Human Activity Recognition based on Deep Learning

1st Nusrat Khan
170204101

2nd Fuad Chowdhury
180104137

3rd Abdur Samad
180104139

4th Shamse Nur Shanto
180104128

***Abstract*—Human activity recognition (HAR) is the practice of using Artificial Intelligence (AI) to recognize and name human actions from raw activity data collected from a variety of sources or devices. These devices include wearable sensors, electronic device sensors like inertial sensors for smartphones or smartwatches, camera devices, closed-circuit television, etc. For its many different application sectors, like healthcare, surveillance, remote care for old people or children living alone, smart homes and offices, and numerous monitoring applications like sports and exercise, HAR is incredibly helpful and significant. The safety and quality of human life are improved by the widespread use of HAR.**

In light of the widespread use and accessibility of sensors, such as accelerometers and gyroscopes, in products like smartphones and smartwatches, human activity identification has recently attracted significant attention in both industrial and academic research. Even though the number of electronic gadgets and their uses is constantly increasing, artificial intelligence (AI) advancements have transformed our ability to extract deep buried information for precise detection and interpretation. To utilize this scope, we have gathered data using smartphone accelerometer and gyroscope sensors in our proposed study. This data is used to train the machine and recognize the correct human action. We have been able to predict 18 different activities using deep learning techniques, with the Artificial Neural Network approach having the highest accuracy (86.98%).

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Over the past few decades, smartphones have evolved into a significant and indispensable aspect of modern life. Many low- and middle-income nations have embedded smartphones into their societal structures. Statistica[1] estimates that there are currently 6.648 billion smartphone users worldwide, which equals 83.37% of the world's population. The quantified self movement, which is a more recent development, has led to a large acceptance of smartwatches as a replacement for traditional watches. Thus, working with the sensor data from smartphones is still quite relevant. Smartphones are equipped with a variety of sensors, including GPS, gyroscopes, and accelerometers. These sensors assist in recording daily activities like walking, running, sitting, etc., and are a major factor in the decision of many people to own a smartphone.

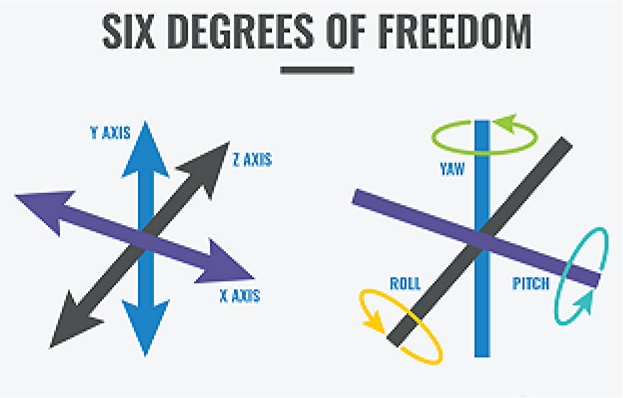


Fig. 1. Six degrees of freedom

Android devices feature integrated sensors to determine orientation (magnetometers), movement (accelerometers, gravity sensors, and gyrator), and various ecological situations (indicators, photometers, and thermometers). The main function of an accelerometer is to detect changes in how smart devices are oriented in relation to data and adjust the orientation to the user's survey angle. For instance, by switching the phone's mode to flat, you can see this scene when looking for a website page with enlarged width. When the direction of the device or camera is changed, the camera mode additionally switches from landscape to portrait or vice versa. Eventually, this sensor senses the adjustment in orientation by a 3D (X, Y, and Z pivot) measurement of the acceleration of the gadget concerning free fall. A gyroscope is used to maintain and control position, level, or orientation that depends on the angular momentum standard. The "Gyros" that is used in conjunction with the accelerometer senses movement on the six axes, or right, left, up, down, forward, and backward. Additionally, it recognizes the roll, pitch, and yaw motions, which are angular moments observed from the three axes, X, Y, and Z. Gyroscopic sensors, which use MEMS (Micro Electrical and Mechanical System) technology, aid in navigation and can recognize the gesture frameworks used by smartphones and tablets.

