2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)
Mody University of Science and Technology, Lakshmangarh, Feb 21-22, 2020

Human Activity Recognition using Accelerometer and Gyroscope Data from Smartphones

Khimraj
Department of Computer Science and Engineering
Malaviya National Institute of Technology
Jaipur, India
2017ucp1706@mnit.ac.in

Praveen Kumar Shukla
Dept. of Electronics and Telecommunication
National Institute of Technology
Raipur, India
pkshukla.phd2017.elx@nitrr.ac.in

Ankit Vijayvargiya
Department of Electrical Engineering
Malaviya National Institute of Technology
Jaipur, India
ankitvijayvargiya29@gmail.com

Rajesh Kumar
Department of Electrical Engineering
Malaviya National Institute of Technology
Jaipur, India
rkumar.ee@mnit.ac.in

***Abstract*—Human Activity Recognition is a procedure for arranging the activity of an individual utilizing responsive sensors of the smartphone that are influenced by human activity. Its standouts among the most significant building blocks for numerous smartphone applications, for example, medical-related applications, tracking of fitness, context-aware mobile, survey system of human, and so forth. This investigation centers around acknowledgment of human activity utilizing sensors of the smartphone by some machine learning and deep learning characterization approaches. Data received from the accelerometer sensor and gyroscope sensor of the smartphone are grouped to recognize the human activity.**

Keywords—Smartphone, Human Activity Recognition, Accelerometer, Gyroscope, Classification, Machine Learning.

I. INTRODUCTION

Today smartphone is one of the most supportive instruments in our daily life. Furthermore, with the propelling innovation they get a great deal of proficient step by step to fulfil customer wants and desires. To make these devices a lot of powerful and practical, designers installed advanced modules and software to these devices. Sensors have an important role in making the smartphone a great practical and mindful of the circumstances, so most smartphones accompany diverse implanted sensors and this makes it achievable to gather huge measures of information with respect to the client's presence and activities. Accelerometer and gyroscope sensors region unit among these gadgets as well [1].

An accelerometer is an identifier that estimates increasing speed also as tilt, tilt edge, slant, revolution, vibration, and crash. To offer common sense with a smartphone, the accelerometer programming bundle ought to translate the gadget output. Smartphones utilize numerous styles of accelerometer, the detector, and programming bundle speaking to the primary varieties between the accelerometer. When connected to a cell phone, an accelerometer will naturally adjustment the gadget's screen direction vertically or horizontally. A gyroscope allows a smartphone to quantify, and look after the direction. Gyroscopic sensors will monitor and management device positions, orientation, direction, angular motion, and rotation. When connected to a smartphone, a gyroscopic detecting component unremarkably performs signal acknowledgment capacities. Also, gyrators in smartphones encourage to decide the position and direction of the phone.

In this paper, a Dataset acquired from the UCI Machine Learning Repository comprise of the signals received from the accelerometer and gyroscope sensors of smartphone conveyed by various subjects while performing various activities. This dataset is classified utilizing the diverse machine learning methodologies and after that their performances are compared in terms of accuracy.

II. RELATED WORK

In the modern era, Human Activity Recognition (HAR) is an active and challenging research area due to its applications in different areas like healthcare, and security. A large portion of works related to this center around breaking down the execution of grouping calculations using different machine learning algorithms like Support Vector Machines (SVM), Decision Trees (DT), Hidden Markov Chain (HMC), Naive Bayes, Multi-Layer Perceptron (MLP), k-Nearest Neighbors (KNN), and Random Forrest (RF). Jun Yang and others separated direction free features from the three feature sets, having magnitude, horizontal and vertical features. Every list of features comprises mean, interquartile, standard deviation, entropy, spectrum centroid, zero-cross rate, and 75 percentiles. Authors utilized Attributed Determination channels to give seven feature subsets and calculate recognition precision on these subsets. Thus, the precision of the classifiers on every subset is lower than that with all the features, for example, Decision Tree equivalents to 90.4% and Decision Tree with all features is 90.6%, Naive Bayes equivalents to 68.3% and Naive Bayes with all features is 68.7% [2]. Sian Lun Lau and others utilized normal four features mean, correlation, standard deviation, and energy of Fast Fourier Transform. They joined features in 3 gatherings: bunch G1 incorporates with normal and standard deviation of estimations of every axis and all of the three-axis, bunch G2 incorporates with the normal and the standard deviation of the FFT coefficients of each axis and all of the three-axis, bunch G3 incorporates with every one of the four highlights of each axis and each of the three-axis. Be that as it may, they

