International Journal of Scientific Research in Multidisciplinary Studies
Research Paper
Vol.8, Issue.9, pp.80-84, September (2022)
E-ISSN: 2454-9312
P-ISSN: 2454-6143

Identification of Different Human Actions through Smart Phone Data

Deep Kumar Bangotra

Department of Higher Education, J&K Govt.India-180016

Author's Mail Id: deepbangotra.ap@gmail.com

Available online at: www.isroset.org

Received: 28/Apr/2022, Accepted: 20/Jul/2022, Online: 30/Sept/2022

Abstract- The identification of various human activities utilising data generated from a user's smart phone is presented in this study. This study uses data from the University of California Machine Learning Repository to identify six human activities. These actions include lying down, sitting down, standing up, walking, and walking both upstairs and downstairs. The Samsung Galaxy S II smart phone's inbuilt gyroscope, accelerometer, and other sensors are used to gather the data. To arrange the training and testing data sets, the data is randomly split into 7:3 ratios. The Principal Component Analysis method is used to reduce the dimensions of the data. Different Machine Learning models, such the Artificial Neural Network, Random Forest, K-Nearest Neighbor, and Support Vector Machine, are used to categorise activity. Using a confusion matrix and random simulation, a comparative examination of these models' performance and accuracy has been presented in this research paper.

Keywords- Random Forests, Artificial Neural Networks, k-Nearest Neighbor, Human Activity Recognition, Support Vector Machine, Principal Component Analysis.

I. INTRODUCTION

Investigating human behavior due of its numerous uses in the health care industry, computer vision, home safety, and robot learning, recognition is in high demand[1]. If sensors collect and monitor patient data in the healthcare industry, significant financial savings may be made. Reports may be provided to doctors automatically if any odd activity is discovered in the patient's health information. To detect human activity, we have employed low-cost sensors that are already present in smartphones. Smartphones are the perfect option for non-invasive body-attached sensors due to their enormous development in popularity, accessibility, and processing power[2]. Smartphones have integrated deeply into daily life. People use their smartphones constantly throughout the day. This enables smartphone built-in sensors to gather information, enabling the system to recognise human behaviour.

The University of California Machine Learning Repository is where the data came from [3]. Imported, cleansed, and normalised data are used. To improve our system's accuracy and performance, the dimensions of the original dataset were lowered using the Principal Component Analysis (PCA) method. The cleaned data is then classified into six categories, including lying, sitting, standing, walking, and walking upstairs and downstairs, using supervised machine learning algorithms including Support Vector Machine, Random Forest, K-Nearest Neighbor, and Artificial Neural Network.

Our goal is to create a flexible and straightforward technology that can detect human activity. By lowering the amount of dimensions in the dataset, we have focused on increasing accuracy and speeding up model training. The model that performs and accurately matches our system has been determined through comparison of many models.
The remaining section of this work is structured as follows:

The background and earlier research projects in the domain of human activity recognition are described in Section 2 of this article. The descriptions of methodology and system design can be found in Section 3. In Section 4, the performances of various machine learning algorithms were compared, and Section 5 draws a conclusion from the analysis.

II. BACKGROUND

Human activity recognition has been the subject of extensive study for many years, and numerous approaches have been put out to address the issue. The application of threshold techniques and machine learning has been used to recognition tasks. While machine learning algorithms are accurate and dependable, threshold methods are quicker and easier to use. Inertial, vision, or a combination of the two sensors can be used to collect data [4]. Attaching several accelerometers and gyroscopes to various body locations is the most popular and straightforward technique [5][6]. These sensors' data collection is very expensive and complex in terms of the various parameters used to characterize human activity.

Additionally, these characteristics are related to one another and do not each separately aid in the identification of an activity.

The paper [7] examines data obtained from a smartphone's acceleration sensor. The data was produced by 29 people who had an Android phone in their pockets. They had to walk, run, climb stairs, descend stairs, sit, and stand. J48, Multilayer Perceptron, and Logistic Regression were the algorithms that were tested. 90% of the time, their results were accurate, except when it came to uphill vs. downhill walking. According to the results, J48 was more accurate at recognizing other activities, while neural networks performed best at detecting jogging and climbing stairs [8].

We have gathered data using integrated sensors in smartphones. These cellphones are easily accessible and reasonably priced in the market. In order to increase the effectiveness of machine learning models, the author of this research employed Principal Component Analysis to decrease the amount of features in the data and make it simpler.

III. SYSTEM DESIGN

This section will outline the method we utilised to design our system, step by step. The machine learning problem we are working on is a classification challenge. To conduct the full experiment, the author have used R Studio version 3.2.5 on a Windows 10 platform.