In a typical day, we perform tens of thousands of tasks, the majority of which require physical movement. The classification of human actions into preset categories such as stand, sit, talk-sit, talk-stand, stand-sit, lay, lay-stand, pick, jump, push-up, sit-up, walk, walk-backwards, walk-circle, run, stair-up, stair-down, table tennis, etc. is known as human activity recognition. Oftentimes, we use this sensor data while working out, exercising, jogging, or walking in order to estimate how long we are participating in this activity. What if, however, we could predict other commonplace actions like talking, lying, climbing stairs, or playing table tennis using similar data?An organized classification of this data into a designated class can benefit users and make it simpler for people to obtain the information they require. Simply acknowledging human activity using the accelerometer and gyroscope's X, Y, and Z axis values is the goal of this research.

We have read a lot of literature on sensor-based human activity recognition. In these works, they have attempted to classify human actions using the analysis of gyroscope and accelerometer 3-axial data. With a different dataset and more classifications, we perform the same classification in this paper. The opportunity to deal with sensor data even more is greatly enhanced by earlier study on this subject. As these actions are frequent in our daily lives, we assumed that by applying the knowledge from the papers we read, we could further classify sensor data. That's why we gathered some datasets on different human activity and merged them in one dataset, then labeled the dataset into 18 categories to train the machine.

A dataset was created by combining over 218,669 samples of data. The dataset has undergone preprocessing through feature value scaling. We used a variety of deep learning techniques to train the computer to classify data after preprocessing our dataset. There have been several techniques used, including Artificial Neural Network, Convolutional Neural Network and Long Short-term Memory. Then, we have compared their accuracy, precision, recall and F1-score.

In the context of ordinary people, human activity recognition will play an important role. Identification of human behaviors from sensor data, signals, photos, or videos has been a critical concern with the rapid advancement of technology and artificial intelligence, and machine learning models to address this issue have largely been successful. And it can be really helpful if we can anticipate routine actions using our smartphones. Working parents who must leave their kids or aging parents behind can keep an eye on their activities for no extra charge. Additionally, it can aid in preventing a variety of stair-related injuries. Hardware embedded systems can make advantage of this forecasting methodology. In addition to sports, industries, and medical research can all greatly benefit from this study. It can also be utilized for societal good, such as creating a smart city and employing surveillance to foresee unforeseen events.

Here comes the objective of this study, which is a multi-level classification of human activity recognition from a smartphone sensor (Accelerometer and Gyroscope) dataset based on their X, Y, Z axis acceleration and rotation. In order to achieve that, we manually labeled our dataset into eighteen different categories and then we utilized a variety of machine learning algorithms to forecast their class and assess their accuracy.

The main contributions of this research are :

- This research might assist working parents monitor their kids' whereabouts and support their elderly parents in avoiding strenuous work. Moreover, this has applications in medical science for healthcare therapies.
- The main contribution is testing human activity recognition techniques on a fresh dataset that has been augmented with new classes and can predict more actions.
- Additionally, by incorporating this prediction into hardware, we can create more intelligent devices and prevent a variety of unusual events from occurring. As a result, it will create a wealth of possibilities for further study in this area.

In the following chapter, we will discuss the literature evaluation of the papers that are pertinent to our study in order to further this thesis objective. After determining the scope of these papers, we next offered our work proposal and provided an explanation of working procedure along with the results we were able to attain. Finally, we have provided a conclusion for this thesis as well as suggestions for next research.

