IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2023, accepted 6 September 2023, date of publication 12 September 2023,
date of current version 18 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314651

RESEARCH ARTICLE

Exploratory Analysis of Smartphone Sensor Data for Human Activity Recognition

S.M. MOHIDUL ISLAM AND KAMRUL HASAN TALUKDER
Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

Corresponding author: S.M. Mohidul Islam (mohid@cse.ku.ac.bd)

This work was supported in part by the Information and Communication Technology (ICT) Division of the Bangladesh Government Ministry of Posts, Telecommunications, and Information Technology under the ICT Fellowship under Grant 22FS15324; and in part by the Khulna University Administration (by paying the publication fee partially).

ABSTRACT Precise recognition of human activities in any smart environment such as smart homes or smart healthcare centers is vital for child care, elder care, disabled patient monitoring, self-management systems, safety, tracking healthcare functionality, etc. Automatic human activity recognition (HAR) based on smartphone sensor data is becoming widespread day by day. However, it is challenging to understand human activities using sensor data and machine learning and so the recognition accuracy of many state-of-the-art methods is relatively low. It requires high computational overhead to improve recognition accuracy. The goal of this paper is to use exploratory data analysis (EDA) to deal with this strain and after analyzing, visualizations and dimensionality reductions are obtained which assists in deciding the data mining techniques. The HAR method based on smartphone accelerometer and gyroscope sensors' data, EDA, and prediction models proposed in this paper is a high-precision method, and its highest accuracy is 97.12% for the HAR smartphone dataset. Heterogeneous models-based two ensembles: stacking and voting are used in this study to identify human activities of daily living (ADL). Three estimators are used: Linear Discriminant Analysis, Linear Support Vector Machines, and Logistic Regression for both stacked and voting generalization. The experimental results show that the generalization algorithms provide an automatic and precise HAR system and can serve as a decision-making tool to identify ADL in any smart environment.

INDEX TERMS Activities of daily living, exploratory data analysis, hard voting, heterogeneous model, smartphone sensor, stacked generalization.

I. INTRODUCTION

Human activity recognition (HAR) has appeared as an interdisciplinary and exciting field of research with the connection of computer science, signal processing, and machine learning. HAR system can automatically detect, classify, and understand human activity. HAR systems empower the expansion of intelligent systems that can familiarize themselves with human actions, heighten personalized services, and make available valuable insights for observing and improving individual health and performance. With the explosion of wearable devices, ubiquitous sensing technologies, and the growing necessity for smart environments, HAR has gained noteworthy consideration as a vital issue in numerous practical spheres, including healthcare, surveillance, sports analysis, robotics, and human-computer interaction [1], [2]. For example, in healthcare, HAR techniques can be employed to monitor patients' physical activities, facilitating the early finding of anomalies and scheming personalized treatment strategies.

Activities of daily living (ADL or ADLs) is a term used in healthcare to denote the fundamental activities people perform in their daily life generally without the help of others such as walking, running, standing, sitting, bathing, dressing, walking upstairs, walking downstairs, lying, brushing, etc. [2]. ADL is used as an indicator of an individual's functional status. A person who cannot accomplish necessary ADLs may have a worse life quality or be risky in their

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

present life situations; hence, they may necessitate the help of other persons and/or mechanical devices [3].

Due to the widespread ease of use of smartphones in recent years, recognizing ADL using smartphone data has gained significant attention. Again, having built-in sensors such as accelerometers, gyroscopes, magnetometers, and GPS, smartphones have become powerful tools for collecting data about human activities. Moreover, researchers have achieved improved recognition performance and better discrimination between similar activities, by exploring the fusion of multiple sensors in smartphones [4], [5], [6], [7]. Sensor fusion techniques, such as feature-level fusion, early fusion, and late fusion have been used to combine data from multiple sensors meritoriously.

That means, existing works in HAR based on smartphones have established the potential of smartphones as trustworthy and handy tools to recognize human activities. Various data mining algorithms, sensor fusion approaches, and signal processing techniques have considerably contributed to improving performance and recognition accuracy.

Changes from the existing works which are typically fit for activity identification part in HAR systems, this study proposes a new and detailed exploratory data analysis (EDA) method for visualizing data to separate human activities. Using smartphone sensors' data for ADLs and combining the results of multiple heterogeneous models, we achieve activity recognition effectively. Our contributions are as follows:

- In this study, a detailed exploratory data analysis is outlined to deal with the recognition of activities of daily living based on smartphone sensor data. So far we know, no existing work has presented the analysis of HAR data in such detail. The detailed EDA highlights the internal characteristics of data so that the data analyst's knowledge of identifying activities improves more and deeper, or changes the understanding with the learning to figure out the real distribution of the activity data.
- We use boxplots of 'five number summary' to visualize the dispersion of data; histograms, and bar of probability distribution functions to find the inception for differentiating the steady and moving activities on univariate analysis. Moreover, we apply kernel Principal Component Analysis (kPCA) as well as T-distributed Stochastic Neighbor Embedding (t-SNE) manifold learning methods to investigate the separability of data on all features. These details EDAs assist in selecting a robust model for the HAR method.
- The HAR method based on smartphone sensor data, detailed EDA, and recognition from multiple heterogeneous models proposed in this paper is a new lightweight ensemble method. Although the machine learning method used here is lightweight, it is a high-precision method. This study attains higher prediction accuracy and lower training time in comparison with state-of-the-art shallow and deep models. Various model evaluation techniques are used to measure the performance of the method to authenticate the estimated results.

The rest part of this paper is prepared as follows. The second section reviews some state-of-the-art in the field of HAR and the use of smartphone sensors to capture data. The third section presents a new and details exploratory data analysis for the selected dataset and details modeling to deal with sensor data for ADL recognition. The fourth section illustrates and describes the experimental results and discussion with estimators and their hyper-parameters. The fifth part sums up the work of this study appeals to a conclusion, and plants some directorial thought for future study.

II. RELATED WORKS

In the past decades, HAR has become an active field of research and many researchers worked on HAR systems for building various HAR applications in smart environments. Generally, a HAR system consists of several common steps [7]: sensing activity data from environment or body sensors, pre-processing and labeling the activity data, segmentation using sliding window, feature extraction from time and/or frequency domains, and modeling using shallow and/or deep learning methods with or without transfer learning. As modeling is an essential and significant part of the HAR system, the selection of a classification model has a prominent effect on the overall precision of the system.

In the literature, there are two types of HAR systems based on the classification algorithms used. One prominent approach is based on the use of shallow learning algorithms. These algorithms learn from labeled datasets where each activity is associated with a specific set of sensor data features. Features commonly used include time-domain features, frequency-domain features, statistical features, and spatial features. Researchers have employed classifiers such as decision trees [7], [8], k-nearest neighbors (k-NN) [8], random forests [7], [8], artificial neural networks (ANN) [8], [9], support vector machines (SVM) [7], [8], [9], [10], etc. to recognize activities such as walking, running, sitting, standing, and cycling. These approaches have demonstrated promising results in accurately recognizing activities with high accuracy rates. Kong et al. [7] proposed a method based on six different shallow learning models and achieved the highest accuracy using linear SVC with the Grid search method of tuning hyper-parameters. They present some data analysis like ours but details exploratory analysis for the smartphone sensors data is not provided. Moreover, although their methods provide certain higher accuracy but require relatively higher training time. Masum et al. [8] captured data using a Xiaomi Redmi 4A smartphone, used PCA for selecting features, and applied several mining algorithms including Dense Neural Network, Decision tree, k-NN, random forests, SVM, and achieved the highest 94.38% accuracy for their prepared dataset. They compared the recognition results based on gender (Male and Female) which was not compared in any former research but their methods provided worse results for highly similar activities such as walking with walking downstairs

and/or walking upstairs. Khan et al. [9] acquired data using an LG Nexus 4 smartphone from five different phone positions in the body from 40 subjects, sampling at 6 different rates, and used data from 30 subjects for offline training and that of 10 subjects for real-time testing. They used kernel discriminant analysis to reduce class variance and ANN for modeling and achieved the highest 87.1% accuracy. They offered lightweight features that do not necessitate higher sampling rates and lengthier time windows for their calculation and so assist in attaining a fast response but those features are not fully position/orientation-independent of the phone such as the phone in the user's hands, in a carrier bag, in a coat's side pocket, etc. Moreover, their recognition accuracy is a bit lower than in many former works. Diney et al. [10] captured accelerometer data using an Android smartphone from a single subject and proposed an SVM model for the recognition of three activities of daily living. The authors developed the depiction of initially engendered vectors into compact clusters but captured data of training and recognition from only one subject, so it cannot be a widespread solution for HAR applications.

Another approach is based on the use of deep learning algorithms for HAR. Deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown remarkable performance in various recognition tasks including HAR. By leveraging the hierarchical representations learned from raw sensor data, deep learning models can automatically extract relevant features and capture complex temporal dependencies in activity sequences. Researchers have designed deep learning architectures for HAR [5], [6], [11], [12], [13], achieving state-of-the-art accuracy rates and robustness to different environments and user populations. Shi et al. [5] used the Boulic kinematic model to construct the dataset from body movement sensors and proposed a Deep Convolutional Generative Adversarial Network (DCGAN) and a pre-trained deep CNN architecture on ImageNet, VGG-16, deep model for recognizing three types of walking activities based on moving speed. This method is decent to expand and enrich training set to escape overfitting and acquire better results even in the case of higher similarity between activities such as fast-walking and really-fast-walking but the downside is that the author works for three types of walking activities only. Ravi et al. [6] proposed CNN models using three different regularizations for each of four different datasets: ActiveMiles, WISDM v1.1, Daphnet FoG, and Skoda and achieved 95.1%, 98.2%, 91.7%, and 96.7% for recognizing 2, 6, 7, and 10 activities respectively. They achieved consistent accuracy for real-time classification in low-power devices using their more discriminative and sensor orientation/placement invariant features for the datasets but their precision and computational times are not better than some former state-of-the-arts. Hammerla et al. [11] worked on three different datasets: Opportunity, PAMAP2, and Daphnet Gait, proposed five different deep models for each of the datasets, and obtained the highest 92.7%, 93.7%, and 76% accuracy for three datasets respectively. The authors presented a unique regularization method, explored the influence of hyper-parameters, and conveyed a recommendation for future researchers who may use deep learning models but their suggested setting doesn't show consistent performance for all benchmark datasets of HAR as well and their guidelines are limited to few deep models (DNN, CNN, and LSTM) and don't advocate whether they will work for other broadly used deep models such as Inception, Gated Recurrent Unit (GRU), etc. for HAR. Xu et al. [12] worked on 18 mid-level gesture activities from the Opportunity dataset, 18 lifestyle activities from the PAMAP2 dataset, and 6 activities of daily living for the dataset used in this study (HAR smartphone dataset) and achieved 94.6%, 93.5%, and 94.5% accuracy using Inception GoggLeNet and GRU for three corresponding datasets. Though this method provides better generalization and consistent performance than existing methods but doesn't explore the class imbalance problem in the data for real-life HAR applications. Bhattacharya et al. [13] proposed Ensem-HAR, where CNN-Net, CNN-LSTM-Net, ConvLSTM-Net, and StackedLSTM-Net are used as base models and Random Forest, is used as a meta-model of stacking and implemented their method on three different datasets including the one used in this paper and obtain 95.05% accuracy for HAR Smartphone dataset. Though their stacking of four deep learning-based models performs better than the other works to which it is compared, its accumulative training time of four different deep learning-based models is so high that it cannot be a typical method for real-time HAR applications.