978-1-7281-1420-0/20/$31.00 ©2020 IEEE

utilized basic features what's more, combined features into gatherings manually [3]. Ville Kononen et. al. has been used 2 feature determination strategies, these are Sequential Forward Selection and Selection to choose features from heart rate signals and the accelerometer to estimate complex classification than with the simple classification Nonetheless, the feature determination method is used to select features and looked at accuracy of the classifier on these features and recognition accuracy of that classifier extend from about 60% to 90% [4].

III. Methodology

A. Dataset

This dataset consists the recording of 30 people having an age range of 19-48 years performing various activities of daily living (ADL) while conveying a waist-mounted smartphone with implanted inertial sensors. These activities are given in Table 1 below with their relating codes. By using embedded accelerometer and gyroscope of smartphone, we collected linear acceleration as well as axial angular velocity along all 3 axes at a constant sampling frequency of 50Hz. The label to the activity in the data is given manually. Then a noise in data is filtered by 20Hz Butterworth filters to achieve more exact outcomes. Another Butterworth filter of 3hz is connected to dispose of the impact of gravity in the accelerometer signals. Data at that point standardized to (-1,1) range. Euclid magnitude of the estimations of 3 measurements determined to combine 3D signal into one dataset [5]. At last, the dataset comprises of 561 features.

TABLE I. ACTIVITIES WITH THEIR CORRESPONDING CODES

Activity	Code
WALKING	1
WALKING UPSTAIRS	2
WALKING DOWNSTAIRS	3
SITTING	4
STANDING	5
LAYING	6

Feature extraction is a process transforms the high dimensional information to a lower-dimensional feature space. The change can be linear or nonlinear. In this paper, we utilized a Linear Discriminant Analysis (LDA).

Linear Discriminant Analysis is a supervised algorithm that takes the label of class into consideration. It may be a way to decrease dimension whereas at the same time preserving as much of the class discrimination data as possible. Fundamentally LDA finds a centroid of each class data point. And after that on the premise of this, it decides a new dimension which is nothing but an axis that should fulfill two criteria. The first is to maximize the separation between the centroid of each class. and second is to minimize the variation inside each category.

In Linear Discriminant Analysis we attempt to project the data set onto a space having lower dimension for great class separability to avoid over-fitting [7]. All the steps of LDA are as follows:

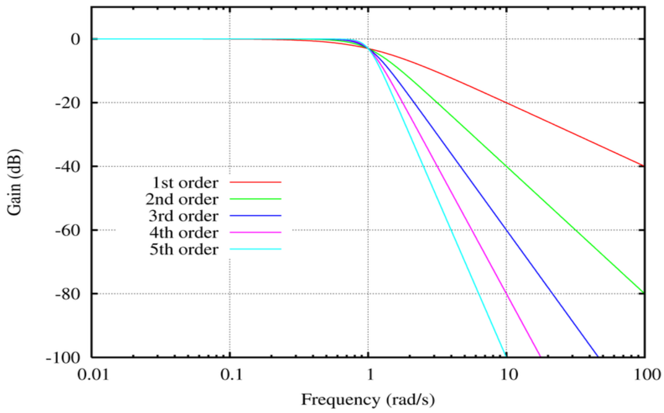

Fig. 1. Low-pass Butterworth filters of orders 1 to 5, with unit cutoff frequency [6].