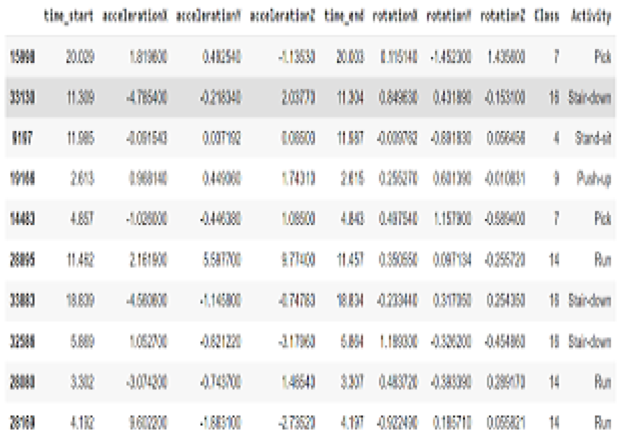
II. Related Works

The intrinsic characteristics of activities and 1D time-series signals were used by C.A. Ronao and S.B Cho for their research paper [**?**] to perform efficient and effective Human Activity Recognition utilizing smartphone sensors. At the same time, a method for automatically and data-adaptively extracting reliable features from raw data was provided. Despite the fact that the difference in feature complexity level reduces with each subsequent layer, experiments demonstrate that convnets do in fact extract relevant and more complex features with each additional layer. It is possible to use a larger time range of temporal local correlation (1–9–14), and it has been demonstrated that a small pooling size (1–2-13) is advantageous. Additionally, Convnets classified moving activities very perfectly in this research, particularly those that were identical and were previously thought to be exceedingly challenging to categorize. For the benchmark dataset produced from 30 volunteer participants, Convnets beat other cutting-edge data mining techniques in HAR, obtaining an overall performance of 94.79% on the test set with raw sensor data and 95.75% with the addition of temporal fast Fourier transform information. Bolu Walade and Sunil Neela [**?**] made use of the well-known WISDM dataset for activity recognition. They discovered a statistically significant difference (p 0.05) between the data produced from the sensors incorporated in smartphones and smartwatches using multivariate analysis of covariance (MANCOVA). By doing this, they demonstrated that due to their different wearable locations, cellphones and smartwatches do not acquire data in the same manner. For the purpose of classifying 15 different hand- and non-hand focused activities, they used numerous neural network designs. Among these were the Long Short-Term Memory (LSTM), Bi-Directional Long Short-Term Memory (BiLSTM), Convolutional Neural Network (CNN), and Convolutional LSTM models (ConvLSTM). With accelerometer data from watches, the generated models performed best. Additionally, they observed that in 12 out of the 15 activities, the end-to-end LSTM classifier's classification precision was worse than that obtained with the convolutional input classifiers (CNN and ConvLSTM). In addition, compared to hand-related activities, the CNN model for the watch accelerometer was better able to categorize non-hand oriented activities. In other studies, one research has been done with Deep Learning approaches and another is done with machine learning approaches. They both have used the same UCI machine learning repository dataset made of utilizing smart phones. From both papers, we can clearly see that deep learning worked better than machine learning models on the same dataset. Depanwita Biswas and Suparna Thakur [?] provided a DL-based method for activity identification using accelerometer and gyroscope data from smartphones. Long short-term memory (LSTM), autoencoders, and convolutional neural networks (CNNs) all have complementing modeling abilities since LSTMs were skilled at temporal modeling, AEs were employed for dimensionality reduction, and CNNs excel at automatic feature extraction. In this study, they combined CNNs, AEs, and LSTMs into a single architecture to benefit from their complementary nature. They examined the proposed architecture, known as the "ConvAE-LSTM," using four distinct widely available standard datasets (WISDM, UCI, PAMAP2, and OPPORTUNITY). The experimental results show that their unique technique is workable and offers comparable smartphone-based HAR solution performance enhancements over existing state-of-the-art methods in terms of computational time, accuracy, F1-score, precision, and recall. Whereas Sandeep Kumar Polu [?] looked at how Random Forest (RF) and Modified Random Forest (MRF) sort calculations are executed in an online Activity Recognition framework operating on Android frameworks. This technique can support online training and class the most effective use of the accelerometer data. The Random Forest classification technique is typically used first, followed by an enhancement called MRF, which is a Modified Random Forest. Modified Random Forest will eliminate the computational complexity of the Random Forest for the purpose of Activity Recognition by building decision trees (creating littler preparing units for each activity and class may be done dependent on those diminished preparing sets).From a series of observations on human motions including sitting, walking, running, resting, and standing in an online activity identification device, they have anticipated that these classifiers will have operated generally. In this research, they proposed to compare the general performance of classifiers with limited preparation records and limited open memory on smart devices to offline execution. And they got the best accuracy with Modified Random Forest which is 93%.

