Artificial Intelligence Prediction of Human Activity Recognition

Archit Jain[1], Aayush Ranjan[2], Mr. Ajay Kaushik[3]

[1]Student, Dept of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India
[2]Student, Dept of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India
[3]Assistant Professor, Dept of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India

---***---

Abstract - *Wearable computing is becoming more and more incorporated into our daily lives. Wearable gadgets have recently attracted a lot of attention and widespread acceptance as a result of their compact size and decent processing power capabilities. These wearable gadgets with sensors (e.g. accelerometer, gyroscope, etc.) are excellent choices for tracking users' daily activities (e.g. walking, jogging, sleeping, and so on). Human Activity Recognition (HAR) has the potential to aid in the development of assistive technologies for the elderly, chronically ill, and those with special needs. Activity recognition can be used to offer information on patients' daily activities in order to aid the development of e-health systems such as Ambient Assisted Living (AAL). Despite the fact that human activity detection has been an active field for the development of context-aware systems for more than a decade, there are still critical issues that, if addressed, would represent a dramatic shift in how people interact with smartphones. A broad architecture of the essential components of any HAR system is described, as well as a data-gathering architecture for HAR systems. Machine learning techniques and technologies were used by HAR systems to generate patterns to characterize, evaluate, and predict data. Because a human activity recognition system should return a label such as walking, sitting, running, sleeping, falling, and so on, most HAR systems are supervised. The goal of this research is to use multiple machine learning methods on the UCI Human Activity Recognition dataset.*

Key Words: **Human Activity Recognition, Artificial Intelligence, Wearable gadgets, HAR, Dense Layer**

1.INTRODUCTION

Machine Learning is “the study of computer algorithms that improve automatically through experience. Applications range from data mining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests.” - IEEE Definition.

The IT industry's new dark is artificial intelligence (AI) and machine learning (ML). From gaming consoles to the administration of vast volumes of data at work, AI can be found in a variety of locations. Computer scientists and engineers are working hard to program machines with intelligent behavior, allowing them to think and react in real-time. AI has moved from a research topic to a stage where it is being implemented in businesses. The science and engineering of constructing intelligent devices, particularly computer programs, is known as artificial intelligence (AI). Artificial Intellect is akin to the job of utilizing computers to study human intelligence, although AI does not have to be limited to physiologically observable ways. Simply expressed, AI's goal is to make computers/computer programs clever enough to mimic the behavior of the human mind. Knowledge engineering research is an important part of AI research. In order to function and behave like people, machines and programs require a lot of knowledge about the world. To perform knowledge engineering, AI needs to have access to attributes, categories, objects, and the relationships between them. AI imbues machines with common sense, problem-solving, and analytical thinking abilities, which is a challenging and time-consuming task. Developers continue to increase artificial intelligence's powers and potential, despite ongoing disputes about its safety. Artificial Intelligence has come a long way since it was first imagined in science fiction. It became a requirement. AI, which is commonly used for processing and analyzing large amounts of data, aids in the handling of tasks that are no longer possible to complete manually due to their increased volume and intensity. Wearable devices, such as smartwatches, Google glasses, fitness trackers, sports watches, smart clothing, smart jewelry, implantable devices, and others, have recently attracted a lot of attention and widespread acceptance due to their small size, reasonable computation power, and practical power capabilities. These wearable gadgets, which are equipped with sensors (such as an accelerometer and a gyroscope), are a strong choice for tracking users' daily activities (such as walking, jogging, and smoking). Wearable and non-intrusive technologies for health and activity monitoring have become more common as wearable technology has advanced. The users are motivated to maintain a healthy lifestyle as a result of the continual monitoring of their lives and everyday activities.

2. HUMAN ACTIVITY RECOGNITION

Human activity recognition has become a hot topic in the medical, military, and security fields. Patients with diabetes, obesity, or heart disease, for example, are frequently asked to adhere to a strict exercise regimen as part of their treatment [1]. HAR has the potential to improve the creation of assistive devices for the elderly, chronically ill, tracking energy expenditure and weight reduction, digital assistants for weight lifting activities, and persons with special needs. The following is an example of smart homes being used to identify and analyze health occurrences.