- Calculate the mean vectors m_i (i=1,2,3,4,5,6) of all the 6 different human activities.
- Calculate Scatter Matrices, Within class scatter matrix (S_W) and Between class scatter matrix (S_B).

$$S_W = \sum_{i=1}^{c} \sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T \tag{1}$$

$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T \tag{2}$$

Where m is the overall mean of all the classes, mi is mean of ith class, Di is ith dimension (column) of data set and Ni is the size of ith class.

- Find the eigenvalue and eigenvector of the matrix SW−1SB.
- Sort the eigenvector by the decreasing eigenvalues.
- Select the top k eigenvectors and form matrix W of k×d dimension.
- Transform the data set on to the new subspace.

$$Y = X \times W \tag{3}$$

where X is a matrix of n×d dimension contains original data set having n samples, and Y is matrix contains the transformed n×k dimensional samples within the new subspace.

B. Classification Methods

We have used some of machine learning and deep learning algorithms to classify different activity from the dataset. These machine learning and deep learning algorithms used different approaches for classification. Planned models initially trained with a 70% of the all dataset, afterward tested for prediction with 30% of the remaining dataset.

Cross-validation is used to evaluate the accuracy of machine learning algorithms on a constrained data test. This approach includes randomly separating the set of data into k partitions, or folds, of roughly equal size. Then a partition is treated as a validation set, and the strategy is fit on the remaining k−1 partitions to the model. In this paper classification accuracy of used models is tested by 5-fold cross-validation [8].

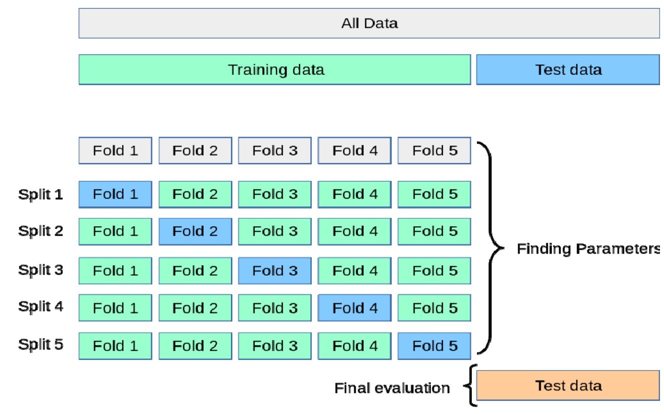

Fig. 2. Block diagram of 5-fold cross validation [9]

Different methods tried in this paper for the classification of human activity are given below:

- K-Nearest Neighbors (k-NN)
- Support Vector Machines (SVM)
- Convolutional Neural Network (CNN)

1) K-Nearest Neighbors (k-NN): K-Nearest Neighbors is a machine learning model and this algorithm shows the characteristics of instance-based learning, as well as lazy learning, in which the function is just calculated locally and all the calculation is conceded until grouping. The k-NN classification is one of the easiest models among all the other machine learning models. It is mostly used as a method of classification, in which grouping of examples is depending on their coordinates and distance from others on the feature space.

Within the case of k-NN, it does not compare the unclassified data with all other, instead it performs a mathematical calculation to measure the distance between the data to makes its classification [10]. All the steps of k-NN are as follows:

- Get the unclassified data.
- Calculate the distance from the new data to all classified data.
- Initialize the value of k.
- Sort the distances.
- Take k nearest neighbors.
- Assign the class to new data by the majority vote of its k nearest neighbors.