III. Dataset Description

Human Activity Recognition (HAR) refers to the capacity of machines to perceive human actions. This dataset[2] is collected from Kaggle which contains information on 18 different activities collected from 90 participants (75 male and 15 female) using smartphone sensors (Accelerometer and Gyroscope). It has 1945 raw activity samples collected directly from the participants, and 9185 subsamples extracted from them. Samples were collected at 100 Hz, gravity acceleration was omitted from the Accelerometer data, and no filter was applied to remove noise. The activities are:
Stand=Standing still (1 min)
Sit= Sitting still (1 min)
Talksit= Talking with hand movements while sitting (1 min)
Talk-stand= Talking with hand movements while standing or walking(1 min)
Stand-sit= Repeatedly standing up and sitting down (5 times)
Lay= Laying still (1 min)
Lay-stand= Repeatedly standing up and laying down (5 times)
Pick= Picking up an object from the floor (10 times)
Jump= Jumping repeatedly (10 times)
Push-up= Performing full push-ups (5 times)
Sit-up= Performing sit-ups (5 times)
Walk= Walking 20 meters (12 s)
Walkbackward= Walking backward for 20 meters (20 s)
Walkcircle= Walking along a circular path (20 s)
Run= Running 20 meters (7 s)
Stairup= Ascending on a set of stairs (1 min)
Stairdown= Descending from a set of stairs (50 s)
Tabletennis= Playing table tennis (1 min)
This dataset includes four zip files among which we are working with "Trimmed Raw Data. zip". This zip file contains samples of "Raw Time Domain Data" files after certain parts of the signals that contained no information on the corresponding activity were trimmed. After extracting this zip file, we found 18 folders of 18 different activities from 0 to 17 which contain minimum 20 to maximum 334 csv files each and in every csv file there are around 7000 to 1000 samples which are noise free. As we are working on a multi level classification, constructing a balanced dataset is a must. For this reason, we have collected the first 2 to 8 csv file's data from each folder or each class and merged them all to build a dataset of 218,669 samples of 18 classes. Here, we got minimum 10,179 to maximum 13,967 data in each class. Dataset instance is given below:

This dataset has 10 features among which 8 features were given and the last two features (Class and Activity) are combined by us. The description of the features are : Time_start, Time_end : exact time (elapsed since the start and end) when the Accelerometer Gyro output was recorded (in ms) Acceleration_X, Acceleration_Y, Acceleration_Z : Acc.meter X, Y, Z axes readings (in m/s$\hat{2}$) Rotation_X, Rotation_Y, Rotation_Z : Gyro X, Y, Z axes readings or Rate of rotation around X,Y,Z axes (in rad/s) Class : Encoded value of each class Activity : Corresponding activity Out of these features,



	time_start	accelerationX	accelerationY	accelerationZ	time_end	rotationX	rotationY	rotationZ	Class	Activity
15998	20.029	1.819600	0.492540	-1.13533	20.033	0.115140	-1.452300	1.435800	7	Pick
33138	11.309	-4.785400	-0.218340	2.03773	11.304	0.849630	0.431890	-0.153100	16	Stair-down
9167	11.985	-0.061643	0.037192	0.08600	11.987	-0.009762	-0.891930	0.056456	4	Stand-sit
19166	2.613	0.968140	0.449060	1.74313	2.615	0.256270	0.601390	-0.010831	9	Push-up
14483	4.857	-1.028600	-0.446380	1.08600	4.843	0.497540	1.157900	-0.589400	7	Pick
28895	11.462	2.161900	5.587700	9.71400	11.457	0.350650	0.097134	-0.255720	14	Run
33883	18.839	-4.560600	-1.146800	-0.74763	18.834	-0.233440	0.317050	0.254350	16	Stair-down
32588	5.889	1.052700	-0.621220	-3.11960	5.884	1.189300	-0.326200	-0.454860	16	Stair-down
28060	3.302	-3.074200	-0.743700	1.46543	3.307	0.483720	-0.383390	0.289173	14	Run
28169	4.192	9.602200	-1.883100	-2.73529	4.197	-0.922490	0.185710	0.055821	14	Run

