CHARM-Deep: Continuous Human Activity Recognition Model Based on Deep Neural Network using IMU Sensors of Smartwatch

Article *in* IEEE Sensors Journal · April 2020

DOI: 10.1109/JSEN.2020.2985374

CITATIONS
30

READS
624

3 authors:

Sara Ashry Mohammed
Egypt-Japan University of Science and Technology
11 PUBLICATIONS **100** CITATIONS
SEE PROFILE

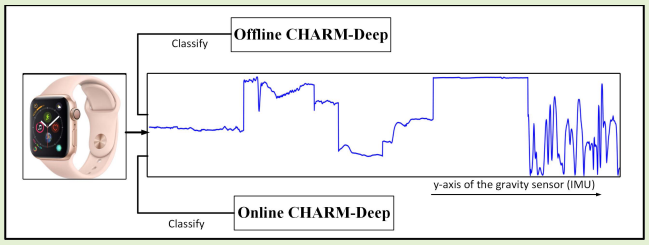
Tetsuji Ogawa
Waseda University
133 PUBLICATIONS **816** CITATIONS
SEE PROFILE

Walid Gomaa
Egypt-Japan University of Science and Technology
163 PUBLICATIONS **1,242** CITATIONS
SEE PROFILE

CHARM-Deep: Continuous Human Activity Recognition Model Based on Deep Neural Network Using IMU Sensors of Smartwatch

Sara Ashry, Tetsuji Ogawa, *Member, IEEE*, and Walid Gomaa, *Member, IEEE*

***Abstract*—In the present paper, an attempt was made to achieve high-performance continuous human activity recognition (CHAR) using deep neural networks. The present study focuses on recognizing different activities in a continuous stream, which means 'back-to-back' consecutive set of activities, from only inertial measurement unit (IMU) sensors mounted on smartwatches. For that purpose, a new dataset called 'CHAR-SW', which includes numerous streams of daily activities, was collected using smartwatches, and feature representations and network architectures were designed. Experimental comparisons using our own dataset and public datasets (Aruba and Tulum) have been performed. They demonstrated that cascading bidirectional long short-term memory (Bi-LSTM) with featured data performed well in offline mode from the viewpoints of accuracy, computational time, and storage space required. The input to the Bi-LSTM is a descriptor which composed of a stream of the following features: autocorrelation, median, entropy, and instantaneous frequency. Additionally, a novel technique to operate the CHAR system online was introduced and shown to be effective. Experimental results can be summarized as: the offline CHARM-Deep enhanced the accuracy compared with using raw data or the existing approaches, and it reduced the processing time by 86% at least relative to the time consumed in executing the Bi-LSTM classifier directly on the raw data. It also reduced storage space by approximately 97.77% compared with using raw data. The online evaluation shows that it can recognize activities in real-time with an accuracy of 91%.**



***Index Terms*—CHAR, Smartwatch, IMU, Bi-LSTM, autocorrelation, entropy, and instantaneous frequency.**

I. Introduction

MONITORING the activities of daily living (ADL) [1] has garnered significant attention owing to the rapid propagation and cost reduction of sensing hardware, which resulted in the astounding outbreak of activity data [2]. In addition to the ubiquity and sensors being more and more embedded into commodity wearable and mobile devices. The motivation of such research is the high demand for various applications such as using in smart cities, in-home monitoring of elderly people's health [3], promoting patient contribution in the management of the chronic diseases, enabling patients and doctors to obtain insights about the progression and the illnesses impact and assessment of therapy. Additionally, it is applicable for different purposes like evaluating sports performance, or in-car monitoring for safe driving.

Initially, vision-based sensing, using cameras, has been the focus of research studies and more recently inertial sensing, using motion-based sensors like inertial measurement unit (IMU) that can be attached to the user's body or embedded in wearable and mobile devices, such as smartphones, has been investigated [4].

For recognizing ADLs, two categories of a data acquisition system have been used: the surrounding fixed-sensor based system and the wearable mobile-sensor based system.

In the former, data are collected from distributed ambient sensors such as microphones, cameras, and motion sensors that are attached to fixed locations in the activity environment (e.g., walls, cupboard doors, microwave ovens, and

Manuscript received February 14, 2020; revised March 27, 2020; accepted March 28, 2020. Date of publication April 3, 2020; date of current version July 6, 2020. This work was supported by the Information Technology Industry Development Agency (ITIDA), Information Technology Academia Collaboration (ITAC) Program, Egypt-Grant Number (PRP2019.R26.1 - A Robust Wearable Activity Recognition System based on IMU Signals). The associate editor coordinating the review of this article and approving it for publication was Dr. Emiliano Schena. *(Corresponding author: Sara Ashry.)*

Sara Ashry and Walid Gomaa are with the Computer Science and Engineering (CSE) Department, Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt (e-mail: sara.ashry@ejust.edu.eg; walid.gomaa@ejust.edu.eg).