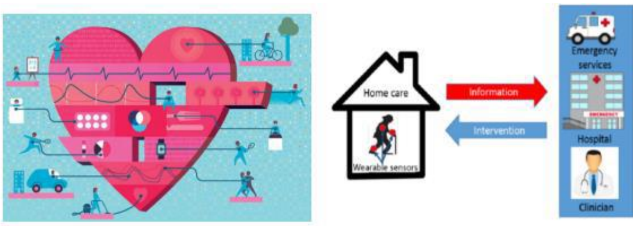


Fig -1: Smart Home-based Health Data Analysis

Using non-intrusive wearable sensors, the home supporting environment provides trend data and event detection. This allows for speedy measuring while also allowing for swift acceptance. Healthcare providers will be able to monitor the subject's motions throughout daily activities and detect unpredictable events, such as a fall, using real-time processing and data transmission. The records of the subjects can be used in medical decision-making, as well as in the prediction and prevention of accidents [2-4]. Image processing with computer vision and the use of wearable sensors are the two most popular ways for HAR. The image processing approach does not require the user to wear any equipment, but it does have certain constraints, such as limiting operation to indoor conditions, necessitating camera installation in every room, illumination and image quality difficulties, and user privacy. Wearable sensors, on the other hand, reduce these issues, despite the fact that they need users to wear the equipment for long periods of time. As a result, using wearable sensors may cause issues with battery charging, sensor placement, and sensor calibration.

3. METHODOLOGY

3.1 Data Collection

- The accuracy of our model is determined by the volume and quality of your data.
- This stage usually results in a knowledge illustration (Guo reduces this to providing a table) that we'll utilize for training.
- Victimization datasets from UCI, UCI, and other sources can still be used in this stage if they have already been collected.

3.2 Data Preparation

- Gather information and organize it in order to coach others.
- Remove duplicates, fix errors, change missing values, standardize data, convert data types, and so on.)
- Visualize knowledge to assist in finding relevant links between variables or category imbalances (bias alert!) or undertake alternative searching analysis by randomizing knowledge.
- Coaching and analysis sets have been separated.

3.3 Choose a Model

- Different algorithms are used for different jobs; choose the one that is right for you.

3.4 Train the Model

- The purpose of training is to get a model to correctly answer a question or make a forecast as often as possible.
- Each repetition of the technique may be a coaching step in the case of linear regression: an algorithmic program would seek to identify values type (or W) and b (x is input, y is output).

3.5 Evaluate the Model

- Uses a metric or a combination of metrics to "measure" the model's objective performance Tests the model against previously unseen knowledge
- This previously unseen knowledge is supposed to be somewhat representative of model performance in the real world, but it still aids in tuning the model (as hostile testing knowledge, that will not)
- Is there a good train/eval split? 80/20, 70/30, or comparable ratios are also applied when considering the domain, knowledge convenience, dataset particular, and alternative considerations.

3.6 Parameter standardization

- Tune model parameters for enhanced performance
- Simple model hyperparameters could include coaching steps, learning rate, formatting values, distribution, and so on.

3.7 Create Predictions

- Using more (test set) knowledge that has been hidden from the model up to this point (and that category labels are known), are likely to look at the model; a lot of proper illustration of how the model can behave in the real world.