Fig. 2. Dataset Sample

class or activity is our target column that we need to predict.

IV. Formulation

The application of our strategy is covered in this part, along with examples of the outcomes.

A. Pre-Processing

Among the 8 features, we have splitted our dataset into a train and test set in a stratified manner so that our model can learn an equal number of samples from each class. The dataset got splitted into 20% test and 80% train. As the values of features are distributed in different ranges, we have used a feature scaling model "Standardization". Standardization is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. StandardScaler() function is used to do this. After completing the preprocessing phase, we have inputted our training data into deep learning algorithms and fit them by learning so that we can use these models on our test dataset to predict activities.

B. Work Flow Diagram

C. Model Training

- **Artificial Neural Network:**
  It is a method for processing data and producing output that mimics the neural network in order to reveal non-linear relationships in a sizable dataset. The information may come through sensory channels and take the shape of text, images, or sounds. The easiest method to comprehend how a natural neural network in the brain functions is to understand how an artificial neural network functions and make comparisons between them. The basic building block of the human brain is called a neuron, and it is this neuron that is responsible for learning and memory as we currently understand it. You can think of them as the brain's processing center. They receive input from the sensory system, process it, and then output information that is used by other neurons.
  The scaled dataset was input into the ANN model containing dense layer (32 units), followed by dense layer (200 units), dense layer (800 units), dense layer (1024 units), dense layer (512 units), dense layer (256 units), dense layer (128 units), dense layer (64 units), dense layer (36 units), dense layer (32 units) and a last dense layer with 18 units (for the number of classes). We used Softmax as the activation function in the last layer, ReLU for the previous layers, and the Adam optimizer. The loss was calculated in categorical cross-entropy. The model was trained for 50 epochs with a batch size of 500.
- **Convolutional Neural Network:**
  In essence, convolutions are feature extractors that pick up the key details of an input. When classifying images, CNN's lower layers focus on details like edges or colors while the upper layers create more abstract representations of the data. The representations that the convolutional layers learned are then subjected to a logistic regression. Deep learning models work on word embeddings, which are continuous representations of the input, for the majority of NLP applications. Word embeddings will be subjected to n-gram convolutions with kernel size n that will learn to emphasize or ignore the input's n-grams. For instance, a kernel of size 2 can learn which bigrams (2-gram) are crucial for the final classification choice in sentence classification.The attention methods employed frequently in encoder-decoder designs are analogous to this idea. The data is reshaped and inputted to a 1-dimensional convolution layer with 64 filters of kernel size 2 followed by a 1-dimensional max-pooling layer with 0.2 pool size, flatten layer, dense layer with 32 units, dense layer with 200 units, dense layer with 800 units, dense layer with 1024 units, dense layer with 512 units, dense layer with 256 units, dense layer with 128 units, dense layer with 64 units, dense layer with 36 units and finally a Dense layer with 18 units for classification. We used Softmax as the activation function in the last layer, ReLU for the previous layers, and Adam optimizer. The loss was calculated in categorical cross-entropy. The model was trained for 40 epochs with a batch size of 1000.
- **Long short-term Memory (LSTM):**
  Long short-term memory blocks are used by the recurrent neural network to offer context for how the program receives inputs and produces outputs. A complicated structure, the long short-term memory block has several parts, including weighted inputs, activation functions, inputs from earlier blocks, and ultimate outputs. Because the program uses a structure built on short-term memory processes to build longer-term memory, the unit is known as a long short-term memory block. Examples of applications for these systems include natural language processing. In order to evaluate a word or phoneme in relation to other words in a string—where memory might

be helpful in sorting and categorizing these types of inputs—the recurrent neural network uses lengthy short-term memory blocks.

The dataset was input into the LSTM model containing 32 LSTM units, followed by a dropout layer (0.2 units), dense layer (200 Units), dropout layer (0.2 units), dense layer (800 units), dropout layer (0.2 units), dense layer (1024 units), dropout layer (0.4 units), dense layer (512 Units), dropout layer (0.2 units), dense layer (256 units), dropout layer (0.2 units), dense layer (128 units), dropout layer (0.2 units), dense layer (64 Units), dropout layer (0.2 units), dense layer (36 units), dropout layer (0.2 units), and a last dense layer with 18 units (for the number of classes). We used Softmax as the activation function in the last layer, ReLU for the previous layers, and the Adam optimizer. The loss was calculated in categorical cross-entropy. The model was trained for 30 epochs with a batch size of 1000.

D. *Classification Result Evaluation*

To classify the activities, we utilized three classifiers, namely Long short-term memory (LSTM), Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). The Precision, Recall, Accuracy and F1 scores were used as evaluation metrics to analyze the performance of the classifiers. Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Accuracy = (TP+TN)/(TP+FP+FN+TN)

Precision, or the caliber of a successful prediction made by the model, is one measure of the model's performance. The proportion of accurately categorized positive samples (True Positive) to the total number of positively classified samples is known as precision (either correctly or incorrectly). Precision = (True Positive)/(True Positive + False Positive)

The recall is determined as the proportion of Positive samples that were correctly identified as Positive to all Positive samples. The recall gauges how well the model can identify positive samples. The more positive samples that are identified, the larger the recall. Recall = (True Positive)/(True Positive + False Negative)

The weighted average of Precision and Recall is the F1 Score. Therefore, both false positives and false negatives are considered while calculating this score. Although F1 is generally more beneficial than accuracy, especially if you have an uneven class distribution, it is not intuitively as simple to understand as accuracy. When false positives and false negatives cost about the same, accuracy performs best. It is preferable to include both Precision and Recall if the costs of false positives and false negatives are significantly different. F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Models	Accuracy	Precision	Recall	F1 score
ANN	0.82	0.82	0.82	0.82
CNN	0.70	0.73	0.70	0.71
LSTM	0.50	0.53	0.52	0.52

TABLE I
EXPERIMENTAL RESULT TABLE

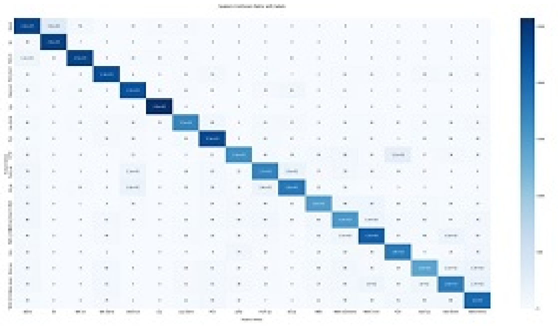


Fig. 3. Confusion Matrix

From the above table, we can see our best performing model is ANN which has achieved 97.22% training accuracy and 82.4% test accuracy. And the second best performing model is CNN which has achieved 87.96% training accuracy and 70.6% test accuracy. The worst performance was shown by LSTM. But when experimentally we tried to do this classification with binary classes, the LSTM model gave 95% accuracy. Therefore, we can say, the LSTM model couldn't handle multiclass classification with good precision.

Comparing ANN and CNN (ignoring LSTM as it performed very poorly), ANN model can predict 82% classes and CNN model can predict 71% classes accurately on average. The best and worst performing class by both ANN and CNN is Lay and Walk-backward by 99% and 70% F1-score. By this table, we analyze that these models can predict static activities more precisely than moving activities and also achieve more than

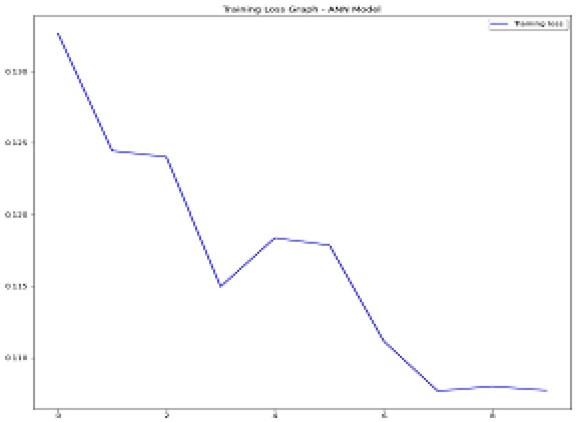


Fig. 4. Loss Function Graph of ANN

class	ANN	CNN	Mean
Stand	0.83	0.54	0.68
Sit	0.92	0.88	0.9
Talk sit	0.94	0.88	0.90
Talk stand	0.89	0.81	0.85
stand sit	0.87	0.77	0.82
Lay	0.99	0.97	0.98
Lay stand	0.82	0.70	0.76
Pick	0.85	0.74	0.79
Jump	0.78	0.72	0.75
Push Up	0.75	0.63	0.69
sit up	0.81	0.70	0.75
Walk	0.77	0.66	0.71
Walk backward	0.70	0.53	0.61
Walk circle	0.79	0.66	0.72
run	0.82	0.74	0.78
stair up	0.72	0.56	0.64
stair down	0.75	0.66	0.70
Table tennis	0.78	0.68	0.73
mean	0.82	0.71	0.76

TABLE II
F1 SCORE FOR EVERY CLASS

70% accuracy in hand-oriented activities.

V. CONCLUSION AND FUTURE WORK

Due to its extensive applicability in a variety of disciplines, including healthcare and assisted living, sensor-based human activity detection is a significant and well-known research subject that emerged from several fields of ubiquitous computing. In this paper, we attempt to create a reliable system for identifying human activity using data from gyroscopes. We have used several machine learning algorithms to detect the classes. Among all the used machine learning algorithms, we have got Random Forest as the highest accuracy to identify human activity.We have got 78.72% accuracy with Random Forest. We also used a balanced dataset and for that we have got a pretty much higher accuracy. We tried to classify 18 different Human movements in this paper with a gyroscope dataset. Among the classes, we have 2 classes named stair-up and stair-down. Fall or slip from stairs is a common case for any age of human. So in future, we can predict by X,Y and Z axis's whether any person has unusual movement on the stairs. By using this, we can make a digital stair that will prevent any unexpected accident from stairs. Physical ability of an athlete can also be detected by human movement. In this way, it can create an impact on our society.

REFERENCES

[1] Charissa Ann Ronao, Sung-Bae Cho, "Human activity recognition with smartphone sensors using deep learning neural networks " April 25,2016.

[2] Bolu Oluwalade1 , Sunil Neela1 , Judy Wawira2 , Tobiloba Adejumo3 and Saptarshi Purkayastha, Human Activity Recognition using Deep Learning Models on Smartphones and Smartwatches Sensor Data.

[3] Dipanwita Thakur and Suparna Biswas, "ConvAE-LSTM: Convolutional Autoencoder Long Short-Term Memory Network for Smartphone-Based Human Activity Recognitio" Digital object Identifier, December 23,2021, pp. 271–350.

[4] Sandeep Kumar Polu, "Human Activity Recognition on Smartphones using Machine Learning Algorithms," IJIRST November 2018.

[5] ¡https://www.kaggle.com/datasets/arashnic/har-1¿ ,accessed 12 December 2022

[6] ¡https://www.geeksforgeeks.com/knn¿ ,accessed 08 December 2022

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.