To measure the distance between two points there are many different ways like Euclidian, Manhattan, Minkowski or Weighted, etc. In this paper, we have used the Euclidean distance. The Euclidean distance between two points p and q are given by below formula:

$$d(p,q)=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+...+(p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q\ )^2} \quad (4)$$

In the k-NN classification, the output for an item is a class from which this belongs. The class of an item is assigned by the majority vote of its k nearest neighbors and the item is assigned to the class which is the most basic between k closest neighbors (generally k is a small positive number) of that item. When k = 1, the item is generally assigned to a class of that single closest neighbor. Cho222osing the appropriate value of k is critical. A lower value of k 2may be influenced by noise. While a higher value of k may be a reason for including different member classes to the base group. Since our dataset noise is filtered 2 times, there is a lesser risk for choosing the lower value of k. Accuracy for k=1 is 87.85%, while when the value of k is 5 accuracy is 90.02%. But if the same model is used after dimension reduction accuracy for k=1 is 97.72% and for k=5 accuracy achieved is 97.99%.

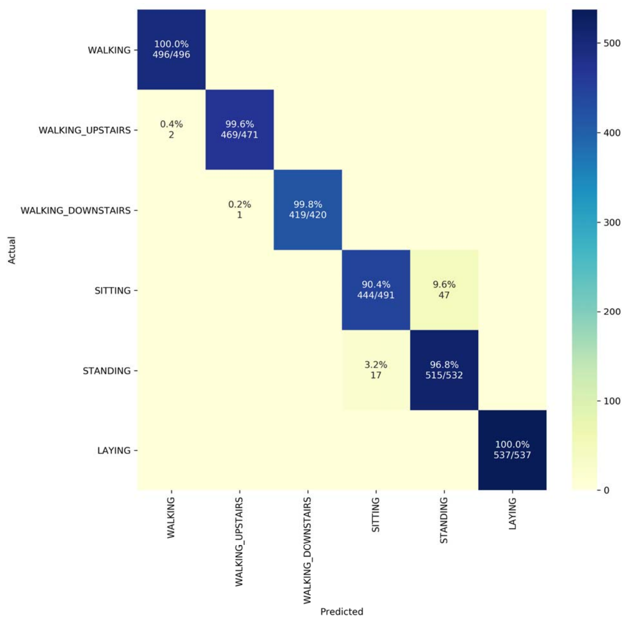

Fig. 3. Confusion matrix of KNN with dimension reduction

2) Support Vector Machines(SVM): Support Vector Machines is a machine learning model and it's a portrayal of the points in space, mapped with the goal that instances of the distinct classifications are isolated by the clear gap which is as large as possible. Then the new examples are also mapped into that equivalent space and a class of these examples is dependent on the side of the gap they are mapped. In other words, given labeled training data, the algorithm outputs an ideal hyperplane which categorizes the new data. Apart from doing the linear classification, SVM can also do a non-linear classification effectively using the different kernels, by mapping their contributions to high-dimensional element spaces. Even though SVM is used for both supervising and without supervising, utilizing supervised SVM is normally quicker and has more accuracy [11].

Support Vector Machine is a supervised machine learning algorithm that can perform an analysis of data for classification as well as regression. Although it can be used for regression, but SVM is mostly used for classification. We carry out plotting of data within the n-dimensional space. The value of each feature is additionally the esteem of the particular coordinate. At that point, we discover the perfect hyperplane that separates the two classes.

SVM algorithm utilizes a set of mathematical functions that are characterized as the kernel. The work of kernel is to require data as input and transform it into the specified frame. The Distinctive SVM algorithm utilizes distinctive sorts of kernel functions. These kernels can be of diverse types. For illustration linear, nonlinear, sigmoid, radial basis function (RBF)and polynomial. The kernel functions return the inner product between two points in an appropriate feature space. Hence by characterizing an idea of similarity, with small computational cost indeed in exceptionally high-dimensional spaces.

Labels of the class are indicated by -1 for negative class and +1 for a positive class in SVM.

$$y \in \{-1,1\} \tag{5}$$

The hypothesis of SVM model is given by equation.

$$h_{w,b}(x) = g(w^T x + b) \tag{6}$$

which is g(z) = 1 if z ≥ 0 and -1 for all other value of z.