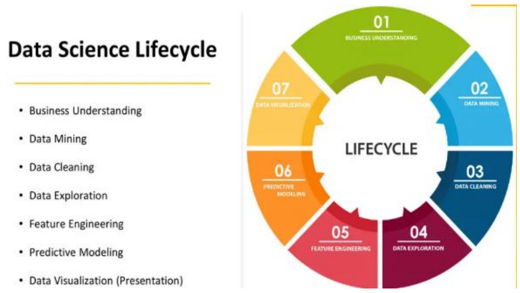


Fig -2: Data Science Project Lifecycle

Machine learning (ML) and Artificial Intelligence (AI) technologies are used in HAR systems to develop patterns that may be used to characterize, evaluate, and predict data [5]. It's used to categorize activity recognition errors. Patterns must be discovered from a set of given examples or observations termed instances in a machine learning scenario. This type of input set is referred to as a training set. Each instance is a feature vector derived from signals collected over a period of time. The examples in the training set may or may not be labeled, that is, allocated to a certain category (e.g., walking, running, sleeping, etc.). Labeling data is not always possible because it may necessitate an expert personally examining the instances and assigning a label based on their knowledge. Many data mining solutions make this procedure difficult, expensive, and time-consuming.

4. IMPLEMENTATION AND RESULTS

Most HAR systems work in a supervised manner since a human activity recognition system should return a label such as walking, sitting, running, and so on. In a wholly unsupervised environment, it may be difficult to distinguish between activities. Certain systems operate in a semi-supervised model, allowing some data to remain unlabeled. This is a matrix such that the element Mij is the number of instances from class i that was actually classified as class j. In a binary classification task, the confusion matrix can yield the following values:

True Positives (TP): The total number of Class A activities classified as such.

True Negatives (TN): The number of Non-Class A activities that were classified as Non-Class A.

False Positives (FP): The number of Non-Class A activities that were classified as Class A.

False Negatives (FN): The number of Class A activities that were classified as Non-Class A.

The accuracy measure is the most common way, to sum up, total classification performance across all classes, and it is defined as follows:

$$\text{Accuracy} =(TP+TN)/(TP+TN+FP+FN)$$

Apart from accuracy, there are three more statistical metrics that can judge the performance of a classification machine learning model. The metrics are as follows:

1. Precision refers to the number of true positives divided by the total number of positive predictions (i.e., the number of true positives plus the number of false positives).

$$\text{Precision} = TP / (TP + FP)$$

2. Recall is the fraction of relevant instances that were retrieved. The recall is defined as the proportion of accurately predicted positive observations to all observations in the class.

$$\text{Recall} = TP / (TP + FN)$$

3. F1 Score is the weighted average of Precision and Recall. As a result, both false positives and false negatives are taken into account in this score.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

We used Python for our experimentation. Python is a free, open-source language with highly active community members available across all platforms (Linux, Mac, and Windows). Due to its simple and easy syntax and the presence of a plethora of libraries in Python, it is highly recommended for coding for machine learning and artificial intelligence applications.

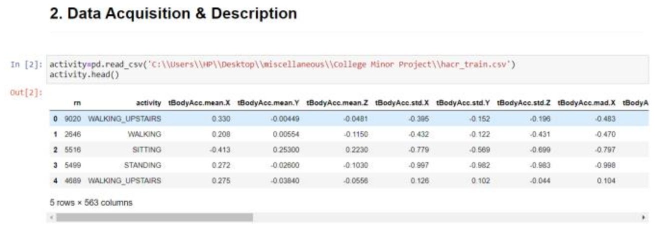
The purpose of this research is to develop a model that can predict the sort of activity or exercise that is conducted based on human movement measurements. Based on a variety of collected data, we applied artificial intelligence techniques to develop a model to predict the style of the exercise, or "activity." The Human Activity Recognition dataset from UCI is used to test artificial intelligence techniques. Dense Layer was the best model as it returned a training accuracy of 89.18 and a testing accuracy of 92.42.

4.1 Dataset:

The data used in this analysis is the Human Activity Recognition Dataset provided by UCI [6]. The train dataset consists of 563 columns and 2887 rows. The test dataset consists of 562 columns and 722 rows. The variables in the dataset describe subjects and their physical movement during activities. The target variable which needs to be predicted is activity column. It consists of 6 activities that are, standing, walking, laying, sitting, walking upstairs and walking downstairs.