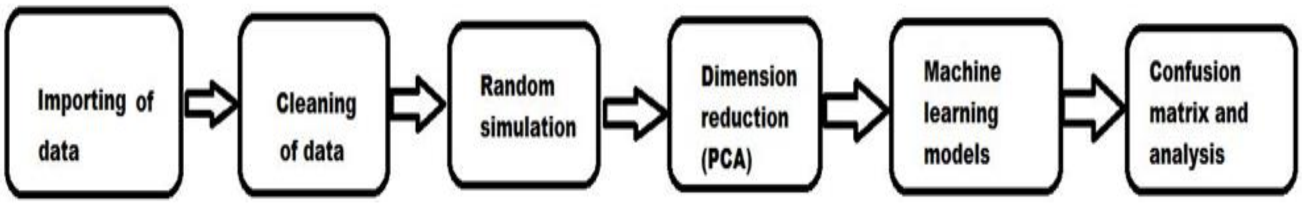

Figure 1. The flow of system design of Human Activity recognition system.

The different steps during the classification are explained below:

3.1 Importing and Cleaning Data
Dataset was imported as a comma separated file into R studio. 586 characteristics made up the imported dataset. Header names for features were altered and used. The mean was used to impute missing data for that specific attribute.

The following formula was used to standardize the data:

$$zi = \frac{xi - \bar{x}}{sd}$$

Where x_i is the data instance, $\bar{x}$ is mean of that feature column and sd is feature column ‘s standard deviation.

3.2 Random Simulation
To evaluate the precision of predictive models and avoid data over- and underfitting, random simulation is performed. Using a random process, the dataset is split into training and testing sets in the proportion of 7:3. To increase the model's accuracy in accordance with the statistical Central limit theorem [9][10], the entire simulation is run 50 times. The testing dataset gives us a way to assess the stability of our model in a real-world setting while also providing an approximation of real-time data.

3.3 Dimensionality Reduction
We have less features in our dataset thanks to Principal Component Analysis. The principle component analysis (PCA) is intended to keep as much of the data set's variance as feasible while reducing the dimensionality of big data sets with numerous connected variables. When performing principal component analysis, the variables in the input dataset are transformed into a new collection of variables called the principal components (PCs). The Principal Components are arranged according to the variation found in each of the original variables and are uncorrelated. The majority of the variation found in the original dataset is contained in the first few components of this ordered collection of main components. The transformation of the data has been shown in Figure 2.

Since the first 100 principal components in the picture explain more than 95% of the variance in the dataset, we just use those 100 and ignore the other components. The training dataset underwent alterations, while the testing dataset underwent the exact same transformation.

3.4 Random Forest
An ensemble learning technique for classification, regression, and other problems is called Random Forest [11]. It maneuvers by building a hefty number of decision trees during training and then producing the class that is the mean forecast of the decision trees. Overfitting of the data is avoided by random forest[12]. The default setting in the R programming environment creates 500 decision trees. We discovered through experimentation that there was no need to build 500 decision trees. If we had just built 80 trees, as is depicted in the figure 3, our model would still operate and yield the same results. We only require 80 decision trees because the figure demonstrates that the mistake rate remains constant after the creation of 80 trees.