Let the vector w will always be normal to the hyperplane. The equation of the hyperplane is as follows:

$$w^T x + b = 0 \tag{7}$$

The final optimization problem SVM solves to fit the finest parameters given that $y_i(w^T x_i + b) \geq 1, \forall x_i$ is:

$$min \frac{1}{2} \| w \|^2 \tag{8}$$

For each vector x_i such that:

- x_i belongs to class 1, then
$$w^T x_i + b \geq 1 \tag{9}$$
- x_i belongs to class -1, then
$$w^T x_i + b \leq -1 \tag{10}$$
- x_i belongs to hyperplane (decision boundary), then
$$w^T x_i + b = 0 \tag{11}$$

At the point when we used supervised SVM with the radial basis function kernel for classification of the dataset, the accuracy of 95.04% is achieved. But when the same model is used after dimension reduction, it gives 98.47% accuracy.

3) Convolutional Neural Network (CNN): Convolutional Neural Network is a deep learning model and it has ability to consider the spatial structure of the input information. CNN is a regularized version of the artificial neural network. Generally, artificial neural network refer to a completely connected network, in which, every neuron in a layer has a connection with all the neurons in the next layer. This property of complete contentedness of these type of networks makes them tend to overfit the data. To overcome this problem, some methods for regularization suggested to include a magnitude estimation of weights to the loss function. CNN also adopts an alternate action for regularization: they can exploit the different level design in data and gather progressively complex patterns utilizing smaller and easier patterns. CNN performs convolution operations instead of matrix multiplication. They were originally designed to work with images. CNNs have much lower training parameters as compared to standard neural networks which make it possible to efficiently train very deep architectures.

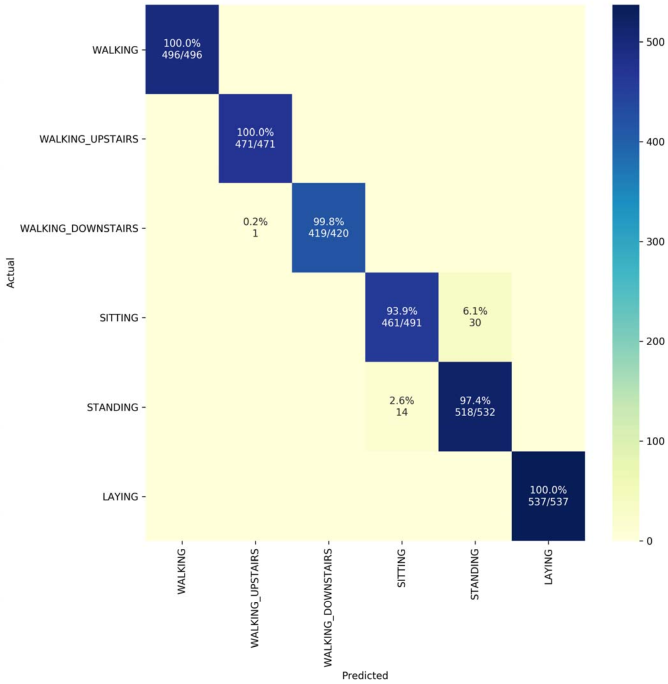

Fig. 4. Confusion matrix of SVM with dimension reduction

A CNN is generally consisting of an input layer, followed by hidden layers and one last layer as output layer. The hidden layers of a CNN regularly comprise of sequences of a convolutional layer that convolves with multiplication or other dot product. The activation function is generally a RELU layer, that is accordingly trailed by extra convolutions, for example, normalization layers, pooling layers (generally max pooling layers) and fully connected layers, implied to hidden layers because their inputs and as well as outputs are masked by the activation function along with final convolution. Final convolution, regularly includes backpropagation to all the more precisely weight the final result [13].