In summary, although plenty of work has been completed to boost and optimize the models in HAR methods; still there are the following deficiencies:

(1) As we know human behavior of performing activity is not only usual and impulsive, but also human beings may perform some unrelated activities. Besides this, there are some variations of performing the same activity by different users. Another challenge is to handle the speed of movement in moving activities. We use a new EDA method to deal with HAR, which can effectively and accurately separate the activities.

(2) There are a variety of machine learning techniques. So selecting the best machine learning model is a challenge. The HAR method based on EDA can find a robust classifier for the dataset to reduce error with low training time.

In this paper, the domain knowledge is enriched by exploratory analysis of the data which in sequence helps to select a robust model for the activity classification task, which provides high precision results by minimizing the associated HAR problems like solving the misperception of highly alike activities such as walking and walking-downstairs. For evaluating the performance of the selected model, numerous comparative experiments are conducted and various performance metrics are used. The experimental outcomes show our

methodology overtakes state-of-the-art and reaches higher accuracy up to 97.12%.

III. METHODOLOGY

The methodology involves the following steps: data collection, data preprocessing and analysis (which includes data cleaning as well as exploratory data analysis to observe imbalance in the data and analysis on the single and multivariable), and finally modeling with sensor data. The conceptual figure of the proposed framework is shown in Fig. 1 and is described in the sections below.

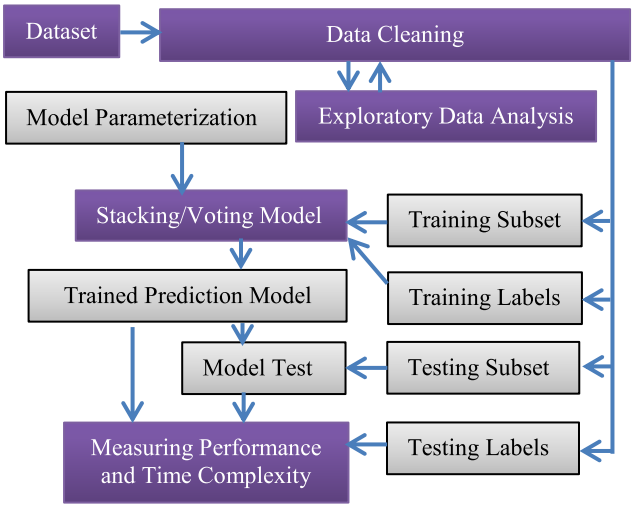


FIGURE 1. Conceptual figure of the proposed framework.

A. HAR SMARTPHONE DATASET

The dataset we used in this paper was collected from the UCI machine learning repository [14] and prepared by Anguita et al. [4]. The data is acquired from 30 participants whose age is between 19 and 48. Each person is asked to perform six activities of daily living: LAYING, SITTING, STANDING, WALKING, WALKING_DOWNSTAIRS, and WALKING_UPSTAIRS by wearing a waist-mounted Samsung Galaxy S II smartphone with embedded inertial sensors, accelerometer, and gyroscope. Using its built-in accelerometer and gyroscope, the three-axis linear acceleration and three-axis angular velocity are captured respectively at a uniform rate of 50Hz. Participants' activities are video recorded so that their activities can be labeled manually. The labeled data is randomly partitioned into train and test sets, where data from 21 participants are selected for generating the training data and the data from the rest participants are selected for generating the test data.

The noise is removed from raw sensor signals (acceleration and angular velocity) by applying a median filter and a third-order Butterworth low-pass filter with a corner frequency of 20 HZ and then segmented in sliding windows of constant-width of 2.56 seconds and 50% overlapping, i.e. 128 readings/window. The raw acceleration signal of the sensor is separated into two components: body acceleration and gravity acceleration by using another Butterworth low-pass filter with a cut-off frequency of 0.3Hz because gravitational force is supposed to have only low-frequency components.

From each segmented window, a feature vector is obtained by estimating variables from both time and frequency domains. The data of each feature are normalized and confined within [-1, 1]. Finally, each record of the dataset contains a 561 feature vector, its activity label, and an identifier of the user who carried out that experiment. Fig. 2 shows the total data in the dataset with their train and test data splitting for each activity and we see that about 70% of the total data is used for training and the rest is used for testing.

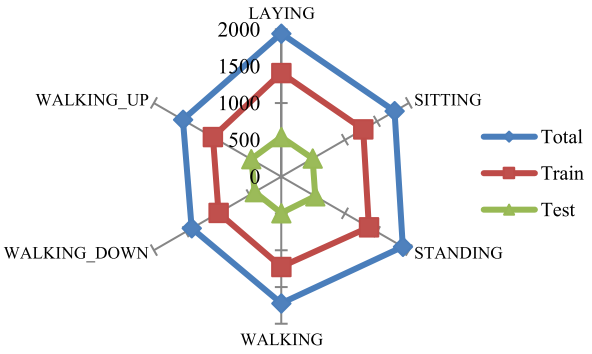


FIGURE 2. Splitting whole data into train and test sets.

B. DATA CLEANING

We perform initial data analysis for cleaning the feature data. From the data overview, we see that there is no outlier; all the values are bounded between −1 to 1. Again, we see that there are no duplicates in the train and test datasets. Moreover, we should not be worried about null values because there is no missing value present in the dataset and we find no feature with irregular cardinality. However the dataset contains some features that are irrelevant for machine learning modeling, so we remove those irrelevant features from the dataset.

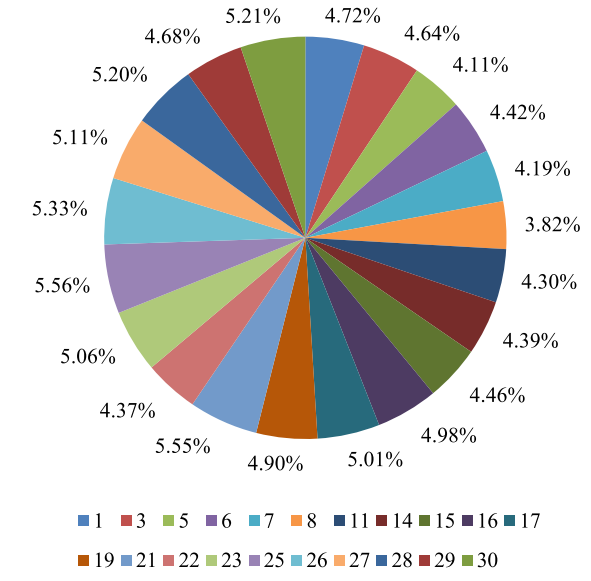
C. EXPLORATORY DATA ANALYSIS

Getting to understand the data is called Exploratory Data Analysis or EDA. It is a statistical way of perceiving and inferring the dataset. Usually, EDA comprises the following [15]:

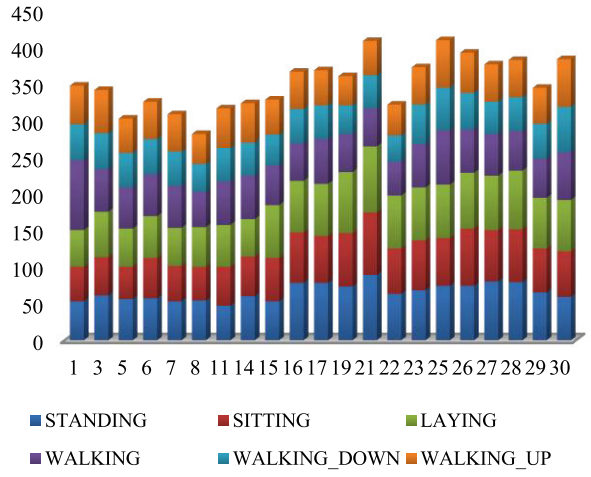
1) Observing data inequality among its various classes.
2) Univariate feature analysis of the dataset, and noticing the implication of a specific feature in classification using data visualization methods usually histograms or boxplots.
3) Multivariate feature analysis of combined features of the dataset, which is usually done using pair plots or dimensionality reduction techniques like PCA or t-SNE.

1) OBSERVING IMBALANCE IN THE DATA

Fig. 3 below shows the data provided by each subject. Fig. 3(a) shows the data percentage by all users and from this figure; we observe that each subject has almost the same number of data. We have only less data from user 8 (eight) compared to others but that's acceptable. So, we should not worry about the difference between them. Fig. 3(b) shows this in more detail, where data of each user is further highlighted based on the user's activities and we see that each user performs each activity in almost equal number of times which means there is no significant amount of gap in their readings.



(a) Ratio of data per subject

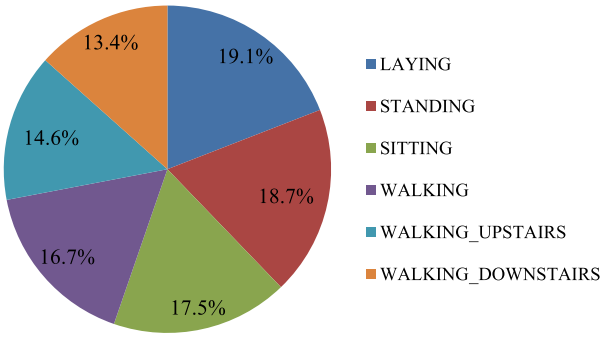


b) Data per activity for all subjects

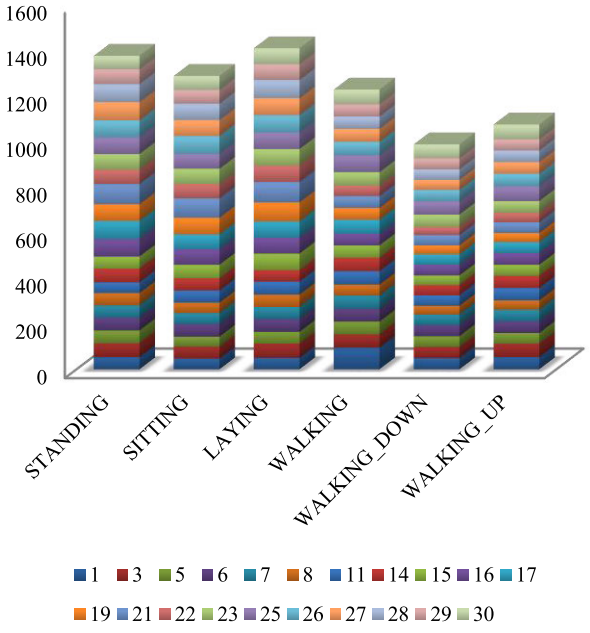
FIGURE 3. Data provided by each subject.

The imbalanced range of percentage of the subjects is 1.74%. So, we conclude that subject-wise data as well as activity-wise each subject data is balanced well.

Fig. 4 below shows the number of data points for each of the six activities of daily living. Fig. 4(a) shows the data percentage of all activities and from this figure; we observe that each activity has almost the same number of data. We have only fewer walking staircase data compared to others but that's reasonable. So we should not be worried about the difference between them. Fig. 4(b) shows this in more detail, where data of each activity is further highlighted based on each user and we see that each activity is performed by each subject in almost equal number of times which means there is no significant amount of gap in their readings.



(a) Ratio of data for six ADLs



(b) Data per subject for all activities

FIGURE 4. Data for each activity class.

The imbalanced range of the percentage of the activities is 5.7%. So, we conclude that activity-wise data as well as subject-wise activity data is balanced enough.

2) UNIVARIATE FEATURE ANALYSIS

Analysis of a single dimension or feature is known as univariate analysis. We performed the following univariate analysis:

a: FEATURE/SENSOR IMPORTANCE FROM DOMAIN KNOWLEDGE

Stationary activities (lying, standing, and sitting) are those where there is no motion of an object. Moving activities (Walking, Walking Upstairs, and Walking Downstairs) are those where there is the motion of an object. That means, in motionless activities, accelerometer information will not be very significant whereas in motion activities accelerometer information will be useful. Fig. 5 shows the number of features in the dataset that come from the accelerometer and gyroscope sensor of the smartphone. As we see, most of the features are constructed from accelerometer sensors so moving activities will easily be distinguished.

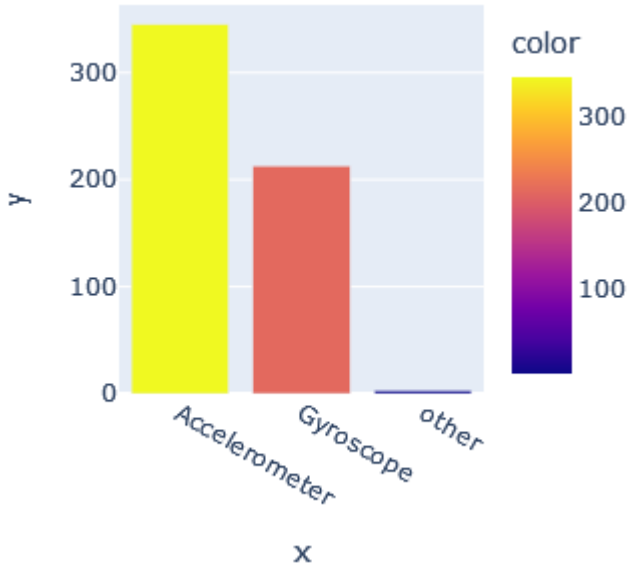


FIGURE 5. Number of features from various sensors.

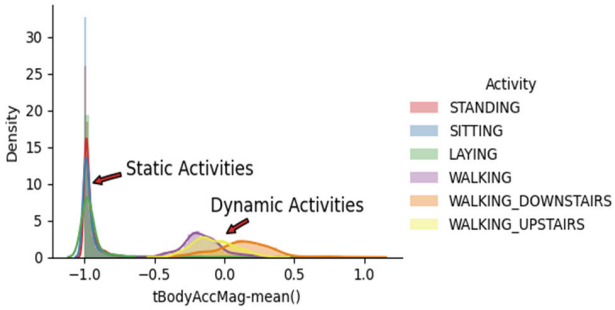
b: STATIC AND DYNAMIC ACTIVITIES ARE UTTERLY DIFFERENT

Fig. 6(a) shows the probability density functions (PDFs) for six ADLs based on the feature 'tBodyACCMag-mean()' (mean value for the magnitude of acceleration in the time domain for body motion). From PDFs, we can observe the difference between motionless and motion activities.

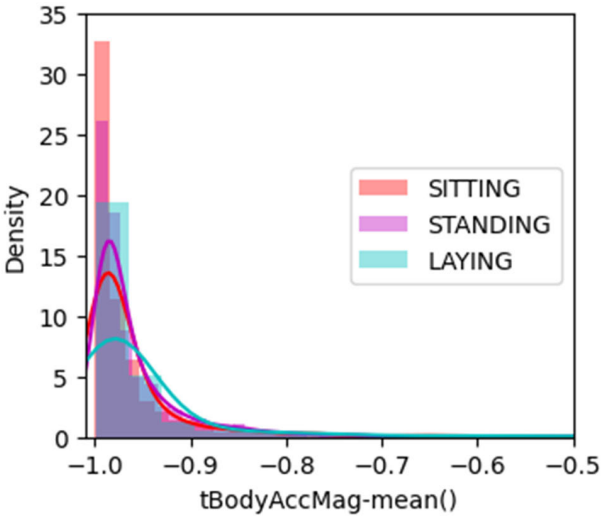
As per the PDF distribution of Fig. 6(a), we look closer by dividing the PDFs into two parts to distinguish inactivity and motion curves, shown in Fig. 6(b) and Fig. 6(c) respectively. Comparing these two figures, we can find that motion activities are less intensive than motionless activities. Moreover, the ranges of feature data for both types of activities are very dissimilar.

c: BODY ACCELERATION CAN SEPARATE IT WELL

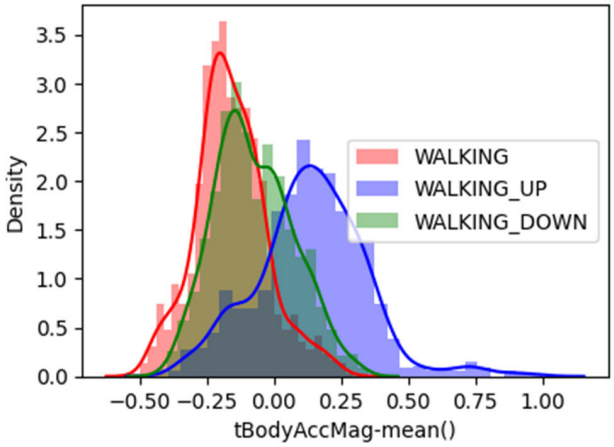
Fig. 7 shows the boxplots for six ADLs based on the feature 'tBodyACC-max()-X' (maximum value of acceleration along X-dimension for body motion). From the boxplots, we can observe the difference between stationary and moving



(a) Histogram of motionless and motion activities



(b) Closure view of motionless activities



(c) Closure view of motion activities

FIGURE 6. Histogram and its nearby view for two types of activities.

activities clearly by their dispersion. We can find that moving activities spread more than motionless activities. Moreover, we can separate those activities by simply using the following threshold statement: if (tBodyACC-max()-X $<$ -0.75) then Activity = "Static" else Activity = "Dynamic".

Also, using boxplot we can easily separate WALKING_DOWNSTAIRS activity from others: if (tBodyACC-max()-X $>$ 0.25) then Activity = "WALKING_DOWNSTAIRS" else Activity = "others".

But still, about 15% of WALKING_DOWNSTAIRS observations are below 0.25 which are misclassified so this condition makes an error of 15% in walking downstairs classification.

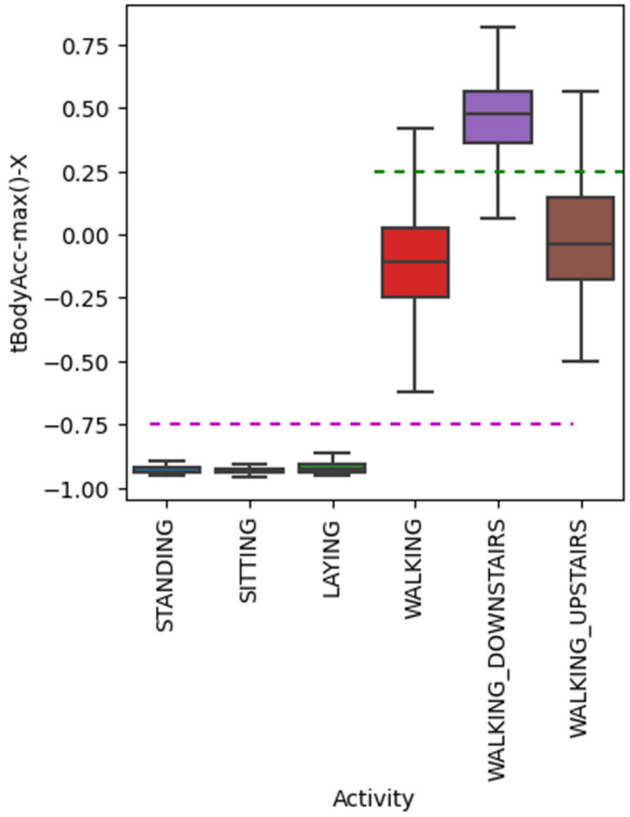


FIGURE 7. Data dispersion of six ADLs for body-acceleration maximum value along *X*-dimension.

Our analysis shows that not only the body acceleration value itself can separate the activities but also the jerk and magnitude of the body acceleration can separate them. Jerk or jolt is the rate at which an object's acceleration changes to time and magnitude is the absolute change in motion regardless of the direction of movement [16]. We observe also that standard deviation values in both time and frequency domains, as well as entropy in the frequency domain of jerk of body acceleration along the X-dimension, can separate the ADL. Last but not least, the mean value in the time domain of magnitude of body acceleration can separate human activities well. These are illustrated in Fig. 19 to Fig. 22, shown in Appendix A.

d: GRAVITY ACCELERATION COMPONENTS ALSO MATTERS

Gravity acceleration components can distinguish matting activity from others. Fig. 8 shows the data extent using boxplots and probability density function (PDF) for six ADLs based on the feature 'tGravityAcc-min()-X' (smallest value of gravity acceleration in the time domain along the *X*-axis). From both boxplots and PDF, we can observe that it perfectly distinguishes all data points belonging to the LAYING activity from other activities by just a single if-else statement: If (tGravityAcc-min()-X $<$ 0.35) then Activity $=$ ''LAYING'' else Activity $=$ ''others''.

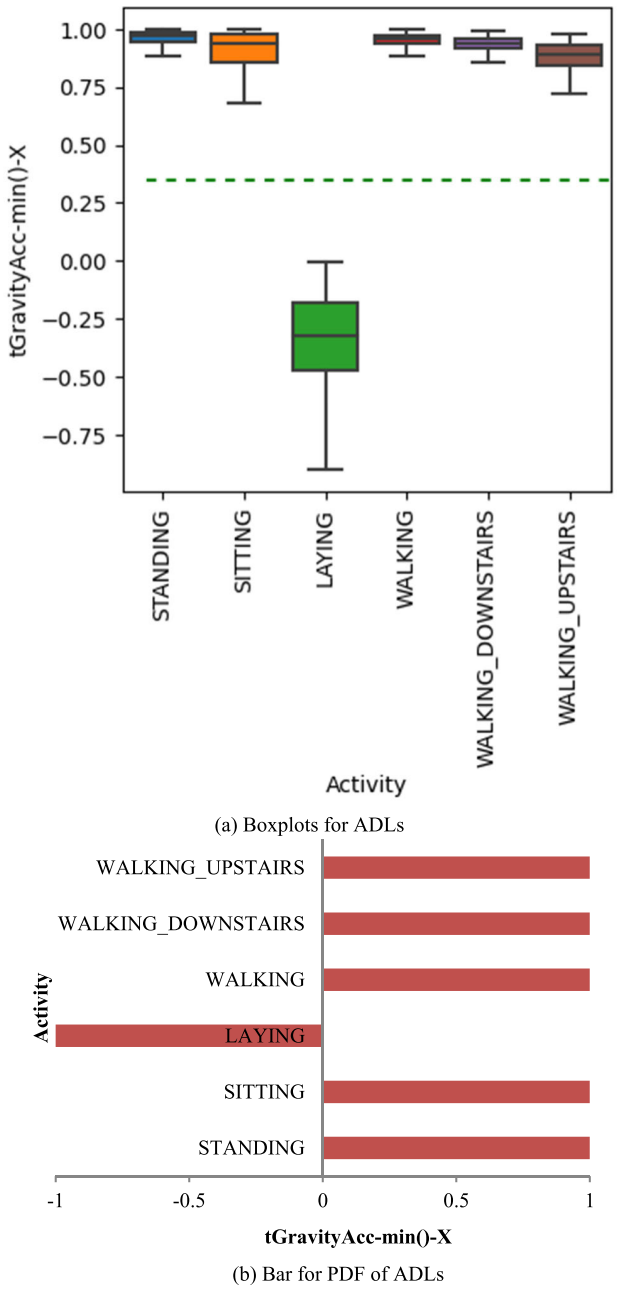


FIGURE 8. Boxplot and PDF of six ADLs for gravity-acceleration minimum value in the time domain along the *X*-axis.

Analysis shows not only the gravity acceleration min in the time domain along the X-axis but also the gravity acceleration max, gravity acceleration mean, and gravity acceleration energy (sum of the squares divided by the number of values) in the same domain along the same axis can separate the matting activity well. Last but not least, the angle between

the X-axis and gravity acceleration mean can also separate the in-bed activity from others. These are illustrated in Fig. 23 to Fig. 26, shown in Appendix B.

e: ANGULAR VELOCITY FROM THE GYROSCOPE IS ALSO A FACTOR

Though accelerometer data is significant to distinct static and dynamic activities, analysis pays attention to that gyroscope data in many cases can discriminate them. Fig. 9 shows the boxplots for six ADLs based on the feature 'fBodyGyro-entropy()-Z' (entropy value of angular velocity from gyroscope along Z-dimension in the frequency domain for body motion). From boxplots, we can see that moving activities can be clearly distinguished with a threshold value as follows: If (fBodyGyro-entropy()-Z $>$ 0.04) then Activity = "Dynamic" else Activity = "Static".

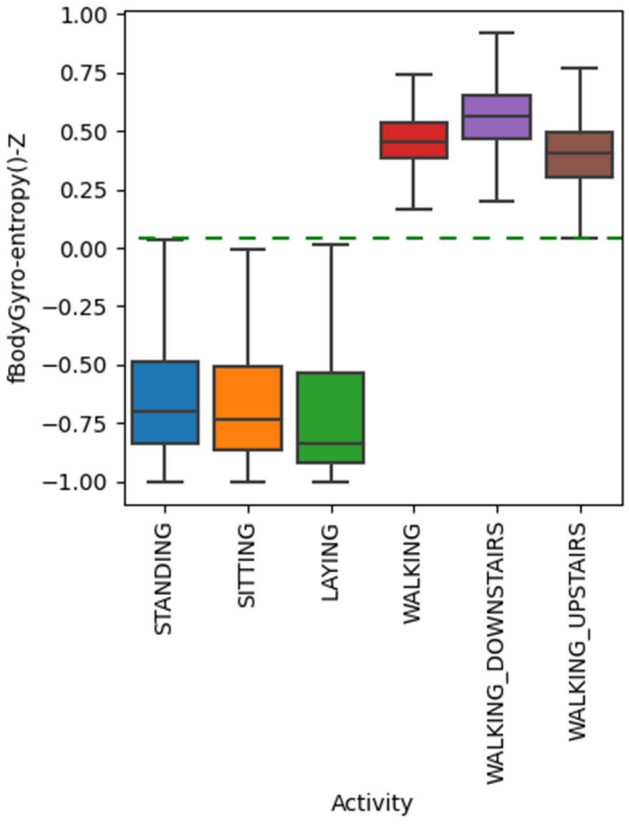


FIGURE 9. Data spread of six ADLs for gyroscope-entropy value along *X*-dimension in the frequency domain for body motion.

3) MULTIVARIATE FEATURE ANALYSIS

Analyzing multiple features together is called multivariate analysis. In the above, we perform analysis on a single feature; here we perform analysis over all 561 features (i.e. excluding 'subject' and 'Activity' features) to investigate the separability of the data through visualization using two non-linear dimensionality reduction techniques: Kernel PCA and T-distributed Stochastic Neighbor Embedding.

a: INVESTIGATING THE SEPARABLITY OF DATA USING KPCA

kPCA is an extension of PCA that achieves non-linear dimensionality reduction through the use of kernels to decompose a multivariate dataset in a set of components that explain a maximum amount of the variance [17]. In PCA the number of components is bounded by the number of features whereas in kPCA that number is bounded by the number of instances [18].

We have used polynomial as well as Radial Basis Function (RBF) as kernel and the resulting figures are shown in Fig. 10(a) for polynomial kernel degree, 9, and in 10(b) for RBF kernel coefficient, 0.05, respectively.

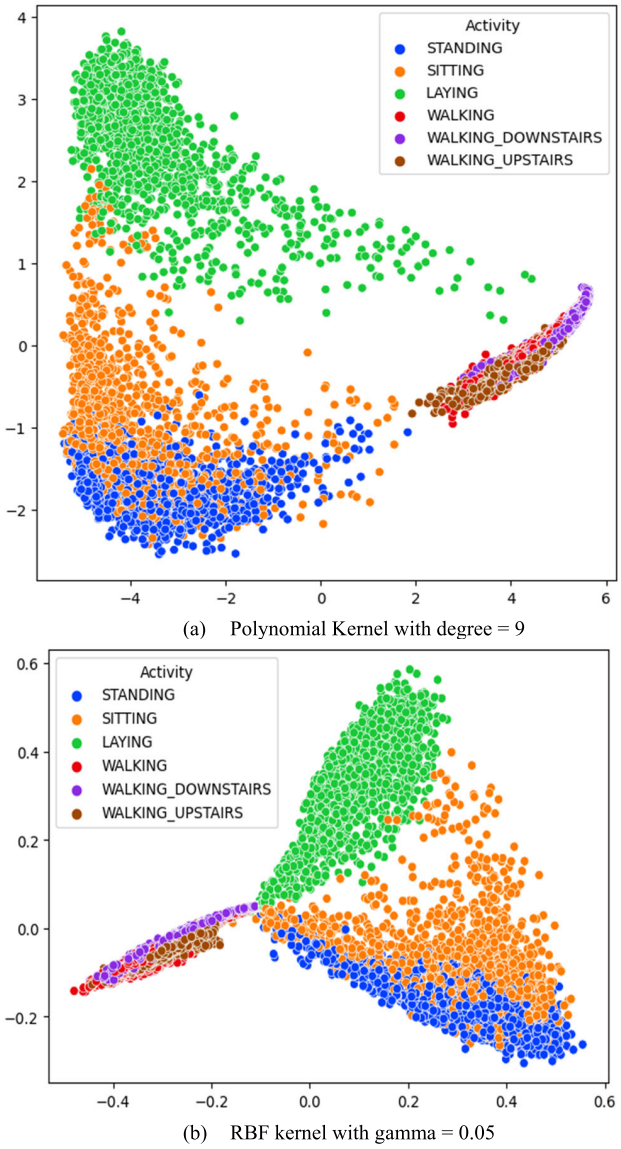


FIGURE 10. kPCA using two different kernels to separate the activities.

From both figures, we see that steady and moving activities can be separated very well. But each of the dynamic activities is not easily separable from each other whereas each of the static activities is easily separable from each other with some

errors in standing and sitting. That means, kPCA is good for separating each static activity but not good for separating each dynamic activity.

b: INVESTIGATING THE SEPARABLITY OF DATA USING T-SNE

t-SNE is another tool to observe the behavior of the data by visualizing them from an extremely high dimensional space to a compelling low dimensional space. Though it projects data to low dimensional space still it retains lots of actual information [19]. It does this by converting affinities between data points to Gaussian joint probabilities in the original space and it tries to minimize the *Kullback-Leibler* divergence by gradient descent between the joint probabilities of the embedding space and the original data. In the embedded space, affinities are represented by Student's t-distributions. This allows t-SNE to preserve the local structure which means data that is close in embedding space remains close, and the far remains far [20]. It's a powerful dimensionality reduction technique that reveals data to lie in multiple, manifolds or clusters.

Perplexity is the number of closest neighbors of each point t-SNE contemplates when producing conditional probabilities. The perplexity value has an impact on the optimization of t-SNE and therefore on the quality of the resulting embedding. That's why we analyze different plots with different perplexities: 2, 5, 20, 30, 50, 60, 80, and 100. A higher perplexity considers a larger number of neighbors and ignores more local information in favor of the global structure of data. Conversely, lower perplexities lead to smaller nearest neighbors and thus less sensitivity to global information in favor of the local neighborhood [20]. Four other factors control the performance of the resulting embedding: early exaggeration, learning rate, maximum number of iterations, and angle [20]. The resulting image for perplexity 80 with early exaggeration, 12.0, learning rate, 153.167, the maximum number of iterations, 1000, and angle, 0.5, is shown in Fig. 11.

We select perplexity 80 because it balances attention between local and global characteristics of data than other perplexities. For clarification, the figures for four other perplexities: 5, 20, 50, and 100 are shown in Fig. 27 to Fig. 30 in Appendix C.

In Fig. 11, we see the data points in 2 dimensions and we observe the behavior of those data points. We can see the six activities in three folds/clusters. Again we observe that all other classes are fairly separable instead of 'standing' and 'sitting' classes, because of similarities in sensor values, and it is expected because both are static actions. Maybe other sensors like the heartbeat sensor can assist in discriminating this because the heart rate is different at resting and standing poses. Laying activity is totally in a different position. Walking, Walking downstairs, and walking upstairs are some kind of similar so they are clustered together but separable from each other. So, t-SNE is good for separating each of both types of activities, especially all dynamic and matting activities.

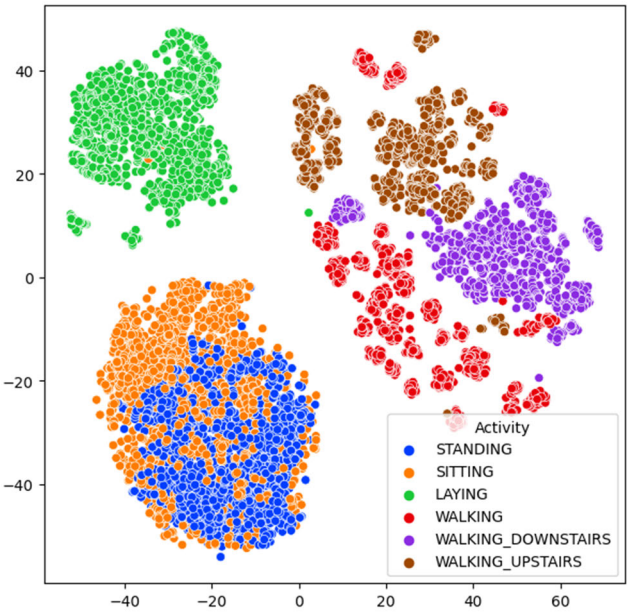


FIGURE 11. t-SNE with perplexity 80 for separating the activities.

D. MODELING WITH SENSOR DATA

We have used those two ensemble approaches where the final prediction result is obtained from multiple conceptually different or heterogeneous learning models: Stacking, and Voting. In both approaches, we have combined the predictions of three classical machine learning linear models: Linear Discriminant Analysis (LDA), Linear Support Vector Machines (LSVM), and Logistic Regression (Logit). These heterogeneous models are applied with the same hyper-parameters in both stacking and voting classification cases, but their learning and recognition strategies are different. Both stacking and voting classifiers improve generalizability or robustness over a single classifier [21]. In the below, the three estimators that are used in our both stacked and voting generalization are outlined with their hyper-parameters so that one can reproduce the result and then the learning and recognition process of both ensembles are delineated.

1) ESTIMATORS AND THEIR HYPER-PARAMETERS

The parameters that are not directly learned within models are called hyperparameters. They are provided as arguments to the model classes' constructors in *Scikit-learn* [22].

Linear Discriminant Analysis provides a linear decision boundary which is generated by fitting a class conditional Gaussian density to each class and using Bayes' rule [23]. As the dataset contains six ADLs, the desired dimensionality here is five. We have used the least squares solution as a solver and automatic shrinkage as a form of regularization (to improve the estimation of covariance matrices) using the Ledoit-Wolf lemma [24].

Support Vector Machines use only a small subset of training data in producing the decision boundary (called support vectors) [25]. We have used linear support vector classification i.e. SVM with linear kernel and so the multiclass is handled according to the one-vs-the-rest scheme. The maximum iteration used here is 500 with a random state of 42.

Logistic Regression or Logit regression is a maximum-entropy log-linear classification model in which the probabilities unfolding the potential outcomes of a single test are modeled using a logistic function [26]. We have used L2 regularized logistic regression with maximum iteration 50 and the multiclass is handled using cross-entropy loss.

The list of hyper-parameters for all the above baseline algorithms is shown in Table 1.

TABLE 1. Hyper-parameters of the baseline estimators.

Estimator	Hyper-parameters
LDA	shrinkage = 'auto' , solver = 'lsqr', priors=None, n_components=None, store_covariance=False, tol=0.0001, covariance_estimator=None
LSVM	random_state = 42, max_iter = 500, penalty='l2', loss='squared_hinge', *, dual='warn', tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0
Logit	max_iter = 50, penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

2) LEARNING AND RECOGNITION PROCESS OF STACKING CLASSIFIER

Stacked generalization is a stack of estimators (base estimators) with a final estimator (meta-estimator) which reduces the bias of individual estimators. This allows for combining the different strengths of all individual predictors [27], [28]. In our proposed method, among the linear models mentioned above the first two are used as base models and the third one is used as a meta-model, stacked in two different layers, as shown in Fig. 12.

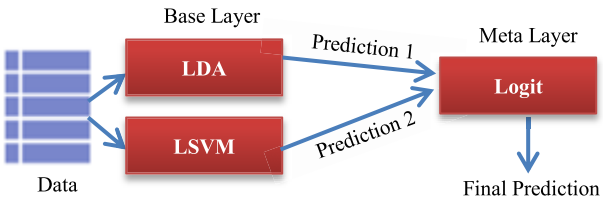


FIGURE 12. Proposed stacking model.

We first remove irrelevant features from both the HAR train and test dataset. During training, the whole train data are fed to the first base estimator, LDA, directly and to the second base estimator, LSVM, after standardization whereas

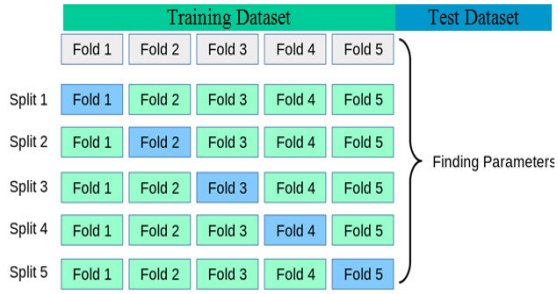


FIGURE 13. Training strategy of the meta-model.

the Logit model is fitted on out-samples through 5-fold cross-validation i.e. using cross-validated predictions of base models to generalize and avoid over-fitting. The training strategy of base-layer estimators is shown in (1) and (2), and that of the meta-estimator is shown in Fig. 13.

$$\text{fit (LDA)} = \text{LDA\_fit(x\_train, y\_train)} \quad (1)$$

$$\text{fit (LSVM)} = \text{LSVM\_fit(x\_train, y\_train)} \quad (2)$$

During recognition, the outputs of base-layer estimators, LDA and LSVM, are stacked together in parallel on the test data, and the Logit estimator in the second layer uses those outputs as input to compute the final activity class of the stacking model. The recognition process of those estimators is shown in (3), (4), and (5) respectively.

$$\text{predict (LDA)} = \text{LDA\_predict(x\_test, y\_}\textit{test}\text{)} \quad (3)$$

$$\text{predict (LSVM)} = \text{LSVM\_predict(x\_test, y\_test)} \quad (4)$$

$$\text{predict (Logit)} = \text{Logit\_predict (predict (LDA) , predict (LSVM))} \quad (5)$$

In the above equations (also in (6) to (12) below), *x_train* and *x_test* are the train and test data after removing irrelevant and target features from the training and test set respectively. Similarly, *y_train*and *y_test* are the target activity classes of those datasets. The *fit* and *predict* functions (in various forms) represent the learning and prediction by related models on given data correspondingly.

3) LEARNING AND RECOGNITION PROCESS OF VOTING CLASSIFIER

The voting generalization of the proposed method is a majority rule classifier that combines the three heterogeneous linear models mentioned above and uses a majority vote (hard voting) i.e. the mode of the predicted labels to recognize the human activity. This is useful to balance out the weaknesses of each model [21]. All models: LDA, LSVM, and Logit are worked in the same layer, as shown in Fig. 14.

After removing irrelevant features from the dataset, the same train data are fed to all estimators directly, except LSVM where data is fed after standardization. That means all estimators are trained on the whole training data. In the recognition stage, each estimator predicts the activity, and the final output is selected based on the maximum vote. The whole

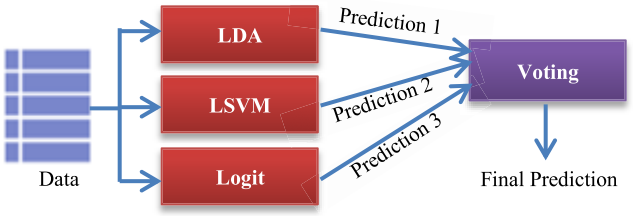


FIGURE 14. Proposed voting model.

voting strategy is illustrated in the following equations where (6), (7), and (8) show the training process of each estimator and (9), (10), and (11) show the prediction process of each estimator respectively and (12) shows the final recognition by voting classifier.

$$\text{fit (LDA)} = \text{LDA\_fit(x\_train, y\_train)} \tag{6}$$
$$\text{fit (LSVM)} = \text{LSVM\_fit(x\_train, y\_train)} \tag{7}$$
$$\text{fit (Logit)} = \text{Logit\_fit(x\_train, y\_train)} \tag{8}$$
$$\text{predict (LDA)} = \text{LDA\_predict(x\_test, y\_test)} \tag{9}$$
$$\text{predict (LSVM)} = \text{LSVM\_predict(x\_test, y\_test)} \tag{10}$$
$$\text{predict (Logit)} = \text{Logit\_predict(x\_test, y\_test)} \tag{11}$$
$$\text{predict (VOTING)} = \max(\text{predict (LDA)}, \text{predict (LSVM)}, \text{predict (Logit)}) \tag{12}$$

where *max* represents the majority class of the predictions.

IV. RESULT ANALYSIS

We have implemented our method using the *Scikit-learn* library for machine learning in Python 3.0. The following section describes our experimental results and later the comparison with the state-of-the-art and a discussion of the results is outlined.

A. EXPERIMENTAL RESULTS

Developing the machine learning model by properly tuning the hyper-parameters of the estimators, the heat map of the confusion matrix is obtained. Fig. 15 shows the confusion matrix for both stacking and voting classifiers. From the heat maps, we outline three remarks: (1) The correlation degree of the stacking model is higher than that of the voting classifier (2) The correlation degree of inactivity is higher than that of moving activity. (3) The correlation degree among static activities is higher than the correlation of any static activity with any other moving activity. This means that if any static activity is misclassified, it is misclassified to another static activity not to any motion activity. The opposite is also true.

Besides this, the detailed classification reports i.e. activity-wise precision, recall, F_1-score, and support, and their weighted averages are also obtained for both ensembles (Table 2 and Table 3). By analyzing these tables, we can find that the difference between precision and recall is not so high which indicates that for any human activity class, the classifier misclassified it to any other class as well as any other class is misclassified to it; in both cases, the error is almost equal and too low. The higher support value of static activities than that of dynamic activities focus on the disguise that human is too lazy to do exercise activities. The high average values of precision, recall, and F_1-score show the robustness of the models and we see that the stacking model is better than the voting model for the HAR smartphone dataset.

(a) Confusion matrix for stacking

(b) Confusion matrix for voting

FIGURE 15. Confusion matrix for ensemble models.

Again, the learning accuracy and learning time as well as recognition accuracy and recognition time of the proposed models are also found (Table 4). Training accuracy illustrates that models are not fully over-fitted, that's why we get comparable prediction accuracy for both models. Comparing the

TABLE 2. Classification report for stacking ensemble.

Activities	Precision	Recall	F_1-score	Support
Laying	1.00	1.00	1.00	537
Sitting	0.96	0.92	0.94	491
Standing	0.93	0.97	0.95	532
Walking	0.97	1.00	0.98	496
Walking_Down	1.00	0.97	1.09	420
Walking_Up	0.98	0.97	0.97	471
Weighted Average	0.97	0.97	0.97	2947

TABLE 3. Classification report for voting ensemble.

Activities	Precision	Recall	F_1-score	Support
Laying	1.00	1.00	1.00	537
Sitting	0.95	0.90	0.93	491
Standing	0.92	0.96	0.94	532
Walking	0.97	0.99	0.98	496
Walking_Down	1.00	0.97	0.98	420
Walking_Up	0.97	0.98	0.97	471
Weighted Average	0.97	0.97	0.97	2947

TABLE 4. Accuracy and required time for the ensembles.

Model	Train accuracy (%)	Test accuracy (%)	Train time (seconds)	Test time (seconds)
Stacking	98.98	97.12	36.535	0.101
Voting	99.05	96.71	10.022	0.120

TABLE 5. Comparison with state-of-the-art.

Reference	Model	Accuracy
W. Kong *et al.* [7]	Linear SVC + GridSearchCV	96.56%
C. Xu *et al.* [12]	Inception GoogLeNet + GRU	94.5%
D. Bhattacharya *et al.* [13]	Stacked model of CNNs and LSTMs with Random Forest	95.05%
F.J. Ordóñez *et al.* [29]	DeepConvLSTM	91%
J. Yang *et al.* [30]	CNN	91.5%
F. Cruciani *et al.* [31]	CNN	91.98%
N. Nair *et al.* [32]	ED-TCN	94.6%
This study	**Voting**	**96.71%**
This Study	**Stacking**	**97.12%**

data within the table, we see that the stacking ensemble shows better results than the voting classifier in all cases except in the case of training time complexity, though the training time is at a negligible level. The proposed stacking classifier shows 97.12% accuracy.

Moreover, for clarifying the performance of the proposed models the Cohen's kappa score, Jaccard score, Mathew correlation coefficient, Hamming loss, and Zero one loss are also obtained and shown in Fig. 16. These metrics also show the robustness of both models and as before show little better performance for stacking model.

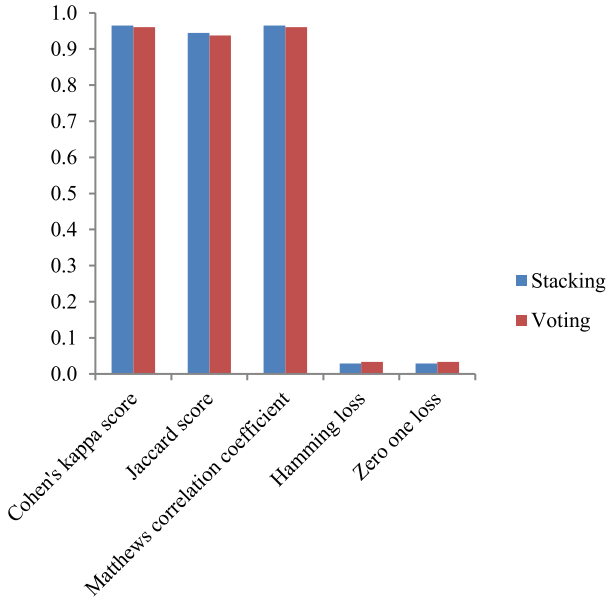


FIGURE 16. Five other classification performance metrics for proposed stacking and voting models.

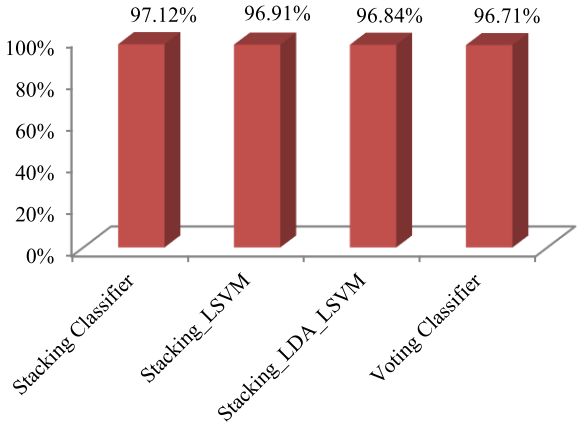


FIGURE 17. Recognition accuracy of all considered models.

TABLE 6. Training time comparison with state-of-the-art.

Reference	Model	Training time (mm:ss)
W. Kong *et al.* [7]	Linear SVC + GridSearchCV	01:10.761
This study	**Voting**	**00:10.022**
This Study	**Stacking**	**00:36.535**

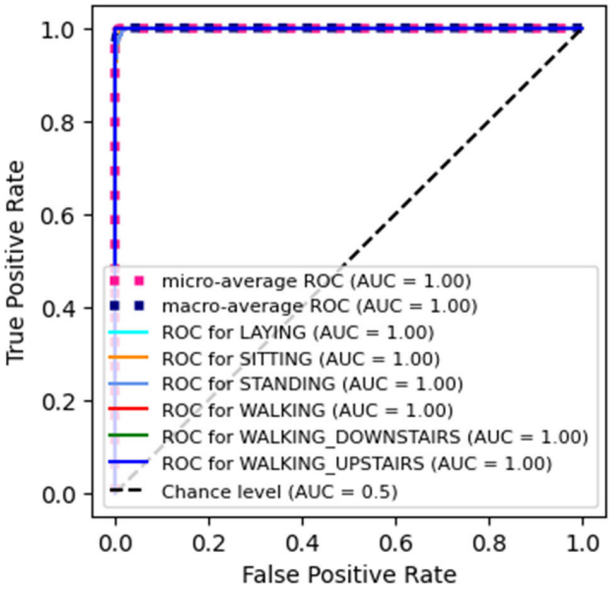
B. COMPARISON AND DISCUSSION

Some state-of-the-art who worked on the same dataset have been compared with the results of the proposed study, given in Table 5.

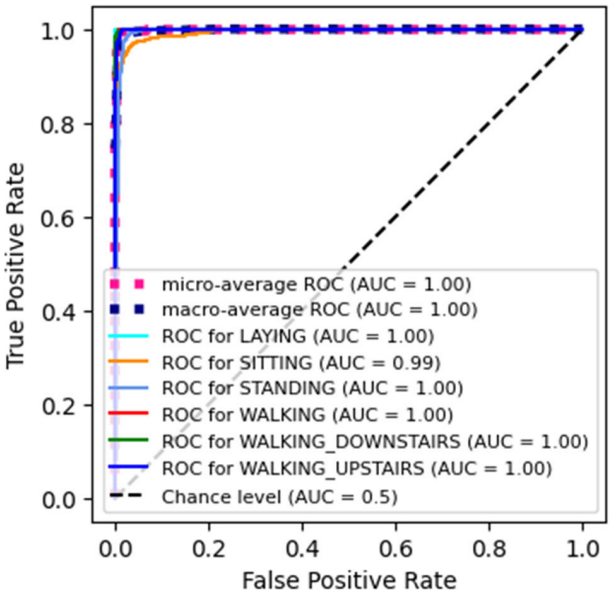
Comparing the models developed in this study with similar studies in the literature, it is seen that the results of this study are pleasing with an accuracy rate of 97.12% for the same

TABLE 7. Comparison of the proposed stacking model with its baselines.

Reference	Model	Accuracy
This study	LDA	96.34%
This Study	LSVM	96.71%
This study	Logit	94.88%
This Study	**Stacking**	**97.12%**



(a) Training ROC curves of each class and their average



(b) Testing ROC curves of each class and their average

FIGURE 18. Training and testing ROC curves of each activity class and their average for the proposed Stacking ensemble.

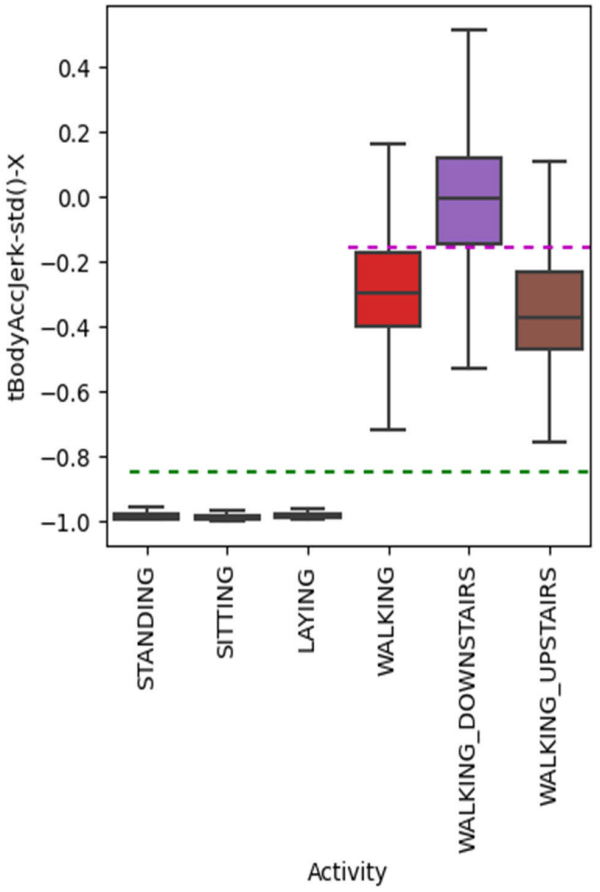


FIGURE 19. Data dispersion of six ADLs for the body-acceleration-jerk standard deviation in the time domain along *X*-dimension.

dataset. Again, as can be seen from the table, both of our ensemble methods outperform the results of state-of-the-art. We also observe that many of the existing works in the above table use deep learning as a model but our ensemble models show better results than them, so deep learning methods are not automatically every time the best selection when dealing with machine learning and sensory data models. Though our voting classifier outperforms a little, our stacking classifier outperforms a better amount of percentage for recognizing human activities of daily living from the HAR smartphone dataset. This result of the stacking ensemble concludes that stacked generalization can be a good choice for future HAR systems.

To promote the results and discussion of our study, we experimented with two other stacking classifiers: (1) using only the LSVM model as both base-layer estimators and meta-estimator (2) using LDA and LSVM as base-estimators and again, LSVM as meta-estimator. The result of all of our experimented methods is shown in Fig. 17.

From this figure, we see that the two newly considered stacking models show 96.91% and 96.84% accuracy respectively. That means all three stacking methods (the new two with the proposed one) outperform both our voting model as well as all state-of-the-art methods mentioned in Table 5.

Again, the highest accuracy is achieved by W. Kong et al. [7] among the existing methods listed in Table 5 and only that paper mentioned the training time of

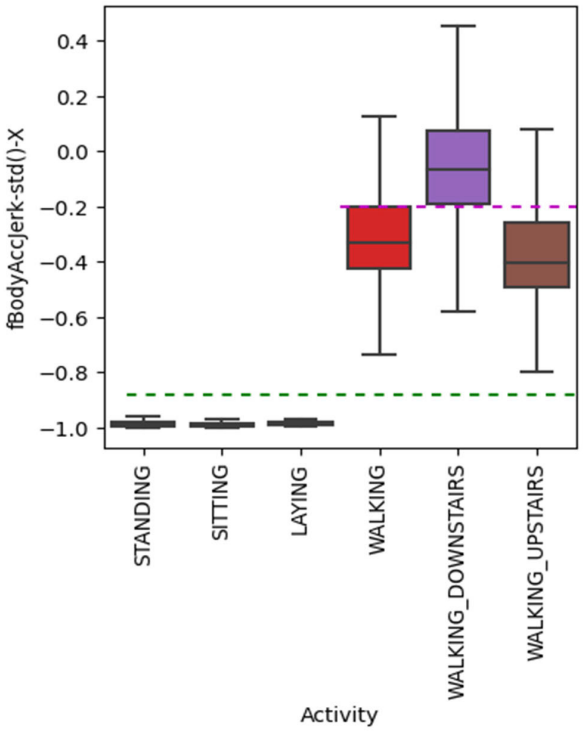


FIGURE 20. Data dispersion of six ADLs for body-acceleration-jerk standard deviation in frequency domain along*X*-dimension.

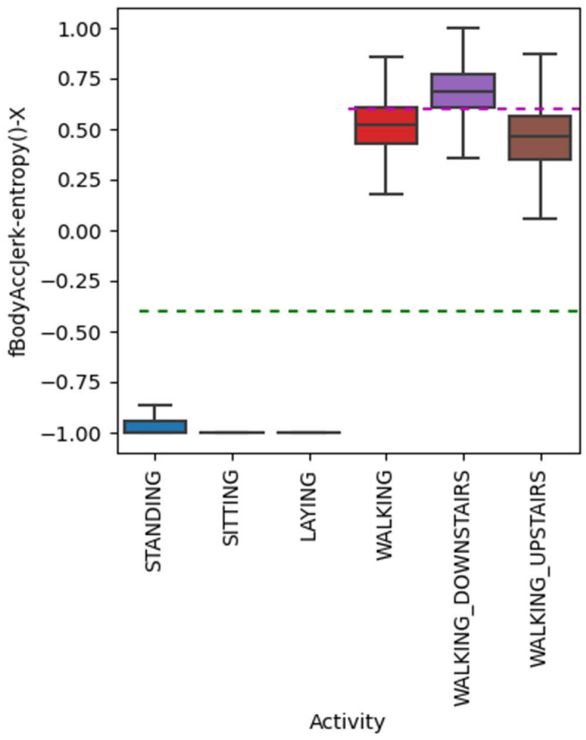


FIGURE 21. Data dispersion of six ADLs for body-acceleration-jerk entropy in frequency domain along *X* -dimension.

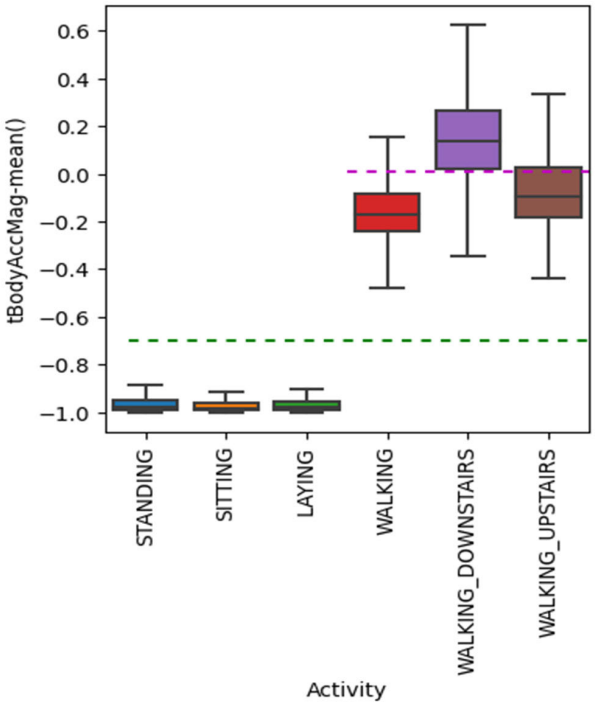


FIGURE 22. Data dispersion of six ADLs for the body–acceleration-magnitude mean in the time domain.

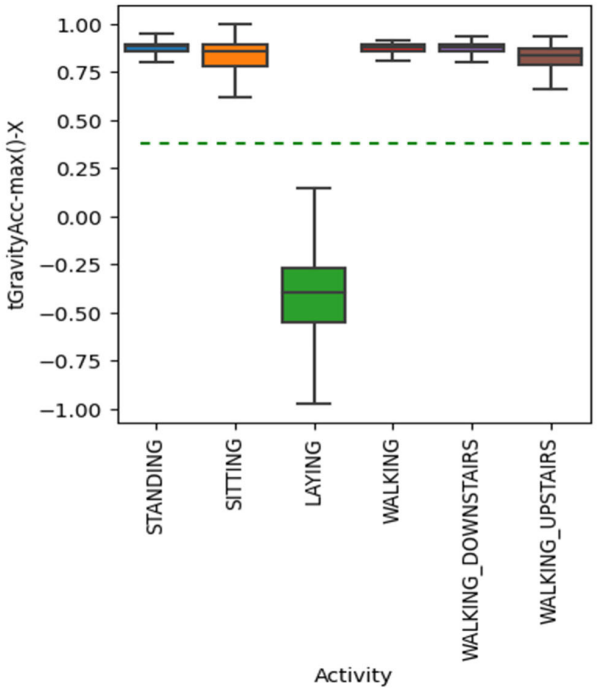


FIGURE 23. Data extent of six ADLs for gravity-acceleration maximum value in time domain along *X*-axis.

their used models. That's why the model training time of our proposed method is compared with that of [7], shown in Table 6.

The authors of [7] developed six different models in their work and among their models; Linear SVC with *GridSearchCV* shows the best accuracy. From the above table, we see that both of our ensemble models' training times are

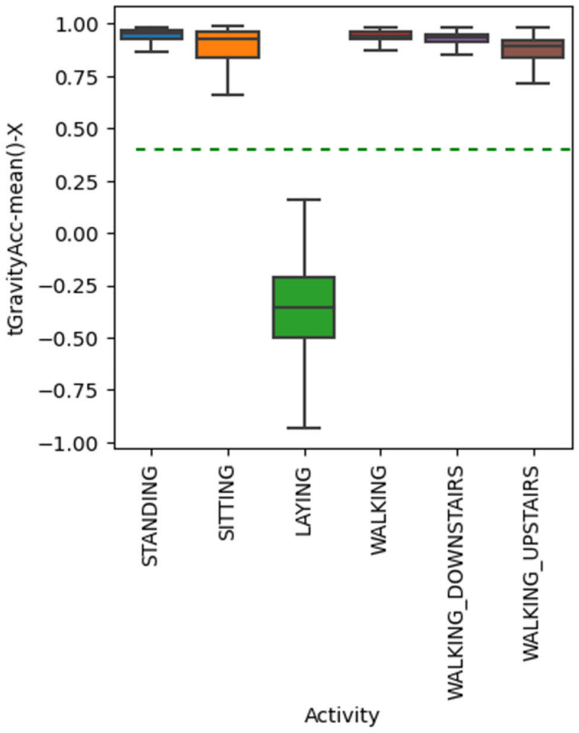


FIGURE 24. Data extent of six ADLs for gravity-acceleration mean in the time domain along the *X*-axis.

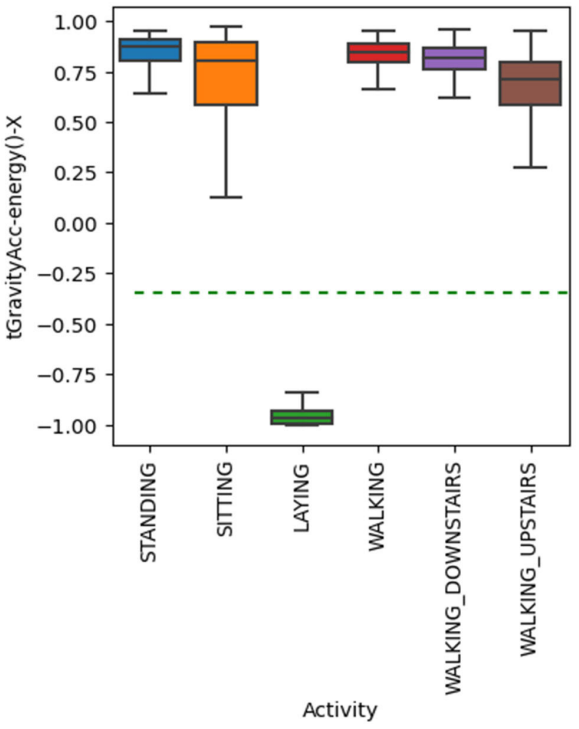


FIGURE 25. Data extent of six ADLs for gravity-acceleration energy in the time domain along the *X*-axis.

less than that of their best model. Therefore, the above two comparison tables show the strength of our models in terms of accuracy and time complexity.

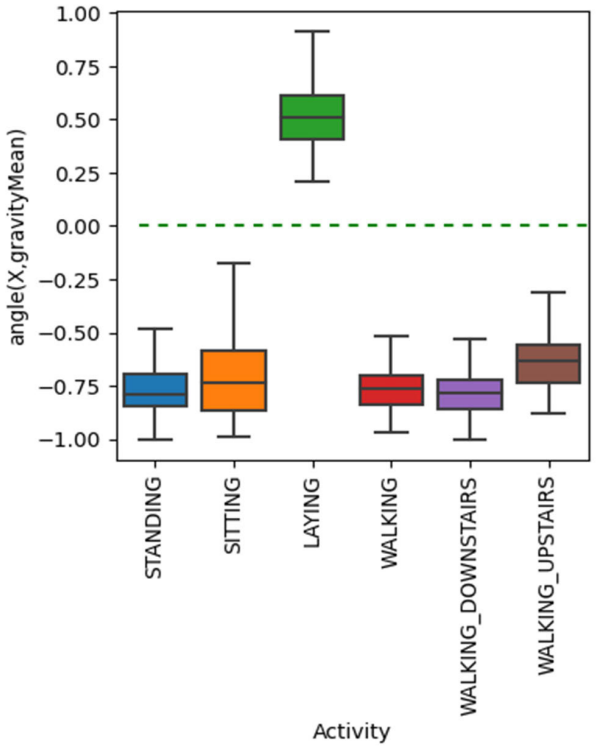


FIGURE 26. Data extent of six ADLs for the position of gravity-acceleration mean with *X*-axis.

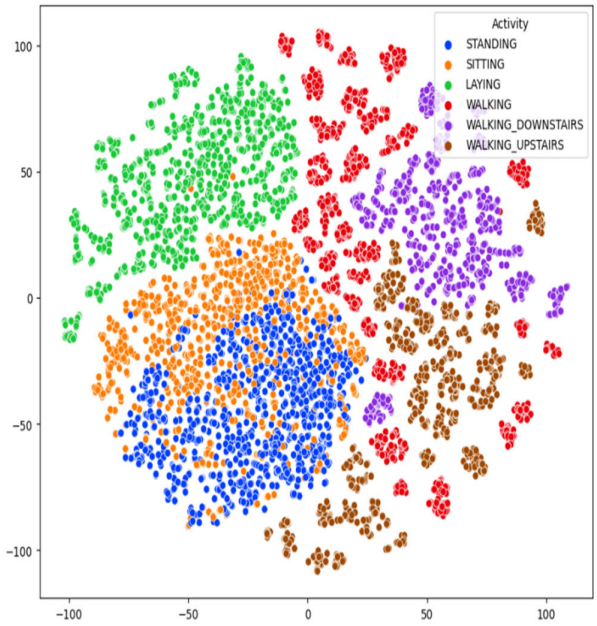


FIGURE 27. t-SNE with perplexity 5 for separating the activities.

Moreover, the accuracy of our best model (stacking model) is compared with its baseline algorithms in Table 7.

The above table shows that the stacking model provides better results than its baseline methods and so, this table focuses on the significance of organizing baseline models

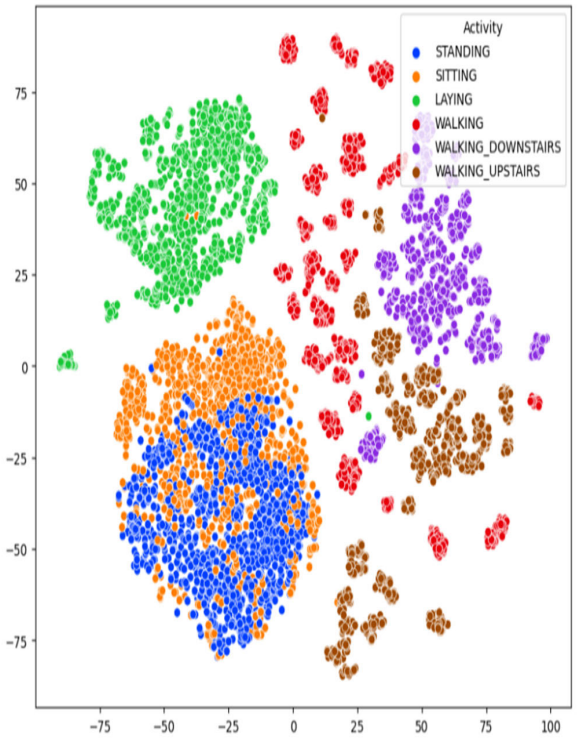


FIGURE 28. t-SNE with perplexity 20 for separating the activities.

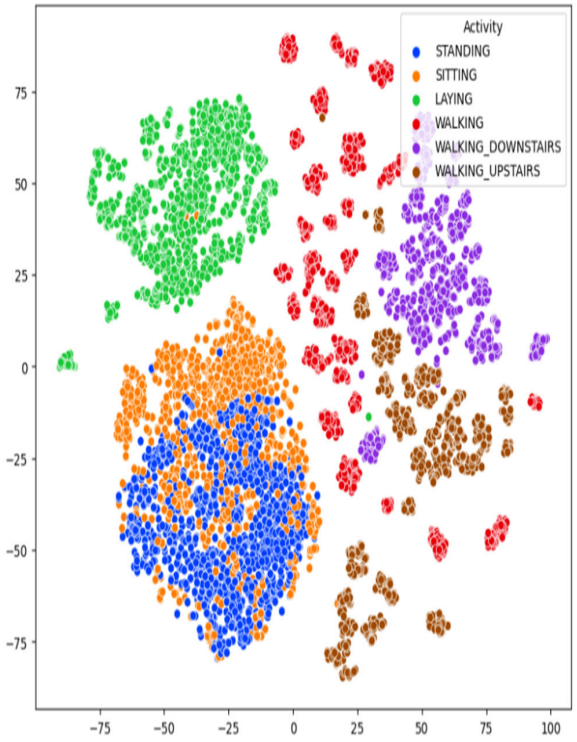


FIGURE 29. t-SNE with perplexity 50 for separating the activities.

according to the proposed stacking model framework, which is the best model than our other proposals and mentioned state-of-the-art.

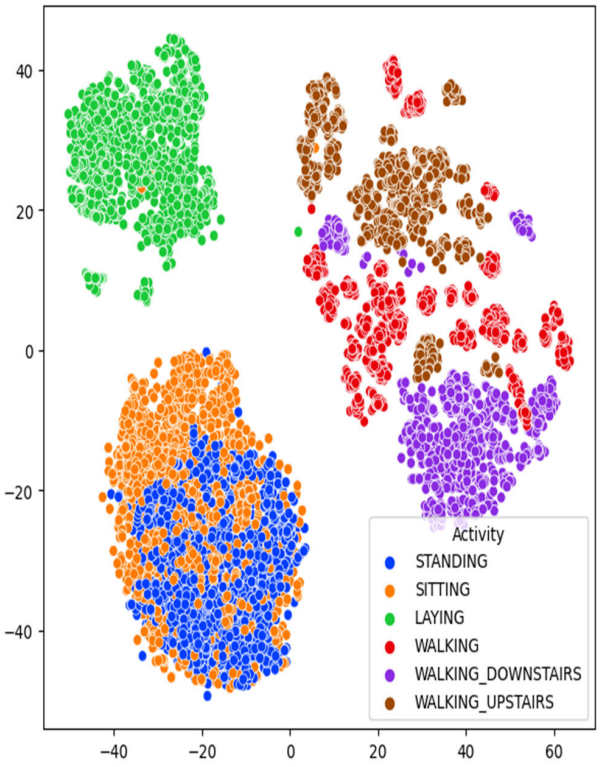


FIGURE 30. t-SNE with perplexity 100 for separating the activities.

Finally, the Receiver Operating Characteristic (ROC) curves for training and test datasets of our best model (stacking model) are shown in Fig. 18. From the two plots of Fig. 18, we see that the area under the curve (AUC) for each activity class and their micro-, as well as macro-average, is 1 for all cases, for both training and test set except 0.99 for that of SITTING class in the test set, which is very near to (area) 1. Therefore, these AUCs illustrate the robustness of the proposed stacking model again.

The above experimental results and comparisons demonstrate that the exploratory data analysis method has better generalization performance than the traditional data analysis method and as an estimator both the proposed stacking and voting classifier have good performance whereas stacked generalization has better performance than voting, no matter for different human activities or different evaluation strategies. These conclusions offer direction for the strategy of HAR methods using smartphone sensors in the future.

V. CONCLUSION AND FUTURE WORK

The proposed HAR method is a comparatively higher accuracy method. Additionally, the method of this study is lightweight, precise, and reasoning. Moreover, the exploratory data analysis outlined in this paper visualizes the detailed nature of activities, from which more new HAR systems can be developed. The experimental results show that our proposed activity visualization and identification method

has the potential to increase the performance of present HAR applications. Moreover, the solution can be extended to the classification of any multivariate time series data for other applications. As our experiment is carried out offline, our future target is to use our EDA to build and publish a real-time system to classify human activities. Again, as human beings may accomplish numerous activities at the same time, the future HAR hopes to be capable of recognizing parallel activities. Moreover, our future target is to extend HAR based on EDA to other human activities, human interactions, and relationships.

APPENDIX A
FIGURES MENTIONED IN THE SUBSECTION 'BODY ACCELERATION CAN SEPARATE IT WELL'

See Figs. 19–22.

APPENDIX B
FIGURES MENTIONED IN THE SUBSECTION 'GRAVITY ACCELERATION COMPONENTS ALSO MATTERS'

See Figs. 23–26.

APPENDIX C
FIGURES MENTIONED IN THE SUBSECTION 'INVESTIGATING THE SEPARABILITY OF DATA USING T-SNE'

See Figs. 27–30.

APPENDIX D
THE CODES OF THIS STUDY

The codes used to produce the outcomes of our study are available in the following Github link: https://github.com/SM-Mohidul-Islam/Exploratory-Analysis-of-HAR-Smartphone-Sensor-Data.

REFERENCES

[1] C. A. Ronao and S.-B. Cho, "Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Xiamen, China, Aug. 2014, pp. 681–686.

[2] E. Gambi, G. Temperini, R. Galassi, L. Senigagliesi, and A. D. Santis, "ADL recognition through machine learning algorithms on IoT air quality sensor dataset," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13562–13570, Nov. 2020, doi: 10.1109/JSEN.2020.3005642.

[3] P. F. Edemekong, D. Bomgaars, S. Sukumaran, and S. B. Levy, Eds., *Activities of Daily Living*. Bethesda, MD, USA: StatPearls Publishing, 2020, Accessed: Mar. 15, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK470404/

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. Esann*, Bruges, Belgium, vol. 3, 2013, p. 3.

[5] X. Shi, Y. Li, F. Zhou, and L. Liu, "Human activity recognition based on deep learning method," in *Proc. Int. Conf. Radar (RADAR)*, Brisbane, QLD, Australia, Aug. 2018, pp. 1–5.

[6] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. IEEE 13th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, San Francisco, CA, USA, Jun. 2016, pp. 71–76.

[7] W. Kong, L. He, and H. Wang, "Exploratory data analysis of human activity recognition based on smart phone," *IEEE Access*, vol. 9, pp. 73355–73364, 2021, doi: 10.1109/ACCESS.2021.3079434.

[8] A. K. M. Masum, S. Jannat, E. H. Bahadur, M. G. R. Alam, S. I. Khan, and M. R. Alam, "Human activity recognition using smartphone sensors: A dense neural network approach," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, Dhaka, Bangladesh, May 2019, pp. 1–6.

[9] A. Khan, M. Siddiqi, and S.-W. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones," *Sensors*, vol. 13, no. 10, pp. 13099–13122, Sep. 2013, doi: 10.3390/s131013099.

[10] P. Dinev, I. R. Draganov, O. L. Boumbarov, and D. Brodić, "Preprocessing and clustering raw accelerometer data from smartphones for human activity recognition," in *Proc. CEMA*, Athens, Greece, 2017, pp. 20–24.

[11] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. IJCAI*, New York, NY, USA, 2016, pp. 1533–1540.

[12] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902. 2019, doi: 10.1109/ACCESS.2018.2890675.

[13] D. Bhattacharya, D. Sharma, W. Kim, M. F. Ijaz, and P. K. Singh, "Ensem-HAR: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring," *Biosensors*, vol. 12, no. 6, p. 393, Jun. 2022, doi: 10.3390/bios12060393.

[14] *Human Activity Recognition Using Smartphones Data Set*. Accessed: Jan. 28, 2023. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[15] *Signal Processing With Machine Learning—Human Activity Recognition Part 1—EDA*. Accessed: Feb. 5, 2023. [Online]. Available: https://medium.com/analytics-vidhya/signal-processing-with-machine-learning-human-activity-recognition-part-i-eda-a1f3b0e91b63

[16] *Step By Step All Classification Model for Beginners*. Accessed: Feb. 7, 2023. [Online]. Available: https://www.kaggle.com/code/devson/stepbystep-all-classificationmodel-for-beginners

[17] B. Schölkopf, A. Smola, and K. R. Müller, "Kernel principal component analysis," in *Proc. ICANN*, Lausanne, Switzerland, 1997, pp. 583–588.

[18] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, Aug. 1987, doi: 10.1016/0169-7439(87)80084-9.

[19] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, Nov. 2008.

[20] A. J. Izenman, "Introduction to manifold learning," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 4, no. 5, pp. 439–446, Sep. 2012, doi: 10.1002/wics.1222.

[21] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. MCS*, Berlin, Germany, 2000, pp. 1–15.

[22] *Tuning the Hyper-Parameters of an Estimator*. Accessed: Feb. 17, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html

[23] R. H. Riffenburgh, "Linear discriminant analysis," Ph.D. dissertation, Dept. Stat., Virginia Polytech. Inst., Blacksburg, VA, USA, 1957.

[24] O. Ledoit and M. Wolf, "Honey, I shrunk the sample covariance matrix," *J. Portfolio Manage.*, vol. 30, no. 4, pp. 110–119, Jun. 2004, doi: 10.3905/jpm.2004.110.

[25] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998, doi: 10.1109/5254.708428.

[26] T. G. Nick and K. M. Campbell, "Logistic regression," in *Topics in Biostatistics*, W. T. Ambrosius, Ed. Totowa, NJ, USA: Humana Press, 2007, pp. 273–301.

[27] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.

[28] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2, 2nd ed. New York, NY, USA: Springer, 2009, pp. 1–758.

[29] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/s16010115.

[30] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. IJCAI*, Buenos Aires, Argentina, vol. 15, 2015, pp. 3995–4001.

[31] F. Cruciani, A. Vafeiadis, C. Nugent, I. M. P. Cleland, K. Votis, and R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Trans. Pervasive Comput. Interact.*, vol. 2, no. 1, pp. 18–32, 2020, doi: 10.1007/s42486-020-00026-2.

[32] N. Nair, C. Thomas, and D. B. Jayagopi, "Human activity recognition using temporal convolutional network," in *Proc. 5th Int. Workshop Sensor-Based Activity Recognit. Interact.*, Berlin, Germany, Sep. 2018, pp. 1–8.

S.M. MOHIDUL ISLAM was born in 1983. He received the B.Sc. and M.Sc. degrees (Hons.) in CSE, in 2007 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with Khulna University, Bangladesh.

He achieved an ICT Fellowship from Bangladesh Government for the M.Sc. degree in engineering research. He joined Khulna University, as a Faculty Member, in 2008. He has published several research papers in international journals and conferences. His research interests include human activity recognition, data science, machine learning, and smart technology. He is a Life Member of the Engineer's Institution Bangladesh (IEB) and the Bangladesh Computer Society (BCS). He is the former Joint Secretary of the Khulna Region of Bangladesh Open Source Network (BdOSN). He is the former Joint Secretary and a current Branch Member of the Khulna Branch of the Bangladesh Computer Society.

KAMRUL HASAN TALUKDER received the Bachelor of Science degree (Hons.) in CSE, the M.Sc. degree in computer science from the National University of Singapore (NUS), in 2004, and the Doctor of Engineering (D.Eng.) degree from Hiroshima University, Japan, in 2008.

He was the Head of the Computer Science and Engineering Discipline for three years. He joined Khulna University, as a Faculty Member, in 2000. He was a Postdoctoral Fellow with the Japan Society for the Promotion of Science (JSPS), Hiroshima University, for two years. He is currently a Professor with the Computer Science and Engineering Discipline, Khulna University. He is also the Dean of the Science, Engineering, and Technology School, Khulna University. He has published more than 70 peer-reviewed research articles over the years. His research interests include image analysis, software engineering, networking, and the IoT.