4.2 Data Loading:

Firstly, we loaded the data into memory using the following



2. Data Acquisition & Description

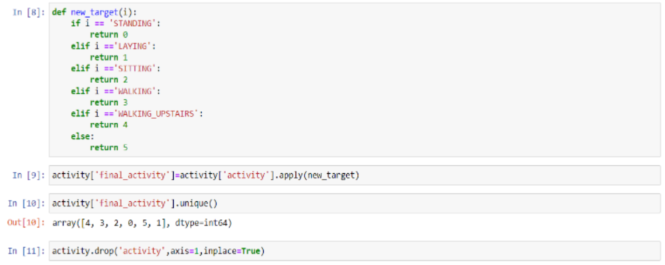
```
In [2]: activity=pd.read_csv('C:\\Users\\HP\\Desktop\\miscellaneous\\College Minor Project\\hacr_train.csv')
        activity.head()
Out[2]:
```

	rn	activity	tBodyAcc.mean.X	tBodyAcc.mean.Y	tBodyAcc.mean.Z	tBodyAcc.std.X	tBodyAcc.std.Y	tBodyAcc.std.Z	tBodyAcc.mad.X	tBodyA
0	9020	WALKING_UPSTAIRS	0.330	-0.00449	-0.0481	-0.395	-0.152	-0.196	-0.483	
1	2646	WALKING	0.208	0.00554	-0.1150	-0.432	-0.122	-0.431	-0.470	
2	5516	SITTING	-0.413	0.25300	0.2230	-0.779	-0.569	-0.699	-0.797	
3	5499	STANDING	0.272	-0.02600	-0.1030	-0.997	-0.982	-0.983	-0.998	
4	4689	WALKING_UPSTAIRS	0.275	-0.03840	-0.0556	0.126	0.102	-0.044	0.104	

5 rows × 563 columns

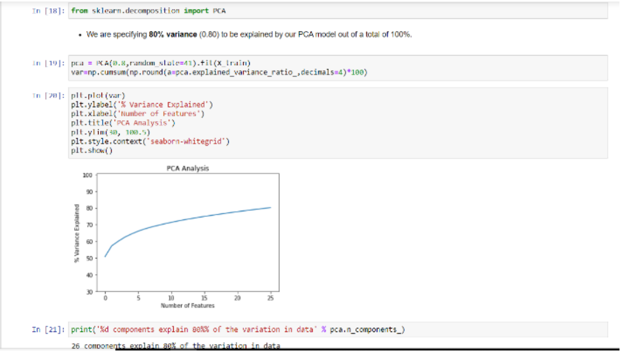
4.3 Data Processing:

Secondly, since the target variable is categorical, we converted it to numerical variable because computers understand numbers better than characters.



```
In [8]: def new_target(i):
            if i == 'STANDING':
                return 0
            elif i =='LAYING':
                return 1
            elif i =='SITTING':
                return 2
            elif i =='WALKING':
                return 3
            elif i =='WALKING_UPSTAIRS':
                return 4
            else:
                return 5

In [9]: activity['final_activity']=activity['activity'].apply(new_target)

In [10]: activity['final_activity'].unique()

Out[10]: array([4, 3, 2, 0, 5, 1], dtype=int64)

In [11]: activity.drop('activity',axis=1,inplace=True)
```

Since there are 562 explanatory variables, we need to apply dimensionality reduction technique to reduce the number of variables. Keeping variability at 80% Principal Component Analysis (PCA) returned 26 columns, this means that 26 columns are able to explain 80% variance in the target variable.



```
In [18]: from sklearn.decomposition import PCA
```

- We are specifying **80% variance** (0.80) to be explained by our PCA model out of a total of 100%.