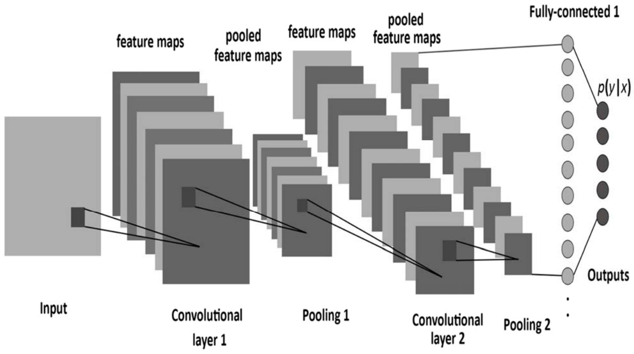

Fig. 5. Block Diagram of Convolutional Neural Networks [12]

A convolution neural network is generally made of three types of layers:

1) Convolution Layer

a) *Forward propagation*

$$a_{ij}^{(k)} = \sum_{s=0}^{m-1}\sum_{t=0}^{n-1} W_{st}^{(k)} x_{(i+s)(j+t)} + b^{(k)} \tag{12}$$

b) *Back propagation to update the weight*

$$\frac{\delta E}{\delta W_{st}^{(k)}} = \sum_{i=0}^{M-m}\sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta W_{st}^{(k)}} = \sum_{i=0}^{M-m}\sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} x_{(i+s)(j+t)} \tag{13}$$

$$\frac{\delta E}{\delta b^{(k)}} = \sum_{i=0}^{M-m}\sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta b^{(k)}} = \sum_{i=0}^{M-m}\sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} \tag{14}$$

c) *Back propagation to previous layer*

$$\frac{\delta E}{\delta x_{ij}} = \sum_{s=0}^{m-1}\sum_{t=0}^{n-1} \frac{\delta E}{\delta a_{(i-s)(j-t)}^{(k)}} \frac{\delta a_{(i-s)(j-t)}^{(k)}}{\delta x_{ij}} = \sum_{s=0}^{m-1}\sum_{t=0}^{n-1} \frac{\delta E}{\delta a_{(i-s)(j-t)}^{(k)}} W_{st}^{(k)} \tag{15}$$

2) Max Pooling Layer

a) *Forward propagation*

$$a_{ij} = max(0, x_{(i+s)(j+t)}) \tag{16}$$

b) *Back propagation*

$$\frac{\delta E}{\delta x_{(i+s)(j+t)}} = \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta x_{(i+s)(j+t)}} = \begin{cases} \frac{\delta E}{\delta a_{ij}^{(k)}}, & \text{if } a_{ij}^{k} = x_{(i+s)(j+t)} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

3) Fully Connected Layer

a) *Forward propagation of RELU activation function*

$$a_{ij} = max(0, x_{ij}) \tag{18}$$

b) *Back propagation of RELU activation function*

$$\frac{\delta E}{\delta x_{ij}} = \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta x_{ij}} = \begin{cases} \frac{\delta E}{\delta a_{ij}^{(k)}}, & \text{if } a_{ij}^{k} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

Where x is input, ak is output after convolution layer k, k is index of layer, W is kernel(filter), m ∗ n is filter size, M ∗ N is input size, b is bias, and E is Cost Function.

We have used CNN model with 6 layers where quantity of each layer is two. This model gives accuracy of 96.16%.

