0.87	0.92	0.89	
3	1.00	1.00	1.00		0.92	0.57	0.70	
4	0.70	1.00	0.82		1.00	1.00	1.00	
5	0.91	1.00	0.95		0.67	1.00	0.81	
6	0.98	1.00	0.99		0.80	0.98	0.88	
7	0.33	0.10	0.15		1.00	0.03	0.06	
8	0.49	0.87	0.63		0.59	0.63	0.61	
9	1.00	0.88	0.94		1.00	0.15	0.26	
10	0.95	1.00	0.03	0.89	0.51	1.00	0.68	0.90
11	1.00	0.02	0.03		0.06	0.05	0.05	
12	1.00	1.00	1.00		1.00	1.00	1.00	
13	0.47	0.97	0.63		0.91	1.00	0.95	
14	1.00	0.98	0.99		0.82	1.00	0.90	
15	1.00	1.00	1.00		0.98	0.95	0.97	
16	1.00	0.53	0.70		0.98	0.90	0.94	
17	1.00	1.00	1.00		0.97	0.98	0.98	
18	1.00	1.00	1.00		0.97	1.00	0.98	
19	0.91	1.00	0.95		0.94	0.98	0.96	

5.3 Integrated machine learning versus traditional machine learning

1) Among the traditional machine learning classification models decision tree classification (DT) is the worst performer. From Table 2, it is clear that the accuracy of decision tree classification in 19 actions A1, A2, A12, recall and F1-score data are all 0. The average accuracy of 10 experiments is only 61%, which does not apply to the classification of human activities.

2) The various data of actions A1 and A3 in the logistic regression are also 0, but the rest of the actions are more accurate in all aspects such as accuracy. The average accuracy of the logistic regression for 10 experiments is 74%, which still does not reach the standard error rate for normal use.

ISSN 2522-3488 Vol. 7, Issue 4: 11-16, DOI: 10.25236/IJNDES.2023.070403

3) Linear supportable vector machine classification (SVM) is the best performer among these traditional machine learning classifications, without the problem that some actions cannot be judged, with high performance in accuracy, recall, and F1-score, and an average accuracy of 84% in 10 experiments, which is usable for human activity classification.

4) The integrated machine learning classification model used in the XGBoost classification from Table 2 can be seen in the action A7 accuracy is 0.33, recall is 0.10, F1-score is 0.15 lower but in other action recognition by very high accuracy, the average accuracy of 10 experiments reached 89% reached the normal classification of human activities.

5) Compared with the traditional machine learning classification integrated machine learning classification XGBoost has better performance in terms of accuracy, recall, and F1-score, and surpasses the traditional classification model in terms of average accuracy.

5.4 Before and after PCA downscaling comparison

As can be seen from Table 3, the accuracy, recall, and F1-score of XGBoost before and after PCA downscaling do not fluctuate much, and the average accuracy of 10 experiments after PCA downscaling is 89% unchanged, but in the experiment without PCA downscaling, the experiment requires 8.4s of running time for each XGBoost classification, while after PCA downscaling After PCA dimensionality reduction, each experiment only takes 5.3s, and the time required for PCA dimensionality reduction is 0.8s, which can speed up 26.5% in each experiment. It can save computational costs and improve computing efficiency when the data volume is larger.

6. Conclusion

In this paper, we propose how to extract and process the feature data from inertial sensors to classify the dataset by the leave-one-out method, and compare the classification models of traditional and integrated machine learning in terms of accuracy, recall, F1-score, and average accuracy. It is found that compared with the traditional classification model, the XGBoost classification with integrated machine learning has the feature of high accuracy and stable classification, which is suitable for classifying human activities, and the PCA dimensionality reduction can greatly reduce the computational cost and make the data processing faster and the data storage space can be reduced.

In this paper, we have found a suitable method to identify and classify human activities, but we have not achieved more perfect results due to the computational cost and the size of the sample data, and there is still room for improvement in the accuracy rate, so we hope to further improve the accuracy rate in the next research.

References

[1] Wei Xuanxuan. Human Posture Recognition Algorithm Based on Inertial Sensor. Intelligent computers and applications. 2022

[2] Kerem Altun, Billur Barshan, Orkun Tunc¸ et el. Comparative Study on Classifying Human Activities with Miniature Inertial and Magnetic Sensors [J]. Pattern Recognition, 2010, 43(10).

[3] Huang Jinjin. Research on Signal Compression and Motion Recognition in Wireless somatosensory Network for Human Motion Monitoring [D]. Zhejiang University of Technology, 2014.

[4] Setthanun Thongsuwan, Saichon Jaiyen, Anantachai Padcharoen, Praveen Agarwal. ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost [J]. Nuclear Engineering and Technology, 2020, 53(2).

[5] Xue Yang. Human Motion Pattern Recognition Based on Single Acceleration Sensor [D]. South China University of Technology, 2011.

[6] Roy Jagabandhu, Saha Sunil. Ensemble hybrid machine learning methods for gully erosion susceptibility mapping: K-fold cross validation approach [J]. Artificial Intelligence in Geosciences, 2022, 3.