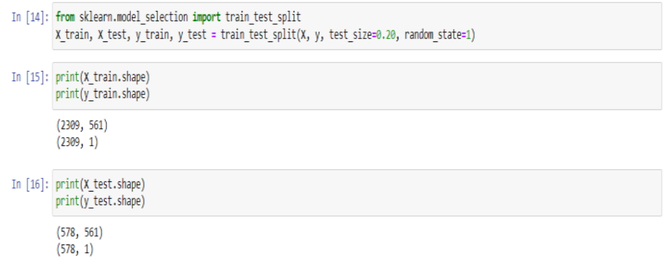
```
In [19]: pca = PCA(0.8,random_state=41).fit(X_train)
         var=np.cumsum(np.round(a=pca.explained_variance_ratio_,decimals=4)*100)

In [20]: plt.plot(var)
         plt.ylabel('% Variance Explained')
         plt.xlabel('Number of Features')
         plt.title('PCA Analysis')
         plt.ylim(30, 100.5)
         plt.style.context('seaborn-whitegrid')
         plt.show()
```

```
In [21]: print('%d components explain 80%% of the variation in data' % pca.n_components_)

         26 components explain 80% of the variation in data
```

4.4 Data Analysis:

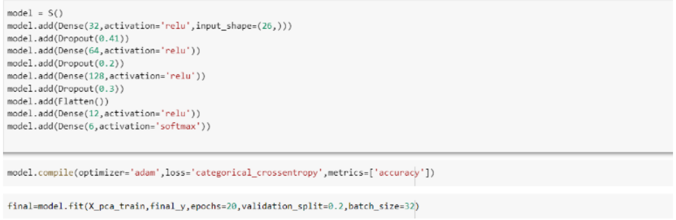
Now, started to analyze our data,

Step 1: Split the data into training testing data. Notice that the ratio of training data to testing data is 4:1.



```
In [14]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=1)

In [15]: print(X_train.shape)
         print(y_train.shape)

         (2309, 561)
         (2309, 1)

In [16]: print(X_test.shape)
         print(y_test.shape)

         (578, 561)
         (578, 1)
```

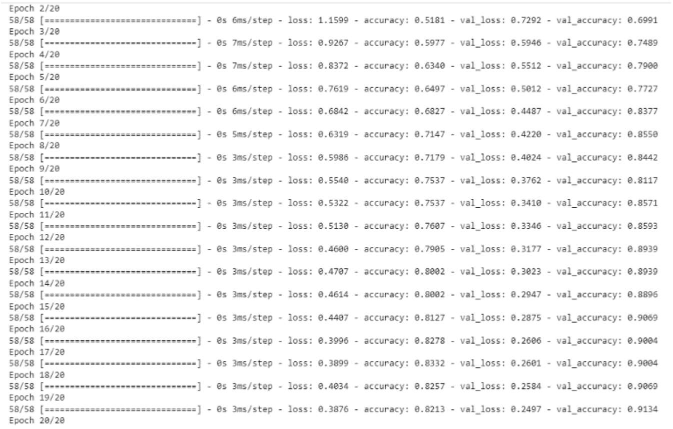
Step 2: Train models with the training data using different AI algorithms. Dense Layer returned best results.



```
model = S()
model.add(Dense(32,activation='relu',input_shape=(26,)))
model.add(Dropout(0.41))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(12,activation='relu'))
model.add(Dense(6,activation='softmax'))

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

final=model.fit(X_pca_train,final_y,epochs=20,validation_split=0.2,batch_size=32)
```