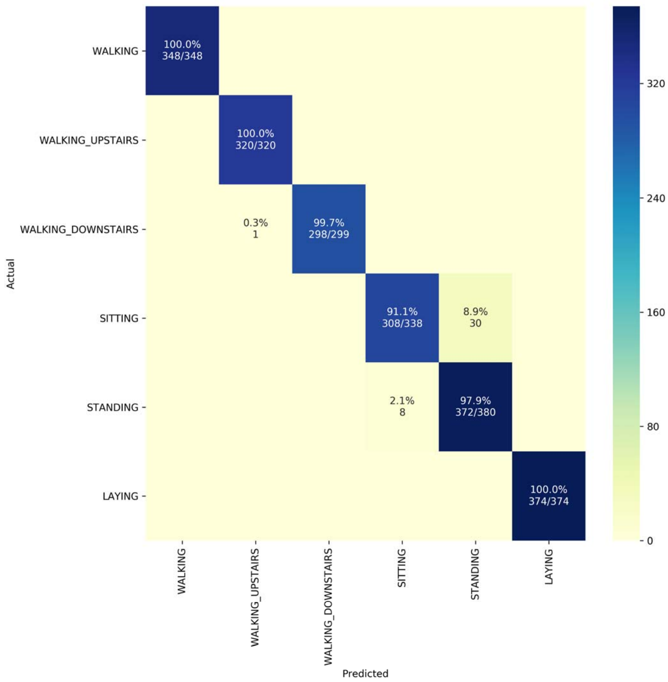

Fig. 6. Confusion matrix of CNN with dimension reduction

But if same model is used after dimension reduction success rate is 98.35%.

IV RESULTS

Accuracy of tried models are tabulated in Table 2 beneath. The most accurate methodology tried is SVM in this paper as found in below table, almost other two methods also make powerful models.

TABLE II. ACCURACY OF TESTED MODELS

Model	Accuracy
Support Vector Machines (SVM)	95.04%
SVM with Dimension Reduction	98.47%
K-Nearest Neighbors (KNN)	90.02%
KNN with Dimension Reduction	97.99%
Convolutional Neural Network (CNN)	96.16%
CNN with Dimension Reduction	98.35%

Bayat et.al. accomplished classification accuracy of 91.15% by utilizing data of accelerator. Anguita et. al. has also utilized the dataset in this work and accomplished a true positive rate of 96% by multi-class SVM model. One other paper that has utilized neural networks made the accuracy of 94.79% [14]. From these examinations, it very well may be said that strategies assessed in this paper exceptionally effective at recognizing human activity by smartphone data.

V CONCLUSION

In this paper dataset utilized consists of data produced from the accelerometer sensor and gyroscope sensor of the smartphone. The accuracy of classification can be increased by expanding the quantity of all activities and circumstances to the group and to include data collected from the other sensors and gadgets that are generally utilized in the smartphones to the data set. A portion of these gadgets is magnetometer, GPS, proximity sensor, light sensor, pedometer, barometer, thermometer. With the assistance of these gadgets, it is conceivable to get data about the area and condition of the client and the circumstance of the condition to order substantially more complex circumstances and activities.

REFERENCES

[1] E. Bulbul, A. Cetin, and I. A. Dogru, "Human activity recognition using smartphones," in 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–6, IEEE, 2018.

[2] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics, pp. 1–10, ACM, 2009.

[3] S. L. Lau and K. David, "Movement recognition using the accelerometer in smartphones," in 2010 Future Network & Mobile Summit, pp. 1–9, IEEE, 2010.

[4] V. Kon¨ onen, J. M¨ antyj¨ arvi, H. Simil¨ a, J. P¨ arkk¨ a, and M. Ermes,¨ "Automatic feature selection for context recognition in mobile devices," Pervasive and Mobile Computing, vol. 6, no. 2, pp. 181–197, 2010.

[5] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones.," in Esann, 2013.

[6] Omegatron, "Low-pass butterworth filters." https://commons.wikimedia. org/wiki/File:Butterworth orders.png, 2005. [Online; accessed October 12, 2019].

[7] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in Advances in neural information processing systems, pp. 1569–1576, 2005.

[8] R. Kohavi et al., "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Ijcai, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[10] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 986–996, Springer, 2003.

[11] F. Gieseke, "From supervised to unsupervised support vector machines and applications in astronomy," KI-Kunstliche Intelligenz¨ , vol. 27, no. 3, pp. 281–285, 2013.

[12] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," Entropy, vol. 19, no. 6, p. 242, 2017.

[13] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," arXiv preprint arXiv:1905.03554, 2019.

[14] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," Expert systems with applications, vol. 59, pp. 235–244, 2016.