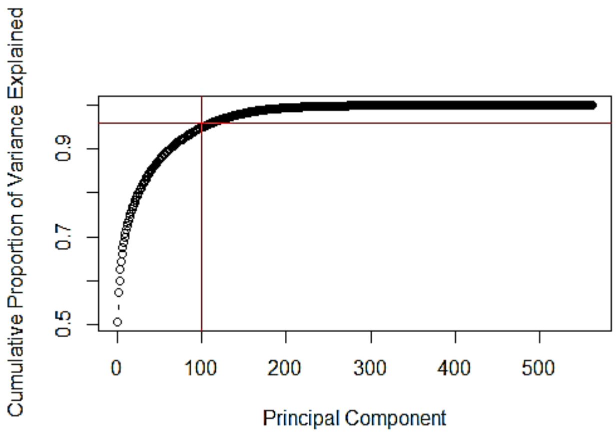

Figure 2. Cumulative Proportions of Variances

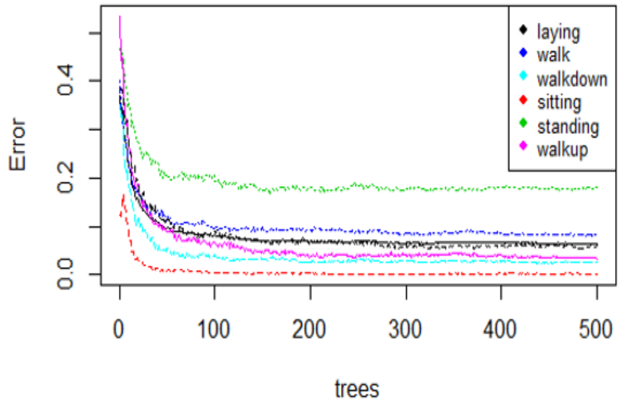

Figure 3. Error rate measurements with respect to Number of Trees

3.5 K-Nearest Neighbor

KNN is a classifier that uses instances. It is based on the idea that classifying unknown instances can be accomplished by connecting an unknown instance to a known instance via some function[13]. This function is either a distance or similarity function. To approximate our learning function, we employed a Euclidean distance function. As can be seen in figure 4 below, we plotted a graph of error rate vs. K value to calculate the value of K.

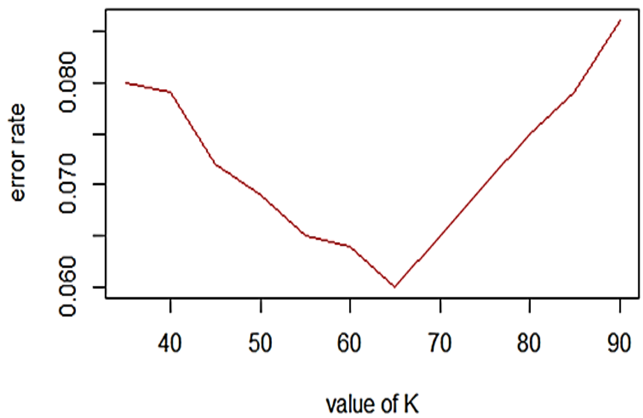

Figure 4. Variation of error rate with respect to the value of K

The error rate drops from K=40 to K=65 and achieves a minimum at K=65, according to the figure 4.

3.6 Support Vector Machine (SVM)

Decision hyperplanes, which specify decision boundaries, are the foundation of support vector machines. A decision plane divides two sets of items into those belonging to different classes. To maximise the decision boundary between hyperplanes, SVMs are used. We trained our dataset using the "e1071" SVM library in R [14]. Kernel type, cost, and gamma are parameters for the SVM algorithm. The similarity function that we selected was the Gaussian (radical) kernel.

Since it keeps the regularisation term constant and prevents data overfitting, the value of the cost is fixed at 1. The vector hyperplane's shape is determined by the gamma value. Given that gamma's value is equal to 1, it is left at 0.013758. (number of features).

3.7 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) use a model developed from our understanding of how a real brain responds to stimuli from inputs to represent the relationship between a set of input signals and an output signal [15].Our dataset was trained using the nnet package [20], which is designed to train feed-forward neural networks with a single hidden layer. The artificial neural network is trained using the backpropagation method using the nnet programme. This approach determines output error before propagating it throughout the network. The weights are modified [15] to reduce each neuron's mistake as much as possible. We are dealing with a classification problem, hence the default setting for the Linout argument is False. The difference in error rate with regard to the amount of nodes in the hidden layer is seen in Figure 5.

Figure 5 shows that the error rate is at its lowest when there are 6 nodes in the hidden layer. As a result, we've given the size parameter a value of 6.

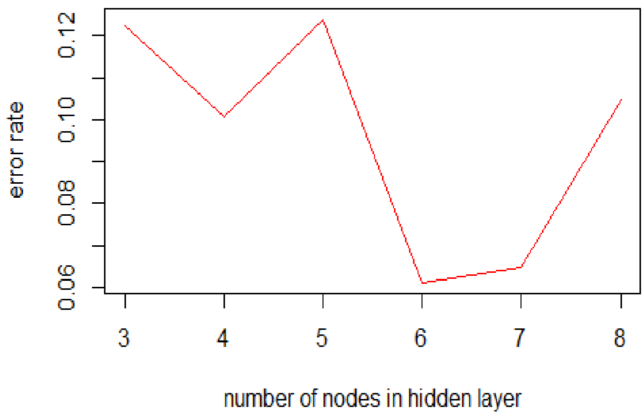

Figure 5. Variation of error rate in the hidden layer

IV. PERFORMANCE ANALYSIS

Based on the confusion matrix and the period it took to train the model, we examined the effectiveness of various machine learning models in this section. The model's training time is determined by taking the average of 30 simulations.

4.1 Random Forest

Table 1: Confusion matrix of data tested with Random Forest Model

Result/ References	**Laying**	**Sitting**	**Standing**	**Walk**	**Walk up**	**Walk down**
Laying	420	0	0	0	0	0
Sitting	14	326	52	0	0	0
Standing	0	35	373	0	0	0
Walk	0	0	0	368	3	2
Walk up	0	0	0	2	284	6
Walkdown	0	0	0	7	17	297

The confusion matrix shows that the percentage success rate is 93.75%, while the percentage error rate is 6.25%. The model was trained in 10.56 seconds.

4.2 *K-Nearest Neighbors (KNN)*

Table 2: Confusion matrix of data tested with K Nearest Neighbour Model

Result/ References	**Laying**	**Sitting**	**Standing**	**Walk**	**Walk up**	**Walk down**
Laying	421	4	0	0	0	0
Sitting	1	269	23	0	0	0
Standing	10	104	407	0	0	0
Walk	0	0	0	365	17	8
Walk up	0	0	0	7	259	5
Walk down	2	0	0	1	16	287

The confusion matrix shows that the percentage success rate is 91.02%, while the percentage inaccuracy is 8.98%.The model needs 2.7 seconds to train.

4.3 *Support Vector Machine (SVM)*

Table 3: Confusion matrix of data tested with Support Vector Machine Model

Result/ References	**Laying**	**Sitting**	**Standing**	**Walk**	**Walk up**	**Walk down**
Laying	429	0	0	0	5	0
Sitting	0	333	42	0	2	0
Standing	0	35	393	0	2	0
Walk	0	0	0	362	11	0
Walk up	0	0	0	0	292	0
Walk down	0	0	0	0	5	295

The confusion matrix shows that the percentage success rate is 95.37%, while the percentage mistake is 4.63%.The model training process takes 12.9 seconds.

4.4 Artificial neural network (ANN)

Table 4: Confusion matrix of data tested with Artificial Neural Network

Result/ References	**Laying**	**Sitting**	**Standing**	**Walk**	**Walk up**	**Walk down**
Laying	432	0	2	0	0	0
Sitting	8	318	50	1	0	0
Standing	1	48	374	3	0	4
Walk	0	0	0	361	3	9
Walk up	0	1	0	4	279	8
Walk down	0	0	0	2	5	293

The confusion matrix shows that the percentage success rate is 93.24%, while the percentage mistake is 6.76%. The model needs 5.2 seconds to train.

V. CONCLUSION

In order to learn machine learning algorithms, we investigated the Human Activity Recognition dataset and acquired knowledge of Random Forest, K-Nearest Neighbor, Support Vector Machine, and Artificial Neural Networks. Using Principal Component Analysis, we were able to moderate the dimensions of our dataset from 586 features to 100 features. After conducting data analysis, we discovered that Support Vector Machines were the most effective at predicting human activity (95.37%). Because we utilized a Gaussian kernel in the smaller dataset, SVM is the most effective method. Due to its simplicity and usage of the Euclidian distance function, K-Nearest Neighbour required the least amount of time to train (2.7 seconds). Support Vector Machine took the most time to train the model (12.9 seconds).

REFERENCES

[1] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Transactions on Information Technology in Biomedicine*, **vol. 14, no. 5, pp. 1166–1172, 2010.**

[2] "Smartphone explosion in 2014 will see ownership in India pass US | Smartphones | The Guardian." [Online]. Available: https://www.theguardian.com/technology/2014/jan/13/smartphone-explosion-2014-india-us-china-firefoxos-android. [Accessed: 08-Oct-2022].

[3] "UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones. [Accessed: 08-Oct-2022].

[4] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2–3 SPEC. ISS., **pp. 90–126, 2006.**

[5] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **vol. 3968 LNCS, pp. 1–16, 2006.**

[6] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **vol. 3001, pp. 1–17, 2004.**

[7] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," *IEEE 4th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2010*, **2010.**

[8] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, **vol. 12, no. 2, pp. 74–82, 2011.**

[9] "Central Limit Theorem (CLT) Definition." [Online]. Available: https://www.investopedia.com/terms/c/central_limit_theorem.asp. [Accessed: 08-Oct-2022].

[10] I. T. Jollife and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **vol. 374, no. 2065, p. 20150202, Apr. 2016.**

[11] A. Salai and T. Nadu, "THE EFFICIENCY OF RANDOM FOREST ALGORITHM IN BIG DATA ANALYTICS FOR," pp. 200–204.

[12] S. Das, "Human Activity Recognition using Machine Learning," *International Journal for Research in Applied Science and Engineering Technology*, **vol. 10, no. 6, pp. 4188–4193, 2022.**

[13] B. V. Dasarathy, "Nearest neighbor (NN) norms : nn pattern classification techniques," **p. 447, 1991.**

[14] "Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien [R package e1071 version 1.7-11]," **Jun. 2022.**

[15] Y. Electronics, O. Source, A. N. Networks, and A. Ann, "Introduction to Artificial Neural Networks (ANN)," no. February, **pp. 1–5, 2009.**