4.5 Evaluation:

Step 3: Now the final step is evaluating the machine learning model. Dense Layer was the best model as it returned training accuracy of 89.18% and testing accuracy of 92.34%. This shows that neither the model is overfitted underfitted.



```
Epoch 2/20
58/58 [==============================] - 0s 6ms/step - loss: 1.1599 - accuracy: 0.5181 - val_loss: 0.7292 - val_accuracy: 0.6991
Epoch 3/20
58/58 [==============================] - 0s 7ms/step - loss: 0.9267 - accuracy: 0.5977 - val_loss: 0.5946 - val_accuracy: 0.7489
Epoch 4/20
58/58 [==============================] - 0s 7ms/step - loss: 0.8372 - accuracy: 0.6340 - val_loss: 0.5512 - val_accuracy: 0.7900
Epoch 5/20
58/58 [==============================] - 0s 6ms/step - loss: 0.7619 - accuracy: 0.6497 - val_loss: 0.5012 - val_accuracy: 0.7727
Epoch 6/20
58/58 [==============================] - 0s 6ms/step - loss: 0.6842 - accuracy: 0.6827 - val_loss: 0.4487 - val_accuracy: 0.8377
Epoch 7/20
58/58 [==============================] - 0s 5ms/step - loss: 0.6319 - accuracy: 0.7147 - val_loss: 0.4220 - val_accuracy: 0.8550
Epoch 8/20
58/58 [==============================] - 0s 3ms/step - loss: 0.5986 - accuracy: 0.7179 - val_loss: 0.4024 - val_accuracy: 0.8442
Epoch 9/20
58/58 [==============================] - 0s 3ms/step - loss: 0.5540 - accuracy: 0.7537 - val_loss: 0.3762 - val_accuracy: 0.8117
Epoch 10/20
58/58 [==============================] - 0s 3ms/step - loss: 0.5322 - accuracy: 0.7537 - val_loss: 0.3410 - val_accuracy: 0.8571
Epoch 11/20
58/58 [==============================] - 0s 3ms/step - loss: 0.5130 - accuracy: 0.7607 - val_loss: 0.3346 - val_accuracy: 0.8593
Epoch 12/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4600 - accuracy: 0.7905 - val_loss: 0.3177 - val_accuracy: 0.8939
Epoch 13/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4707 - accuracy: 0.8002 - val_loss: 0.3023 - val_accuracy: 0.8939
Epoch 14/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4614 - accuracy: 0.8002 - val_loss: 0.2947 - val_accuracy: 0.8896
Epoch 15/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4407 - accuracy: 0.8127 - val_loss: 0.2875 - val_accuracy: 0.9069
Epoch 16/20
58/58 [==============================] - 0s 3ms/step - loss: 0.3996 - accuracy: 0.8278 - val_loss: 0.2606 - val_accuracy: 0.9004
Epoch 17/20
58/58 [==============================] - 0s 3ms/step - loss: 0.3899 - accuracy: 0.8332 - val_loss: 0.2601 - val_accuracy: 0.9004
Epoch 18/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4034 - accuracy: 0.8257 - val_loss: 0.2584 - val_accuracy: 0.9069
Epoch 19/20
58/58 [==============================] - 0s 3ms/step - loss: 0.3876 - accuracy: 0.8213 - val_loss: 0.2497 - val_accuracy: 0.9134
Epoch 20/20
```

5. CONCLUSION AND FUTURE SCOPE

The research looked at human activity identification approaches and described the overall data gathering methodology for HAR as well as the Python-based artificial intelligence -based data analysis process. Among the several artificial intelligence algorithms used, the dense layer technique outperformed others. Because the difference between train and test accuracy was so minimal, this was the best model. This indicates that the model was neither overfitted nor underfitted. The storage of real-time sensor data analysis would be the next stage on this approach.

ACKNOWLEDGEMENT

First and foremost, we'd like to express our sincere appreciation to our mentors Mr. Ajay Kaushik, for his unwavering support throughout the project and research paper, as well as his patience, encouragement, excitement, and vast knowledge. Their advice was invaluable during the research and drafting of this report. For our project, we could not have asked for a greater counsellor and mentor.

REFERENCES

[1] 1. T. van Kasteren, G. Englebienne, and B. Krse, "An activity monitoring system for elderly care using generative and discriminative models", Journal on Personal and Ubiquitous Computing, 2010.

[2] D. Choujaa and N. Dulay, "Tracme: Temporal activity recognition using mobile phone data", in IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 1, pp. 119-126, 2008.

[3] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors", IEEE Transactions on Information Technology in Biomedicine, vol. 10, no. 1, pp. 119-128, 2006.

[4] O. D. Lara and M. A. Labrador, "A mobile platform for real time human activity recognition", in IEEE Conference on Consumer Communications and Networks, 2012.

[5] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly supervised recognition of daily life activities with wearable sensors", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2521-2537, 2011.

[6] https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions