Pervasive and Mobile Computing 78 (2021) 101485

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

ADAM-sense: Anxiety-displaying activities recognition by motion sensors

Nida Saddaf Khan [a,*], Muhammad Sayeed Ghani [a], Gulnaz Anjum [b,c]

[a] Telecommunication Research Lab (TRL), Institute of Business Administration Karachi, Plot # 68 & 88 Garden/ Kayani Shaheed Road, Karachi 74400, Pakistan
[b] Intergroup Relations and Social Justice Lab, Department of Psychology, Simon Fraser University, Burnaby, Canada
[c] Department of Social Sciences and Liberal Arts, Institute of Business Administration Karachi, University Road, Karachi 75270, Pakistan

ARTICLE INFO

Article history:
Received 17 March 2021
Received in revised form 10 October 2021
Accepted 22 October 2021
Available online 9 November 2021

Keywords:
Human Activity Recognition
Anxiety Disorder
Deep learning
CNN-LSTM
Motion sensors
Healthcare
Sensors data analytics

ABSTRACT

In the field of Human Activity Recognition (HAR), human activities are recognized based on sensors' streaming data. HAR has been utilized widely in various fields of application where the studies of human behaviors are conducted such as healthcare, personal care, aged care, and several other domains. This approach can also be beneficial in the field of psychiatry where the patients suffer from mental, emotional, and behavioral disorder. According to American Psychiatric Association (APA), the most common form of mental disorder is Anxiety Disorder (AD) effecting 30% of the adult population at some point in their life. In this paper, a HAR based method is proposed to recognize some behaviors pertaining to anxiety display. To make such model, a novel dataset of anxious behaviors is also created with unique features using motion sensors of smartphone and Inertial Measurement Unit (IMU). Several deep learning-based models are created and compared against random forest and gradient boost algorithms, where a deep model comprising Convolution Neural Network (CNN) and Long-Short Term Memory (LSTM) is shown to perform better than other models and could recognize anxiety-related behaviors with over 92% accuracy.

1. Introduction

Human Activity Recognition (HAR) refers to the recognition of certain human behaviors related to the physical movements in performing an activity. It has made great progress in recent years due to the recent development in sensory technology and ubiquity of computing resources. There are mainly two types of recognition systems based on sensors, one is video based, in which cameras are used as a device to collect video recordings or still images to recognize the activities [1,2]. The other is non-vision based, in which various sensors are used to recognize activities. For non-vision based HAR, there are many types of sensors available for this purpose including motion sensors (e.g., accelerometer, gyroscope), direction sensors (e.g., magnetometer), connection sensors (e.g., Bluetooth), location sensors (e.g., GPS), and sound sensors. Among all smart sensors, accelerometer (A), gyroscope (G) and magnetometer (M) are the most widely used sensors for HAR. There are many reasons for their wide acceptance, such as that they are commonly available in

Abbreviations: HAR, AD; SDA, DL

* Corresponding author.
E-mail addresses: nskhan@iba.edu.pk (N.S. Khan), sghani@iba.edu.pk (M.S. Ghani), ganjum@iba.edu.pk (G. Anjum).

https://doi.org/10.1016/j.pmcj.2021.101485
1574-1192/

embedded devices like smartphones and smartwatches that eliminate the need for acquiring specific devices. Furthermore, these sensors have been used extensively in the field of HAR where they have achieved remarkable performance [3,4].

Neural networks and deep learning have outperformed other learning models in various tasks such as time series forecasting [5–7]. Deep learning has proven to be the most effective solution particularly in the field of HAR, due to its ability to learn complex patterns and temporal dependencies even from raw sensors' data [8]. Among these are the widely used Convolution Neural Networks (CNN) [9–11] and for temporal analysis the Recurrent Neural Networks (RNN) [12–14]. In HAR, the data has not only certain patterns that are repetitive in an activity but also have the temporal relationships among these patterns. This requires the mixture of learning capabilities of both CNN and RNN. Hence, there are various research works where the researchers have combined these two algorithms to recognize the activities with resulting greater accuracies [15,16].

It has been observed that some individuals perform certain physical movements in their state of anxiety. Previous literature has indicated that physical movements related to anxiety can be uniquely identified as associated with heightened arousal, and they are used for regulation of anxiety [17]. For instance, activities such as scratching, pulling hair or nails, hand tapping, muscle tension, higher heart rate, hyperactive behaviors, and obsessive compulsions can be distinguished from the lower arousal in other related psychiatric as well as non-psychiatric activities or states [17,18]. In comparison to these physical markers of anxiety, those for depression or prolonged stress are also unique [19], i.e., depressive markers would be difficulty in concentration, ruminative thoughts, lack of energy, loss of appetite, moving and talking more slowly, and sleeping much more or much less than usual [20]. Therefore, it is vital to note that sensory and behavior-based activities for anxiety are unique and possible to measure using the right tools. Moreover, if the pattern of such physical movements is learned, and frequency of its occurrence is monitored, then this can serve as an indicator of the presence of anxiety in the individual. Hence this study may be used in the development of a system to monitor occurrences of one or more of the proposed activities that the targeted individual is in a habit of performing during his/her state of anxiety.

In this research, a dataset is created comprising of eleven activities for possible 'anxiety displaying' behaviors. This dataset is unique in its structure as it is not collected in a strict lab-controlled environment and participants performed the activities according to their style and choice. Moreover, the activities are recorded for three possible body states (postures) i.e., sitting, standing, and lying down. To the best of our knowledge there exists no other dataset with such features. Furthermore, a novel deep learning-based algorithm is proposed to recognize such activities, so that this recognition may subsequently be used for designing the strategies to deal with the anxiety.

The remainder of this paper is organized as follows. Section 2 reviews the published research on the topic. Section 3 summarizes the process of data collection and its features. Section 4 describes the overall algorithm, system design and individual models constructed in this study for activity recognition. Section 5 presents the details of experiments, results, and their analysis on the dataset. Finally, Section 6 concludes the paper with a brief discussion.

2. Literature review

There has been a growing interest in the use of sensing technologies and machine learning for detecting psychological disorders such as stress, anxiety, depression, bipolar disorders, etc. [21]. In this section, the contributions and the limitations of various studies are highlighted to analyze the recent on-going research in this domain. Depression-based studies are first presented followed by Internalizing Disorder and Anxiety Disorder (AD) studies using body-worn sensors.

Detection of depression by body-worn sensors has been analyzed in [22,23], and [24]. In [22], late-life depression (depression in older persons) is studied using wearable accelerometers by associating it with the decline in physical activities. While in [23], depression is detected and monitored to provide the in situ intervention support using 3-dimensional accelerometers, Wi-Fi, GPS, and mobile phone usage data, to create the context information of subjects. Although, they were able to detect some behavioral patterns that may indicate depression, however they were not able to achieve very good accuracy (60.1%). Furthermore, they focused mainly on a walking activity, to find out its duration. Similarly in [24], authors have analyzed the specific behaviors related to sleep and communication patterns to detect the signals of depressive episodes. They used the sensors (accelerometer, gyroscope, GPS), and the communication log of the smartphone as inputs for a K-means clustering method, to model the normal behavior. Outlier detection was used to find the anomalies in a patient's normal behavior as indicative of depression.

Internalizing Disorder is a type of behavioral and emotional disorder in which the patients keep their problems to themselves. This disorder has also been studied in young children by wearable sensors and machine learning [25]. In this research, the participants' behavior was analyzed by eliciting fear responses, in which a snake is shown to the children suddenly, so that their instant behavior could be captured through motion sensors. Although this study has good accuracy, but it is limited in scope i.e., could only be used in very specific situations (fear). Furthermore, the study was carried out in a strict lab-controlled environment, lacking the ability to be applied in real-world situations.

The study of AD with wearable sensors has also been investigated and reported by many researchers [26–28]. [26] surveyed and presented many studies related to AD that have used wearable sensors for this purpose. These studies belong to various disorders of anxiety such as panic, post-traumatic stress, social anxiety, generalized anxiety, and OCD disorders. Although the authors in [26] presented many studies related to AD, however all of them were conducted using Electrocardiogram (ECG) which is not commonly available as are smartwatches or smartphones. Hence, these studies

Fig. 1(a). Sparkfun 9DoF Razor IMU M0.

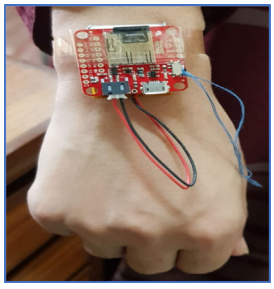

Fig. 1(b). Sparkfun 9DoF Razor IMU worn at wrist position.

required the employment of proper sensor setup in a highly lab-controlled environment and cannot be carried out in real-world situations. In [27], authors have suggested use of wearable devices to help in completing the daily log for exposure therapies. Exposure therapy is a psychological treatment in which people are helped to confront their fears. Authors have highlighted the issues in self-reporting by patients themselves and hence suggested use of audio, video, Heart Rate (HR), and GPS sensors for this purpose. Although, they have used wearable sensors, however their contribution is limited in scope and the choice of sensors have privacy, comfort, and availability issues.

In [28], researchers detected the span of anxiety using HR and Spontaneous Blink Rate (SBR), which are employed as wearable sensors. Here again the experiments were conducted in a strict lab-controlled environment where the subjects were involved in a task that triggers their anxiety so that the anxiety span could be studied. The dataset was also created in a strict lab-controlled environment where every aspect of the dataset was designed and communicated to the subjects. Moreover, the SBR was recorded using a device worn on eyes, which is not commonly worn by people. The band to measure the HR was worn at torso, which is again not a comfortable position of a body to wear a sensor device. Hence, making this dataset hard to be applied in real-world situations.

The presented studies have been conducted with the help of wearable sensors but their implementation in a real-world situation hinders due to the requirement of specific lab environment and the choice of the sensors that are not commonly available. These issues have been alleviated by our study in which we are using commonly available sensors to identify the behavior markers pertaining to anxiety. These behavior markers detect the presence of anxiety that may serve as a first step to the diagnosis and treatment of AD.

3. Data collection

Ten participants (one male & nine female, with ages ranged from 20–50 years) belonging to different backgrounds participated in our data collection process. Inertial sensors *A*, *G* and *M* were used to record the physical movements. The devices used for collecting sensors' data are smartphones and Inertial Measurement Units (IMU). The IMU units used in this study are Sparkfun 9DoF Razor IMU M0 shown in Figs. 1(a), 1(b). This unit is an Arduino-compatible, 32-bit ARM Cortex-M0+ microcontroller powered by LiPo battery and features three-axis sensors of *A*, *G* and *M*. The IMU is worn at the wrist position of the dominant arm whereas the smartphone is placed in the front pocket position of trousers of the alternate side of the body. The orientation of the devices is not kept fixed. Participants placed the phone in different orientations. For smartphone data collection, a publicly available Android app is used that records the sensors' stream along with the activity label in a csv file and stores it in the phone memory [29]. The sample rate or the frequency of the smart phone and IMU both are kept at 50 Hz. The raw sensors data of the participants were recorded for eleven behaviors as listed in Table 1 and are referred as anxiety-displaying activities in this paper for the sake of simplicity.

All the above activities were performed by all the participants and activity recording duration was kept at 2 min to capture the specific hand movement patterns associated with each activity. The activities were performed indoors

Table 1
Anxiety-displaying activities.

AID	Behavior	Named as activity
1.	Ear rubbing or scratching	*Ear_rubbing*
2.	Forehead rubbing or scratching	*Forehead_rubbing*
3.	Hair pulling	*Hair_pulling*
4.	Hands rubbing or scratching	*Hand_scratching*
5.	Hands tapping	*Hand_tapping*
6.	Knuckle cracking	*Knuckles_cracking*
7.	Nail biting	*Nail_biting*
8.	Nape rubbing or scratching	*Nape_rubbing*
9.	Smoking	*Smoking*
10.	Idle sitting and not performing any specific behaviors (including the above)	*Sitting*
11.	Idle standing and not performing any specific behaviors (including the above)	*Standing*

and outdoors with the locations based on the preference of the subjects. For the activities of sitting and standing, participants recorded the activities without doing any specific physical activity except talking occasionally. For other activities, participants were asked to conduct the specific activity in a natural way similar to how they usually perform these activities. For example, for hair pulling, some participants would pull the hair of their forehead, while some pulled the hair around the ears and while some pulled the hair around their nape. Similarly, it was observed that the pulling action was also performed differently by each subject. Some held their hairs in both hands while some twirled it along the length of their hair. This was the case with every activity where the subjects exhibit their natural pattern of conducting the activity.

The subjects were required to perform the series of activities in a specified order. The labeling was done manually via specifying the name of the log file that contained the data of an activity by each participant. The data collection process through both the devices was started simultaneously and the sampling period in both devices was the same. Two separate files were generated, and the corresponding rows were merged into a single file as the data from each device had been collected in sync.

Our dataset [30] has the following unique features: (a) it relates to anxiety-displaying behaviors containing a combination of both simple and complex activities, (b) it is collected in close to a natural environment, (c) the activities were performed in three different body states, i.e., standing, sitting, and lying-down states. These aspects of the dataset make it unique as to the best of our knowledge there are very few datasets, which are collected in a natural environment and none are available for anxiety-displaying behaviors and in particular with the inclusion of different body states. While our dataset has been collected in close to a natural environment, since the subjects were required to perform the activities in a specified sequence and the dataset lacks the other activities of daily living (ADL), we shall term it as a *semi-natural* environment. The reasons for terming this dataset semi-natural are:

- The positions selected for data collection are pocket and wrist. Researchers have also recommended other positions such as hips, arms, torso, ankles etc. However, we have only included the positions where people commonly keep phones and wear watches.
- The devices used in this study are smartphone and IMU (to emulate smartwatch) which are again common devices and do not require to purchase special sensors such as SBR etc.
- The collection was not carried out in any specific lab setting. The subjects performed them in the location of their choice.
- The subjects were not directed to perform the activities in any specific way. This was rather left on their choice, preference, style, and pace to perform it the way they usually do.
- The activities are recorded in three states of sitting, standing, and lying down.

4. Approach/methodology

In this section, the algorithm to recognize the anxiety-displaying behaviors is specified. Various deep learning-based models were used and evaluated on the dataset (discussed in Section 3). The algorithmic steps to clean, prepare, transform, and fit the model are summarized in Table 2 and explained in the following subsections.

4.1. Setting parameters & variables

The first step is to set the parameters and other variables required for learning the model such as combination of sensors, window size and other parameters etc. The details of these settings are given in Table 3. Window size represents the size of window for the method of sliding window that is used to transform time series into supervised learning dataset [31]. Determination of the combination of sensors is an important challenge in HAR. Accelerometer is considered to be the most useful sensor for activity recognition and it has been used solely for this purpose in other research

Table 2
Algorithm to recognize anxiety-displaying activities.

Step #	Detail
1. Setting parameters & variables	(a) Combination of sensors i.e., *A*, *G*, *M* (b) Position of sensors (c) Use of magnitude (T/F) (d) Splitting criteria (e) Window size (f) Overlap percentage i.e., 0%, 25%, etc.
2. Pre-processing	(a) Drop unnecessary fields e.g., device, etc. (b) Normalize (z-score) (c) Encode class to integer (d) One-hot encoding (e) Train/test split
3. Transformation	Reshape according to the type of the model, window size and overlap percentage.
4. Deep learning models	Apply the model on the training data and get evaluation on test data.

Table 3
Details of parameters and other variables used in the algorithm.

Settings/parameters	Values
Sensors	3-dimensional data of different combinations of *A*, *G* & *M*.
Positions	Wrist (dominant arm), Pocket (alternate side)
Magnitude	True or false
Epochs	10, 25 with early stopping
Splitting technique	Leave-one-out
Window size	Ranges from 60 to 600
Overlapping percentage	{0, 25%, 50%, 75%}, {70%, 80%, 90%}
Dropout	0.5
Early stopping patience	3, 5

works [13,32–35]. Several experiments were conducted with different combination of sensors worn at both positions. Due to the importance of accelerometer worn at the wrist position, this combination is included in all cases.

In the present form, the dataset has sequences of evenly spaced and ordered data collected at regular intervals by sensors. These sequences can be represented by tuples (t_i) where each tuple has 18 values comprising of the 3-dimensional *A*, *G*, and *M* values for each worn position of either wrist or pocket. Hence, t_i = (WAx_i, WAy_i, WAz_i, WGx_i, WGy_i, WGz_i, $WMxi$, WMy_i, WMz_i, $PAxi$, PAy_i, PAz_i, PGx_i, PGy_i, PGz_i, $PMxi$, $PMyi$, $PMzi$), where *W* represents wrist, *P* represents pocket and *i* represents the *i*th time step. These tuples must be reframed into supervised learning dataset by the method of sliding window so that a supervised model can be applied. In the sliding window method, a group of prior time steps are used to forecast the next time step [31]. But since HAR is not a forecasting problem, hence the group of prior values are used for classification of a label. The data is reframed into samples of fixed-length time series tuples, also known as windows, as input variables and the corresponding activity label as the output variable. *WS* represents the window size or the number of tuples and *OP* represents the percentage of overlapping tuples between consecutive windows. The Step Size (*SZ*), representing the number of tuples that are skipped in the next window, can be obtained easily by using *WS* and *OP* as given in Eq. (1a). After applying the sliding window method, the number of reframed samples (N) can be obtained by using *NR*, *WS* and *SZ* as given in Eq. (1b). In this equation, *NR* represents the total number of time series tuples. The ceiling function is applied to get an integer number.

$$SZ = WS\left(1 - \frac{OP}{100}\right) \tag{1a}$$

$$N = Ceil\left(\frac{NR - WS + 1}{SZ}\right), \text{ where } WS >= SZ \tag{1b}$$

The first window (W_1) comprises of the values from tuple t_1 till t_{WS}. The next window (W_2) comprises of the tuples from t_{SZ} till t_{WS+SZ} and so on. An example, with $WS = 100$, and $OP = 90\%$, gives $SZ = 10$. Now if the dataset has $NR = 10,000$ time series tuples, then $N = 991$ samples. This example of the sliding window method is illustrated in Fig. 2.

The optimal *WS* is considered to be an application-dependent task [36] and typically its depiction is based on two factors, i.e., type of activities and sample rate. The activities could be of two types as well, simple, or complex. For simple activities (e.g., sitting, running) a *WS* having duration of 2 s has been reported to be sufficient [37]. But for relatively complex activities like eating, drinking tea, smoking etc. a *WS* having duration of 2 s may not be sufficient due to the extended nature of such activities. Another factor that affects the *WS* is the sample rate. The sample rate, on the other

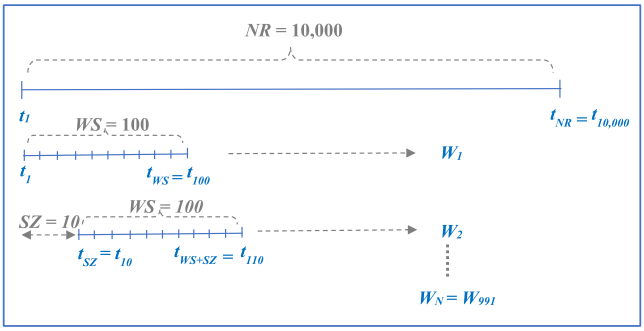

Fig. 2. Illustration of sliding window example.

Table 4
Required shape of input dimensions for each type of model where N = No. of samples (Eq. (1)), SC = Sensor channels, NS = No. of sub-sequences, SS = size of sub-sequences.

Model No.	*Model*	*Input shape*	*Dimension*
1	DNN	*(N, WS x SC)*	2
2	CNN	*(N, WS, SC)*	3
3	LSTM	*(N, WS, SC)*	3
4	CNN-LSTM	*(N, NS, SS, SC)*	4
5	ConvLSTM	*(N, NS, 1, SS, SC)*	5

hand, plays an important role in determining the complexity of the model. If the sample rate is high, then there would be larger number of values increasing the overall complexity of the model. Due to the importance of *WS*, various experiments are conducted in which the dataset was sampled with different window sizes and these are discussed in Section 5 of this paper. Another concept that is associated with the sliding window is the degree of overlap in adjacent windows. The use of overlapping has been reported to be useful and effective in activity recognition problems [36]. For this reason, several overlapping percentages between adjacent windows have been tried below in order to analyze the most effective *WS* along with the appropriate degree of overlapping. The number of epochs in the deep learning model has been kept between 10 and 25 during our experiments, as it was found that the models usually stop before 25 epochs due to early stopping. The early stopping and dropout features have also been used in our model to avoid overfitting of the model.

4.2. Pre-processing

In pre-processing of the dataset, at first the unnecessary fields are removed such as time, position, and device. The field of 'user' is kept in the dataset so that it can be used for splitting in training and test data later. After removal of unnecessary fields, the stream data values of three-dimensional sensors were normalized by using z-score.

Another pre-processing requirement of implementing a machine learning algorithm is to encode the class variable using one-hot encoding. For which a categorical class is converted into integer type at first and then one-hot encoding is applied. One-hot encoding removes the integer class variable and adds a new binary variable for each unique class label. This encoding converts a categorical type of class variable into a type where calculation can be performed and gets rid of the natural ordering of integer type encoding [38].

The dataset is split into training and test set using frequently used technique of leave-one-out, i.e., holding data of one user and using it for testing the model [8]. This helps to build a generalized model that is capable of recognizing the activities performed by any user. The model should be capable enough to learn the pattern associated with these activities and ignore the variations associated with each user.

4.3. Reshaping of data

Preprocessing is followed by data transformation according to the selected *WS* and *OP* so that a supervised learning model can be applied. Next, five reshaped datasets are created in conformation of different deep learning models. For example, a DNN model requires a 2-dimensional dataset whereas CNN requires the dataset to be in 3 dimensions. The reshaped data is a multidimensional array as given in Table 4.

4.4. Deep learning algorithms

To recognize the activities accurately several deep learning models have been applied that include deep Artificial Neural Network (DNN), CNN, Long-Short Term Memory (LSTM), and several hybrid models that are combinations of two of the above models. For each type of model, multiple architectures are created and evaluated to study the performance and behavior of each model. *NA* and *DL* acronyms are used below to represent the Number of Activities and number of neurons in Dense fully connected Layer respectively.

4.4.1. DNN

DNN is a deep Artificial Neural Network (ANN) model comprising of three types of layers, input layer, output layer and hidden layers. The input layer takes the input data from dataset and output layer produces the output in the desired format (class label). Where the set of hidden layers resides between input and output layer and does all the processing. Three types of deep ANN models have been used where DNN1 has one hidden layer of 100 neurons, DNN2 has two hidden layers of 100 and 50 neurons, and DNN3 has 3 hidden layers having 100, 50 and 50 neurons respectively.

4.4.2. CNN

Unlike a standard DNN, CNN have convolutional layers having filters that slide on the sub-regions of input data. They also have pooling layers, usually following the convolutional layers, and are used to obtain the reduced and summarized representation. Three types of CNN models are created to explore their feature extraction and classification capabilities. All three models differ in the architecture and use the following parameters: K = number of feature maps (filters), F = size of the filter, S = stride, and L = size of max-pooling filter. They are named as CNN1, CNN2, and CNN3. CNN1 has one convolution layer with $K = 64$, CNN2 has two convolution layers with $K = 64$ in both layers and CNN3 has two convolution layers with $K = 128$ in each layer. The convolution layer/s in each model is followed by a pooling layer. The other parameters are kept the same for all three models of CNN and are $F = 3$, $S = 1$, $L = 2$, $DL = 100$, and $NA = 11$.

4.4.3. LSTM

Simple Recurrent Neural Network (RNN) or vanilla RNN are known for their capability of learning temporal dynamics from sequential data by using cyclic connections. But their training is a challenging task due to the exploding or vanishing gradient problem for long-range sequences that hinders the ability to backpropagate gradients in the network [12,13]. Due to this reason, LSTM-based RNN has been used in this research. Three models of LSTM architecture are created namely LSTM1, LSTM2 and LSTM3. Each model has one LSTM layer having U equal to 20, 50 and 100 respectively. Where U corresponds to the number of units in an LSTM layer.

4.4.4. CNN-LSTM & ConvLSTM

As will be shown in the next section, the hybrid model where CNN and LSTM features are combined (model 4 in Table 4), turns out to be the most successful classifier for the activity recognition. Here the CNN layers provide the functionality of features learning whereas LSTM layers help in learning the temporal dynamics of the sensors' stream data. Such hybrid models have applications not only in HAR but also have been created in various application domain such as visual recognition [39], voice search task [40], computer vision and natural language processing [41].

In this research, three such models, based on hybrid models (models 4 & 5 in Table 4), have been created to explore their usefulness in recognizing the activities. These models are named as convLSTM, CNN-LSTM1, and CNN-LSTM2. CNN-LSTM model is created in a layered fashion i.e., the layers of convolution and LSTM are placed one after another. Whereas ConvLSTM uses layers of special units that performs the convolutional operation directly inside the LSTM units. The researchers in [42] have used this model for precipitation nowcasting. In this research, one model of ConvLSTM was created having a single layer of U (64) units followed by a fully connected layer of DL (100) units and a soft-max layer of NA units.

For CNN-LSTM models, the temporal sequence (window) having size WS is divided into small sub-sequences so that the local features can be extracted from sub-sequences and these extracted features are combined in the next step to get features for the whole sequence. The convolution applied to each sub-sequence share the weights i.e., the weights for one sub-sequence is learned and passed to other current convolutions layers. This reduces the number of weights required to be learned besides reducing the computation time. The idea is taken from video analysis, where a block of images are passed as a large input, where each single image is dealt as one subsequence [39]. The weights of a convolution layer are shared by other convolution layers for a sequence to learn a repetitive pattern. The temporal relationship associated with the repetition of this behavior needs to be analyzed on an entire sequence. For this purpose, an LSTM layer is applied on the whole sequence by combining the max pooling output in the form of a larger flattened layer.

The input for the CNN-LSTM models is a four-dimensional array of *(N, NS, SS, SC)* where NS is set to 4 to divide one temporal sequence into four sub-sequences. For example, if the temporal sequence has WS 60, then it will be divided into four sub-sequences each having a size of 15. The CNN layer reads sub-sequence of size SS and number of features given as SC. In the next step, max-pool layer summarizes the extracted features from each sub-sequence and then they are flattened on a single layer. Finally, an LSTM layer is added to read the output from the previous layer and extracts its own features in the form of temporal dynamics before mapping the activities. Finally, a fully connected layer is added for classification and a soft-max layer for probability computation. The CNN-LSTM model is shown in Fig. 3. Two variants of this model are created namely CNN-LSTM1 (having $K = 64$, and $U = 50$) and CNN-LSTM2 (having $K = 128$ and $U = 100$). Both models use the following same parameters with $F = 3$, $S = 1$, $NS = 4$, $SS = {}^{WS}/_{NS}$, $DL = 100$, and $NA = 11$.

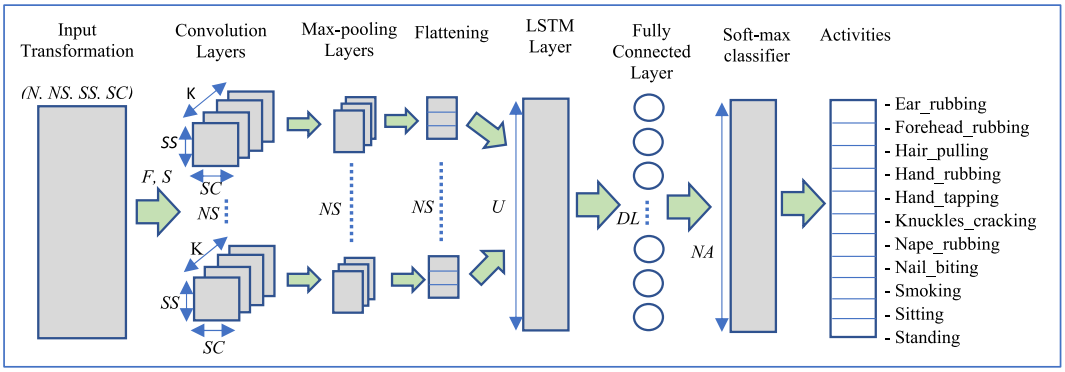

Fig. 3. CNN-LSTM model.

4.5. Non-deep learning models

Along with various deep learning models, ensemble models have also been created and evaluated for anxiety-displaying behaviors. They are a class of predictive algorithms in machine learning that create two or more related but different models to improve the overall performance of predictions. Two types of ensemble models Random Forest (RF) and Gradient Boosting (GB) are created to study their predictive power on our dataset. The RF algorithm takes a parameter *NE*, the number of estimators or the number of decision trees to be created within the ensemble model. Three RF models are created with a different value of *NE* of 10, 50 and 100, and named as RF1Est_10, RF2Est_50, and RF3Est_100, respectively. To implement a GB model, certain parameters are needed to be defined as well such as number of estimators (*NE*) and learning rate (*LR*). Two GB models are created having *NE* of 50 and 100 respectively. Both models are evaluated with *LR* of 0.01 and 0.05 and hence named as GB1Est50_LR0.01, GB1Est50_LR0.05, GB2Est100_LR0.01, and GB2Est100_LR0.05. It was determined that *NE* greater than 100 was taking a great deal of time without any significant improvement in the performance.

5. Experiments and results

The experiments are conducted to evaluate the models through different design setups, such as, window size, percentage of overlap between consecutive windows, type of sensors and position of sensors. At first, the impact of varying the size of sliding window and percentage of overlap on performance are analyzed for each model. Next the impact of type of sensors and position of sensors are studied on these types of behaviors.

HAR is a time series problem where the data is retrieved from triaxial motion with adequate window size [32,43,44]. As already explained in previous sections the smaller window sizes perform faster in terms of activity recognition and require lesser resources such as energy and time. Performance deterioration is observed if larger window size is used for simple activities or smaller window size is used for complex activities. Since our data has a mixture of both type of activities, we conducted experiments with various values of *WS* to find the best time slice to capture the pattern of an activity. The sample rate is the rate of number of sensors readings in one second by a sensor device. In this study, the sample rate of 50 Hz is used, which means a sensor reading is recorded after every 0.02 s. The *WS* is kept in such a way that is completely divisible by 4 as at later stages, it is divided into 4 sub-sequences for hybrid models. We conducted experiments by varying *WS* in the range 60–600 and for each *WS* the performance of the model was evaluated.

Initially, the percentages of overlap between consecutive windows are kept at 0%, 25%, 50% and 75%. Preliminary results show poor performances when the overlap is kept below 75%. More experiments with 70%, 80%, and 90% are conducted in which 90% gave the best performance in almost all the cases. The dataset has a total of 709,583 (*NR*) records for eleven activities from which training data equals to 648,857 (nine subjects) and test data equals to 60,725 (one subject). After splitting into training and test data, the data is transformed as given in Eq. (1). For example, if $OP = 90\%, WS = 60$, and $NR = 648,857$ (training), the corresponding N_{train} is 108,133 and for $NR = 60,725$ (test), the corresponding N_{test} is 10,111. After the conversion by *WS*, the dataset is reframed according to the requirement of each model as given in Table 4.

The initial experiments are given in Table 5 in which *WS* is kept from 60 to 600 number of tuples and *OP* is kept 90%. The combination of *A* and *G* of wrist position are used, which is most widely used and recommended by other researchers [13,15,34,45]. The best accuracy of each model is indicated in bold and italics, and the associated *WS* is given in the last column.

Majority of the models gave their best performance when the window size was above 300. Fewer models have given their best shot at window sizes of 60 and 100 as well but their accuracies were very low such as LSTM1 and LSTM2, and hence not significant. From this table, it can be inferred that the preferable window size for this dataset should be 360

Table 5
Test data accuracy (%) for different models where the percentage of overlap is 90%, sensors are *A* and *G* on wrist position, and evaluation is performed by hold-one-out method in which user having ID 3 is kept for testing and the rest for training. The last column represents the most suitable *WS* for which the best accuracy is achieved.

	WS (No. of tuples)												Best
Models	60	100	160	200	260	300	360	400	460	500	560	600	*WS*
CNN1	69.41	64.38	69.47	66.48	69.45	62.7	***70.01***	67.59	67.16	66.73	58.78	60.6	360
CNN2	70.81	73.92	74.57	73.92	***79.32***	72.7	70.2	73.49	69.55	77.7	78.47	73.07	260
CNN3	72.68	75.93	74.57	69.33	***79.72***	68.7	68.55	74.56	73.17	68.79	71.12	68.1	260
LSTM1	***53.08***	48.01	50.6	47.05	42.8	44.7	34.5	35.11	33.42	30.58	29.8	36.87	60
LSTM2	***67.13***	61.95	53.2	54.42	37.69	49.9	47.69	35.89	33.99	31.92	31.33	33.11	60
LSTM3	61.83	***65.22***	58.88	56.1	58.59	39.9	44.51	28.5	37.61	61.06	30.1	31.57	100
CNN-LSTM1	69.99	74.89	71.34	73.99	72.01	69.5	***77.49***	66.6	73.17	71.76	67.96	73.62	360
CNN-LSTM2	69.09	67.78	67.31	70.73	70.17	70.5	70.51	68.66	70.12	68.35	72.14	***73.51***	600
ConvLSTM	66.95	67.26	64.92	63.57	65.95	66	***69.82***	66.31	68.15	66.82	68.78	62.58	360
DNN1	60.99	67.36	65.49	***70.29***	62.9	67	64.68	63.89	66.91	60.43	56.94	54.64	200
DNN2	65.11	64.2	***66.55***	65.49	63.66	65	60.43	63.11	64.53	60.52	54.49	52.76	160
DNN3	***67.23***	64.27	64.22	60.9	62.49	61.8	63.35	55.08	62.55	56.83	61.33	54.42	60
GB1Est50_LR0.01	74.49	74.28	73.16	75.05	73.01	75.1	75.02	72.78	75.88	76.62	***77.14***	61.04	560
GB1Est50_LR0.05	73.39	74.28	74.08	74.98	73.77	75.7	75.78	73.49	***79.26***	76.53	78.27	68.87	460
GB2Est100_LR0.01	74.51	75.24	74.63	75.26	74.04	76.9	76.73	74.63	76.3	***78.33***	77.96	69.65	500
GB2Est100_LR0.05	74.43	75.67	75.5	76.42	75.21	77.1	76.6	75.05	79.67	77.7	***79.8***	75.83	560
RF1Est_10	62.83	62.09	61.74	62.41	62.28	62.7	63.92	64.61	61.98	63.13	61.43	***77.37***	600
RF2Est_50	67.41	67.13	67.26	69.09	68.68	69.9	68.86	68.51	68.89	68.79	67.86	***78.37***	600
RF3Est_100	68.79	68.52	69.11	70.12	69.54	69.7	70.13	70.29	71.44	70.41	69.18	***77.7***	600

Table 6
Test data accuracies of GB and RF models to investigate the trend with the increase in window size.

Model	660	700	760	800
GB1Est50_LR0.01	76.84	76.15	76.05	74.89
GB1Est50_LR0.05	76.84	77.06	77.92	73.51
GB2Est100_LR0.01	78.19	77.59	77.2	76.88
GB2Est100_LR0.05	78.43	78.11	79.65	76.72
RF1Est_10	58.7	59.9	58.87	59.57
RF2Est_50	68.14	68.55	66.52	68.3
RF3Est_100	68.75	68.81	67.68	70.75

or above. A drop in performance is also seen with the increase in window sizes, such as CNN and CNN-LSTM models. Whereas, for gradient boost (GB) and random forest (RF), mostly the best performances are obtained when the window is either 560 or 600. To investigate the most preferable window size for GB and RF, some more experiments were conducted for these models and are given in Table 6. Here, the window sizes are kept from 660 to 800 readings and a downward trend is noticed in accuracy with the increase in window size. Hence, subsequent experiments use the range from 200 to 600 in all models.

At this stage, the performances of the models are classified into three levels i.e., best performance (having accuracy equal and above 75%), mediocre performance (having accuracy range from 70% to 74%) and poor performance (having accuracy below 70%) are summarized below.

- **Higher Performance Models** CNN2, CNN3, CNN-LSTM1, GB1Est50_LR0.01, GB1Est50_LR0.05, GB1Est100_LR0.01, GB1Est100_LR0.05, RF1Est_10, RF1Est_50, and RF1Est_100.
- **Mediocre Performance Models:** CNN1, CNN-LSTM2, and DNN1.
- **Poor Performance Models:** LSTM1, LSTM2, LSTM3, ConvLSTM, DNN2 and DNN3.

The poor performance models were not capable enough to learn the complexity and dynamic nature of this dataset, as there is a great deal of variation involved in the data of each activity. Hence, from now on these models will not be included in our analysis except for ConvLSTM, which has been further analyzed due to its peculiar nature, which is best suited to learn the time series. As far as the other models are concerned, their performances are comparable with each other. Some performed better with smaller window size such as CNN2 and CNN3, and some performed better with larger window size such as GB2. It is evident that the largest accuracy is 79.8 and is achieved by GB2 at window size of 560. On the other hand, CNN2 and CNN3 are also very close with the accuracy of 79.32 at a lower window size of 260. Whereas CNN1, DNN1 and CNN-LSTM2 have shown mediocre performances for all window sizes.

As already mentioned, orientation of devices was not fixed. The reason is that in everyday life, people do not care much about the orientation of the devices, and they carry it in different ways. This has brought greater variation in our data. In the literature, it has been reported that the use of magnitude of a sensor would help in the dataset where the orientation of the devices is not fixed [37]. The magnitude is calculated by the formula given in Eq. (2) where S_x, S_y, and

Table 7
Best accuracies on test data by different combination of sensors, where the left side represents the performance without using magnitude column and right side represents the performance with magnitude column.

Sensor	Without magnitude			With magnitude		
combination	Model	Window	Accuracy (%)	Model	Window	Accuracy (%)
WAGM_	CNN-LSTM2	300	79.83	CNN-LSTM1	560	85.71
WA_PA	CNN2	560	64.9	CNN-LSTM1	500	68.62
WA_PG	CNN3	360	77.84	CNN-LSTM1	560	75.2
WA_PAG	CNN-LSTM2	400	69.07	CNN2	560	63.47
WAG_PA	CNN3	600	79.47	CNN-LSTM2	500	84.89
WAG_PG	GB2Est100_LR0.01	360	77.84	GB2Est100_LR0.05	500	82.46
WAG_PAG	CNN-LSTM2	560	75.71	CNN-LSTM2	560	78.98
WAGM_PAGM	CNN-LSTM2	560	72.04	CNN-LSTM1	600	80.68
WAGM_PA	CNN-LSTM2	500	79.23	CNN-LSTM1	**600**	**92.16**
WAGM_PG	ConvLSTM	400	80.61	CNN2	600	85.54
WAGM_PM	CNN2	260	81.25	CNN-LSTM1	460	82.55
WAGM_PAG	CNN-LSTM2	300	79.42	CNN-LSTM1	560	88.9
WAGM_PAM	CNN-LSTM1	300	86.92	CNN3	600	76.71
WAGM_PGM	ConvLSTM	600	84.66	CNN2	460	82.4

S_z correspond to three dimensions of each sensor. By adding the magnitude, the dimension of each sensor increases by one and now each sensor has one more dimension comprising of the magnitude of its three dimensions.

$$MagS = \sqrt{\left(S_x^2 + S_y^2 + S_z^2\right)} \tag{2}$$

We have data of three types of sensors i.e., *A*, *G* and *M* for two body positions, wrist, and pocket. There are studies where the researchers have used only *A* for activity recognition [33,34] and they achieved comparable performances. On the other hand, there are some researches where *A*, *G* and *M* are used in combination to achieve better results [13]. The decision of the number and type of sensors depends on the complexity of the dataset. If the dataset is not complex and the activities are simple, then a good performance may be achieved just by using *A*. On the contrary, if the dataset is dynamic in nature and is for complex activities, then the use of just *A* may not be sufficient for good performance. This suggests that other combination of sensors and other body position should also be considered as well. A notation has been used to represent the combination and position of the sensors used in our experiments. For this purpose, *W* is used to represent wrist position and *P* is used for pocket position and the sensors are *A*, *G* and *M*. For example, *WAG_PG* denotes that *A* and *G* of wrist position and *G* of pocket position. The experiments are run by each of the combination that can be of any meaning to investigate and find the most suitable combination of sensors for our study. Accelerometer is the most important sensor and wrist is the most important position to recognize the activities of our dataset since they mainly based on hand movements. Considering these two facts, the combinations are made in such a way where the accelerometer of wrist position is always included in each of the combination. The results are given in Table 7, where the best performing model for each combination is given with their respective window size and accuracies.

In Table 7, the performance of each sensors' combination has increased in almost all cases by adding a dimension of magnitude. It is observed that the sensor combinations where *A*, *G* and *M* of wrist position are used with one or two sensors of pocket position, such as WAGM_PA and WAGM_PAG, have shown better performances. Whereas the combination where all the sensors of wrist and pocket are used, such as WAGM_PAGM, did not give the best performance. The maximum accuracy achieved by this combination is 80.68%. One would expect that the above combination should have given better performance as all the sensors are used however our results do not support this. This may be due to the inclusion of more sensors increasing the complexity of the model. Since the number of samples to train the model are not increasing, hence, increase in the input variables does not improve the accuracy.

The CNN-LSTM model has generally outperformed other models in the recognition of these activities. The best accuracy achieved is 92.16% by the sensor combination of WAGM_PA by model CNN-LSTM1 at window size of 600 when used with magnitude. A graph of Confidence Interval (*CI*) of accuracy against the *WS* is plotted to determine the most suitable window size and is given in Fig. 4. For this analysis, CNN-LSTM1 was run five times for each *WS*, so that the range of *CI* could be obtained. The value of α is kept 0.05, which means 95% *CI* is calculated for the sample size of 5. The average and *CI* are plotted against *WS*, where *WS* is kept from 200 (4 s) till 1000 (20 s). The graph shows that, there is no significant increase in the average performance of the models when *WS* increases beyond 600. Furthermore, the *CI* range at $WS = 600$, is narrower suggesting a lower margin of error for this *WS*. The graph has a peak point at $WS = 600$, having almost the same accuracy (around 90%). On the contrary, the graph *CI* is quite overlapping, giving an overlapping performance range. This indicates that the most suitable *WS* for this dataset cannot be given by a specific value. The experiments may in the future be expanded and repeated for better investigation.

To analyze the accuracies at the activity level, the confusion matrix is created by the highest performance giving model and settings, i.e., CNN-LSTM1 (with 92.16% accuracy) shown in Fig. 5. The confusion matrix $\boldsymbol{C}$ is a metric in which a cell $\boldsymbol{c_{i,j}}$ represents the number of observations belonging to class $\boldsymbol{i}$ but assigned to class $\boldsymbol{j}$. The confusion matrix gives a more detailed view of performance compared to just a number such as accuracy. The diagonal elements in the confusion matrix

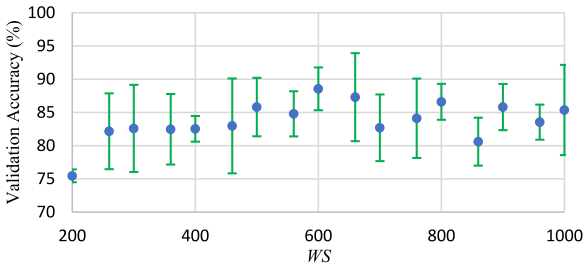

Fig. 4. Confidence interval plot of test accuracy of CNN-LSTM1, for $WS = 200$ till $WS = 1000$.

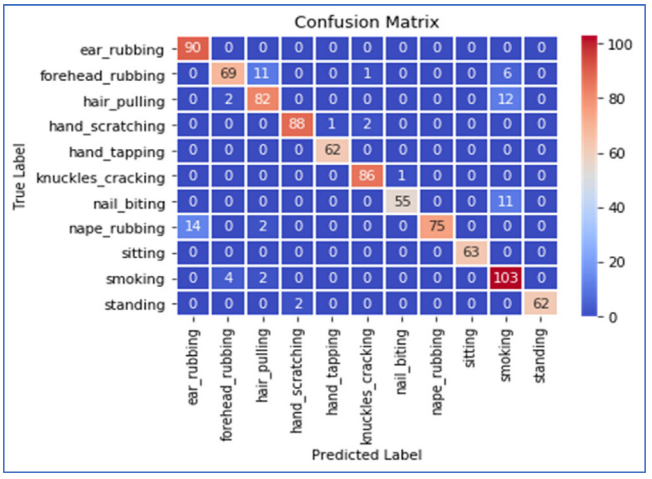

Fig. 5. Confusion matrix of CNN-LSTM1 where the sensor combination is WAGM_PA, window size is 600 and magnitude dimension is used. The test accuracy of this model and settings is 92.16%.

show the correct class recognition, and non-diagonal values are the errors in activity recognition. In the confusion matrix all the instances of the activities of *ear_rubbing*, *hand_tapping*, and *sitting* are accurately recognized by the model and there are zero errors for these activities. The activity of *knuckles_cracking* is also almost perfectly recognized except for just one instance where it is confused with the activity of *nail_biting* by the model.

The model misclassified *nape_rubbing* as *ear_rubbing* 14 times, *hair_pulling* as *smoking* 12 times, *forehead_rubbing* as *hair_pulling* 11 times and *nail_biting* as smoking 11 times. These misclassifications are understandable since the activities are very similar in nature. For example, *nape_rubbing* and *ear_rubbing* involve the similar physical movements around very close areas i.e., ear and nape. Similarly, *nail_biting* and *smoking* involve very similar hand movements around the mouth area. By using the confusion matrix, we can better understand the performance of our recognition model and can highlight the areas where we need the improvements in our model.

6. Discussion and conclusion

This research work focused on the recognition of certain behavior-markers pertaining to anxiety using body-worn sensors and deep learning techniques. This recognition may aid in the analysis, diagnosis, treatment, and progress monitoring of psychological disorders such as AD. To accomplish our objective, a novel dataset was created comprising of some typical anxiety-displaying behaviors using commonly found motion sensors.

An algorithm consisting of various deep learning techniques was evaluated to recognize the anxiety-displaying behaviors. The deep algorithms created in this research were deep ANN, CNN, LSTM, and CNN-LSTM where each algorithm had three different architectures. The deep learning-based models were compared against the famous ensemble models of RF and GB techniques. As the complexity of input data increases with the increase in complexity of design settings (number of sensors, number of sensor channels, window size etc.), RF and conventional deep learning techniques such as deep ANN, LSTM and CNN, did not give prominent performance. More complex model of CNN-LSTM gave the best performance. The maximum performance of test accuracy was achieved by GB, CNN and CNN-LSTM were 82.46%, 85.54% and 92.16% respectively. It can be easily seen that the accuracy of CNN-LSTM outperformed all other algorithms due to its capability of learning the features and temporal dynamics from long sequences.

Limitations & Future Work

Although our algorithm could recognize the activities successfully, however several limitations exist such as its applicability to real-world situations. In real-world data, there are various other activities that are not included in our dataset limiting the scope of this study. To improve this, more activities may be included in the dataset related to the general routine activities of human beings, such as, Activities of Daily Living (ADL). Another limitation is the size of the dataset that is limited to ten subjects. For deep learning models to perform better larger datasets are needed. Furthermore, in this study we trained the model on raw data and did not use statistical features. In the future we plan to extend our dataset with more subjects and ADL activities and to experiment with the statistical features as well.

CRediT authorship contribution statement

Nida Saddaf Khan: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Investigation. **Muhammad Sayeed Ghani:** Conceptualization, Validation, Formal analysis, Writing – review & editing. **Gulnaz Anjum:** Resources, Supervision, Provision of behavioral and ethical protocols, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] H.-B. Zhang, et al., A comprehensive survey of vision-based human action recognition methods, Sensors 19 (5) (2019) 1005, http://dx.doi.org/10.3390/s19051005.
[2] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, Z. Li, A review on human activity recognition using vision-based method, J. Healthc. Eng. (2017) https://www.hindawi.com/journals/jhe/2017/3090343/. (Accessed 10 July 2019).
[3] U. Alrazzak, B. Alhalabi, A survey on human activity recognition using accelerometer sensor, in: 2019 Joint 8th International Conference on Informatics, Electronics & Vision, ICIEV and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition, IcIVPR, Spokane, WA, USA, 2019, pp. 152–159. https://doi.org/10.1109/ICIEV.2019.8858578.
[4] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, J. Gama, Human activity recognition using inertial sensors in a smartphone: An overview, Sensors 19 (14) (2019) 3213, http://dx.doi.org/10.3390/s19143213.
[5] H. Allende, C. Moraga, R. Salas, Artificial neural networks in time series forecasting: A comparative analysis, Kybernetika 38 (6) (2002) 685–707.
[6] P. Lara-Benítez, M. Carranza-García, J.C. Riquelme, An experimental review on deep learning architectures for time series forecasting, Int. J. Neural Syst. 31 (03) (2021) 2130001, http://dx.doi.org/10.1142/S0129065721300011.
[7] N.S. Khan, S. Ghani, S. Haider, Real-time analysis of a sensor's data for automated decision making in an IoT-based smart home, Sensors 18 (6) (2018) http://dx.doi.org/10.3390/s18061711.
[8] N.S. Khan, M.S. Ghani, A survey of deep learning based models for human activity recognition, Wirel. Pers. Commun. (2021) 1–43, http://dx.doi.org/10.1007/s11277-021-08525-w.
[9] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, T. Kechadi, Human activity recognition with convolutional neural networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, vol. 11053, 2019, pp. 541–552.
[10] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, A. Aliverti, A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone, Sensors 19 (17) (2019) 3731, http://dx.doi.org/10.3390/s19173731.
[11] W. Jiang, Z. Yin, Human activity recognition using wearable sensors by deep convolutional neural networks, in: Proceedings of the 23rd ACM international conference on Multimedia, 2015, pp. 1307–1310, http://dx.doi.org/10.1145/2733373.2806333.
[12] D. Arifoglu, A. Bouchachia, Activity recognition and abnormal behaviour detection with recurrent neural networks, Procedia Comput. Sci. 110 (2017) 86–93, http://dx.doi.org/10.1016/j.procs.2017.06.121.
[13] A. Murad, J.-Y. Pyun, Deep recurrent neural networks for human activity recognition, Sensors 17 (11) (2017) 2556, http://dx.doi.org/10.3390/s17112556.
[14] N.Y. Hammerla, S. Halloran, T. Plotz, Deep, convolutional, and recurrent models for human activity recognition using wearables, in: International Joint Conference on Artificial Intelligence, IJCAI, New York, USA, 2016.
[15] S. Yao, S. Hu, Y. Zhao, A. Zhang, T. Abdelzaher, DeepSense: A unified deep learning framework for time-series mobile sensing data processing, in: WWW '17: 26th International Conference on World Wide Web, Perth, Australia, 2017, pp. 351–360.
[16] F.J. Ordóñez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) (2016) 115, http://dx.doi.org/10.3390/s16010115.
[17] J.M. Cisler, B.O. Olatunji, Emotion regulation and anxiety disorders, Curr. Psychiatry Rep. 14 (3) (2012) 182–187, http://dx.doi.org/10.1007/s11920-012-0262-2.
[18] A. Bystritsky, S.S. Khalsa, M.E. Cameron, J. Schiffman, Current diagnosis and treatment of anxiety disorders, Pharmacol. Ther. 38 (1) (2013) 30–57.
[19] C. Otte, et al., Major depressive disorder, Nat. Rev. Dis. Prim. 2 (2016) 16065, http://dx.doi.org/10.1038/nrdp.2016.65.
[20] DSMV, Diagnostic and Statistical Manual of Mental Disorders, fifth ed., American Psychiatric Association, 2013.

[21] E. Garcia-Ceja, M. Riegler, T. Nordgreen, P. Jakobsen, K.J. Oedegaard, J. Tørresen, Mental health monitoring with multimodal sensing and machine learning: A survey, Pervasive Mobile Comput. 51 (2018) 1–26, http://dx.doi.org/10.1016/j.pmcj.2018.09.003.
[22] J.T. O'Brien, et al., A study of wrist-worn activity measurement as a potential real-world biomarker for late-life depression, Psychol. Med. 47 (1) (2017) 93–102, http://dx.doi.org/10.1017/S0033291716002166.
[23] F. Wahle, T. Kowatsch, E. Fleisch, M. Rufer, S. Weidt, Mobile sensing and support for people with depression: A pilot trial in the wild, JMIR mHealth uHealth 4 (3) (2016) http://dx.doi.org/10.2196/mhealth.5960.
[24] T. Jeong, D. Klabjan, J. Starren, Predictive analytics using smartphone sensors for depressive episodes, 2016, ArXiv160307692 Cs Stat, Available online: http://arxiv.org/abs/1603.07692. (Accessed 11 March 2019).
[25] R.S. McGinnis, et al., Rapid detection of internalizing diagnosis in young children enabled by wearable sensors and machine learning, PLoS One 14 (1) (2019) e0210267, http://dx.doi.org/10.1371/journal.pone.0210267.
[26] M. Elgendi, C. Menon, Assessing anxiety disorders using wearable devices: Challenges and future directions, Brain Sci. 9 (3) (2019) 50, http://dx.doi.org/10.3390/brainsci9030050.
[27] K. Rennert, E. Karapanos, Faceit: Supporting reflection upon social anxiety events with lifelogging, in: CHI '13 Extended Abstracts on Human Factors in Computing Systems, CHI EA '13, New York, NY, USA, 2013, pp. 457–462. http://dx.doi.org/10.1145/2468356.2468437.
[28] D. Miranda, M. Calderón, J. Favela, Anxiety detection using wearable monitoring, in: Proceedings of the 5th Mexican Conference on Human-Computer Interaction, New York, NY, USA, 2014, pp. 34:34-34:41. https://doi.org/10.1145/2676690.2676694.
[29] Muhammad Shoaib, J. Scholten, P.J.M Havinga, Towards physical activity recognition using smartphone sensors, in: 10th IEEE International Conference on Ubiquitous Intelligence and Computing, UIC 2013, IEEE, Italy, 2013, pp. 80–87, https://www.utwente.nl/en/eemcs/ps/research/dataset/.
[30] N.S. Khan, M.S. Ghani, G. Anjum, Adam-sense: Anxiety-displaying activities recognition by motion sensors, 2021, http://dx.doi.org/10.17632/6g6pxwjj48.1, Mendeley Data. Elsevier Available online: https://data.mendeley.com/datasets/6g6pxwjj48/draft?a=cd0c1648-3ee8-4d69-8210-c51ae57aced9.
[31] M. Vafaeipour, O. Rahbari, M.A. Rosen, F. Fazelpour, P. Ansarirad, Application of sliding window technique for prediction of wind velocity time series, Int. J. Energy Environ. Eng. 5 (2) (2014) 105, http://dx.doi.org/10.1007/s40095-014-0105-5.
[32] N. Twomey, et al., A comprehensive study of activity recognition using accelerometers, Informatics 5 (2) (2018) 27, http://dx.doi.org/10.3390/informatics5020027.
[33] M. Zeng, et al., Convolutional neural networks for human activity recognition using mobile sensors, in: 6th International Conference on Mobile Computing, Applications and Services, Austin, United States, 2014. http://dx.doi.org/10.4108/icst.mobicase.2014.257786.
[34] A. Ignatov, Real-time human activity recognition from accelerometer data using convolutional neural networks, Appl. Soft Comput. 62 (2018) 915–922, http://dx.doi.org/10.1016/j.asoc.2017.09.027.
[35] M.O. Gani, A.K. Saha, G.M.T. Ahsan, S.I. Ahamed, A novel framework to recognize complex human activity, 2017, Available online: /paper/A-Novel-Framework-to-Recognize-Complex-Human-Gani-Saha/1024a5b27c233a452bbca412c322943ae02b9a7c. (Accessed 7 November 2018).
[36] M. Janidarmian, et al., A comprehensive analysis on wearable acceleration sensors in human activity recognition, Sensors 17 (3) (2017) 529, http://dx.doi.org/10.3390/s17030529.
[37] M. Shoaib, et al., Complex human activity recognition using smartphone and wrist-worn motion sensors, Sensors 16 (4) (2016) 426, http://dx.doi.org/10.3390/s16040426.
[38] J.T. Hancock, T.M. Khoshgoftaar, Survey on categorical data for neural networks, J. Big Data 7 (1) (2020) http://dx.doi.org/10.1186/s40537-020-00305-w.
[39] J. Donahue, et al., Long-term recurrent convolutional networks for visual recognition and description, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Boston, MA, USA, 2015, pp. 2625–2634.
[40] T.N. Sainath, O. Vinyals, A. Senior, H. Sak, Convolutional, long short-term memory, fully connected deep neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2015, pp. 4580–4584, http://dx.doi.org/10.1109/ICASSP.2015.7178838.
[41] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: The IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 3156–3164.
[42] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.K. Wong, W. WOO, Convolutional lstm network: a machine learning approach for precipitation nowcasting, in: The 28th International Conference on Neural Information Processing Systems, NIPS'15, Cambridge, MA, USA, vol. 1, 2015, pp. 802–810.
[43] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) (2014) 6474–6499, http://dx.doi.org/10.3390/s140406474.
[44] J. Ortiz Laguna, A.G. Olaya, D. Borrajo, A dynamic sliding window approach for activity recognition, in: International Conference on User Modeling, Adaptation, and Personalization, vol. 6787, Berlin, Heidelberg, 2011, pp. 219–230. https://doi.org/10.1007/978-3-642-22362-4_19.
[45] T. Zebin, N. Peek, A. Casson, M. Sperrin, Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, vol. 2018, Honolulu, HI, 2018, pp. 1–4. https://doi.org/10.1109/EMBC.2018.8513115.

Nida Saddaf Khan received her B.S. degree in computer science and MBA (finance) degree from University of Karachi, Pakistan, in 2007 and 2010, respectively, and the MS degree in computer science from Institute of Business Administration (IBA), Karachi, Pakistan, in 2013. She is currently a Ph.D. student and has been teaching in the department of Computer Science, IBA, Karachi since 2014. She has also served as research assistant in Artificial Intelligence Lab at IBA, Karachi for 2 years. She has also taught as a visiting faculty in the Department of Computer Science, University of Karachi from 2008 to 2013. Her research interests include sensor data analytics, deep learning, machine learning, data mining, probabilistic reasoning and computational intelligence.