Tetsuji Ogawa is with the School of Science and Technology, Waseda University, Shinjuku-ku 169-8050, Japan (e-mail: ogawa@pcl.cs.waseda.ac.jp).

Digital Object Identifier 10.1109/JSEN.2020.2985374

water taps [2], [5], [6]). These sensors, however, have many restrictions owing to their fixed nature, and user activities are not detectable if the user goes outside the space where these sensors are installed. Additionally, from the perspective of privacy, it is neither acceptable nor convenient to monitor people continuously using cameras in his/her private rooms. And even for acceptable places for monitoring, these are susceptible to security breaks.

In the latter approach, it depends on wearable mobile-sensors that can be worn on different body parts such as a wrist, leg, waist, and chest. Wearable inertial measurement unit (IMU) sensors have been generally used in recognizing ADLs as they are small, cheap, and almost embeddable in all recent smartphones, watches, shoes, sensory gloves, and hand straps. In particular, smartwatches are practical for recording data without incurring any uncomfortable feeling for a user during sleeping, working, and swimming. They are also more robust to security and privacy constraints. Therefore, wearable-sensor-based activity recognition is a key feature of many ubiquitous computing applications.

The problem statement of the continuous human activity recognition can be described in both modes: offline and online, each one has its own purposes and applications. For instance, if we are interested in following a person's daily routine or classifying specific scenarios, sensors can collect continuous data that can be uploaded to a server at a later time and can be processed offline for the classification process as well as making a diverse set of analytical purposes [7]. However, in an online monitoring application like a fitness trainer, the system can monitor the user with a specific fitness program contains a set of activities in a certain duration and sequence. Wherefore, we may be interested in what the user is currently doing and if he follows the fitness program efficiently [8]. Another example of the online system is that there might be some interest in collecting information from the user while he is 'walking' in a particular part of a city (GPS is also crucial in this case). Also, online recognition of activities becomes significant in health care applications; for example, the system can send an ambulance to the patient with a heart attack. It also helps doctors know more about the patients' past and current activities as well as continuous knowledge of the patients' physiological parameters.

Several studies have been conducted regarding human activity recognition (HAR) using wearable sensors in smart environments, such as CASAS [9] and PlaceLab [10]. Moreover, new wearable sensor systems have been prototyped, human activity datasets have been constructed, and HAR systems have been developed using mostly machine learning techniques. It is noteworthy that many previous studies have focused on recognizing isolated human activities [11]–[13], and [14]. However, datasets related to a continuous stream of human activities such as the 'Aruba' and 'Tulum', which were recorded using fixed sensors at CASAS smart homes, are still limited [15]. By contrast, the present work focuses on recognizing the continuous stream of human activities keeping transitions among every two consecutive activities. We recorded the activities streams using smartwatches and organized them as a dataset called 'CHAR-SW'.

According to the IDTechEx report [16]: wearable technology products are flourishing, with a total market worth over $50 billion in 2019, having more than doubled in size since just 2014. This historic growth includes smartwatches and other wearables. The market is expected to continue growing at a similar stride in the coming year [17]. That's why; we investigated using a smartwatch for recognizing human activities for the motivation purposes mentioned above. The contributions of this paper can be summarized as follows:

- A 'CHAR-SW' dataset for continuous human activities recognition is collected by only IMU sensors embedded in Apple smartwatch series four and made freely available for researchers.
- A high-efficient 'CHARM-Deep' system is designed for offline and online purposes.
- We have developed novel hand-crafted features as descriptors for the CHARM-Deep to enhance the system performance. The main contribution of these features is to save the need for an extensive training set. It also helps in increasing accuracy and reducing the execution time and storage space.
- The offline CHARM-Deep system is proposed with multiple configurations and network architectures. Its effectiveness is demonstrated using our collected CHAR-SW dataset and two other public datasets, surpassing the state-of-the-art.
- Additionally, a novel approach for the online CHARM-Deep system is introduced. It can accurately detect and classify actions as early as possible on the fly.
- Finally, the CHAR-SW dataset, together with the discussion and knowledge gained from this study, would facilitate the research development in the emerging field of continuous human-activity recognition.

The remainder of the paper is organized as follows: Section II describes relevant studies regarding HAR methods and the advantages of our proposed system. The background techniques used in the present study are explained in Section III. Section IV presents an overview of the CHAR-SW dataset. Sections V and VI describe the proposed offline and online CHARM-Deep systems, respectively. Comparisons between the proposed systems and state-of-the-art methods are provided in Section VII while Section VIII gives a brief analysis and discussion about the knowledge gained from this study. Finally, the output of the paper is concluded in Section IX.

II. Related Work

This section briefly provides related work regarding HAR systems and describes the advantages of the developed CHARM-Deep system. The literature review was conducted according to: a) data acquisition techniques, b) major existing datasets, c) offline techniques for analyzing continuous human activities and existing segmentation approaches, d) online human activity recognition and e) existing feature extraction methods for pattern recognition techniques.

A. Data Acquisition

1) Camera-Based Techniques: Most of the state-of-the-art systems for classifying *streams* of activities use vision-based models [18], in which actions in a video stream are detected in both offline and online [19].

By contrast, the drawback of using a fixed camera with a limited observable area is solved by using wearable sensors, especially smartwatches as in our novel dataset. The proposed dataset with full description is represented in sectionIV where it used only inertial measurement unit (IMU) sensors embedded in the smartwatch to collect streams of activities. The users wear the smartwatch only on their right wrist. Additionally, building a large dataset in realistic conditions without any supervision is an influential factor. The most prominent advantage of smartwatches is that the user can move freely and no uncomfortable feeling is generated when the device is used to record and analyze the data.

2) Sensor-Based Techniques: In designing any sensor-based activity-recognition system, sensors' numbers and locations are critical parameters. Regarding sensor location, body parts from the feet to the chest have been selected. The selected locations were chosen according to the particular activities under study. For example, ambulation activities such as walking and running were detected using a waist sensor. Meanwhile, non-ambulation activities such as brushing teeth and eating were classified using a wrist sensor [20]. Most systems require obtrusive sensors on the throat, chest, wrist, thigh, and ankle connected via wired links, which restricts human movement. In healthcare applications involving older people or patients with heart disease, obtrusive or fixed sensors are not ideal [14]. Furthermore, some datasets have been collected under controlled conditions, and they were used to classify a small number of isolated activities. Researchers [21] and [14] have used smartwatches to classify separately recorded activities.

By contrast, our proposed dataset "CHAR-SW" collects continuous streams of different activities using only smartwatches and then each full stream is classified effectively.

B. Major Existing Datasets

Regarding continuous human activities recognition datasets, we selected the Aruba and Tulum real-world datasets which are the closest possible to the study on which our work is based. These datasets are produced from the Washington State University CASAS smart-home project [9], which were collected using passive infrared (PIR) sensor and door sensors.

Data gathered from the Aruba dataset was taken from 31 motion sensors, three-door sensors, five temperature sensors, and three light sensors. Eleven activities were performed for 7 months. Regarding Tulum dataset, data were collected from 18 motion sensors and two temperature sensor. Ten activities were performed for 4 months by two residents, as indicated in Table I and II. These data are all represented as a sequence of time-stamped sensor data. For example, participants were asked to perform a sequence of activities such as washing dishes, meal preparation, and eating. The study done on those datasets was verified in [22].

TABLE I
ARUBA AND TULUM DATASETS METADATA [9]

Dataset	Gender	Sensors Number	Time Interval
Aruba	Elderly/female	42	7 months
Tulum	2 married residents	20	4 months

TABLE II
ACTIVITIES AND STATISTICS OF ARUBA AND TULUM DATASETS [15]. EVENT NUMBER MEANS THE NUMBER OF OCCURRENCE OF THE ACTIVITY

Aruba dataset		**Tulum dataset**	
Activity	Events Number.	Activity	Events Number.
Bed_to_Toilet	1330	Cook_Breakfast	11343
Eating	16153	Cook_Lunch	5350
Enter_Home	2018	Enter_Home	11998
Housekeeping	10583	Group_Meeting	23787
Leave_Home	1922	Leave_Home	1200
Meal_Preparation	293334	R1_Eat_Breakfast	10395
Relax	72717	R1_Snack	216178
Resperate	542	R2_Eat_Breakfast	12312
Sleeping	32682	Wash_Dishes	24392
Wash_Dishes	10464	Watch_TV	50280
Work	16321	----------	------

C. Existing Segmentation Approaches for Offline Continuous Human Activity Data

For classifications of continuous-activities, methods such as SVM, KNN, CB-KNN, MKENN, MKRENN [22], and seq-seq LSTM [23] have been used. To detect activities in a continuous stream, some segmentation approaches such as activity-based windowing, time-based windowing, and sensor-based windowing [22] have been proposed. However, each exhibits some drawbacks. In activity-based windowing, the activities are generally not well distinguished, thus resulting in imprecise activity boundaries; meanwhile, in time-based widowing, streaming data are divided into fixed-time windows.

The window length (WL) is a selective tuning parameter; if WL is small, the window may contain insufficient information for decision making. Conversely, if the length is long, the information of multiple activities can be embedded in one window. Consequently, the activity that dominates the frame will be represented more compared with other activities, which affects recognition accuracy negatively. According to sensor-based windowing, each window contains the same number of sensor events [22]. A major drawback of this method is that the window may contain sensor events that are widely separated in time, and it may contain sensor events of more than one resident as in the Tulum dataset.

In CHARM-Deep, we used different techniques, particularly, seq-seq BiLSTM-based models that are not time-consuming, to accurately detect activities in a stream, as illustrated in section V.

D. Online Human Activity Recognition

To the best of our knowledge, there is a lack of research studies that addressed the online classification capabilities [24]. The authors in [25] studied online human activity recognition using only the accelerometer sensor of the Android phone. Their objective was the classification of fundamental movements of a user, such as walking, running,

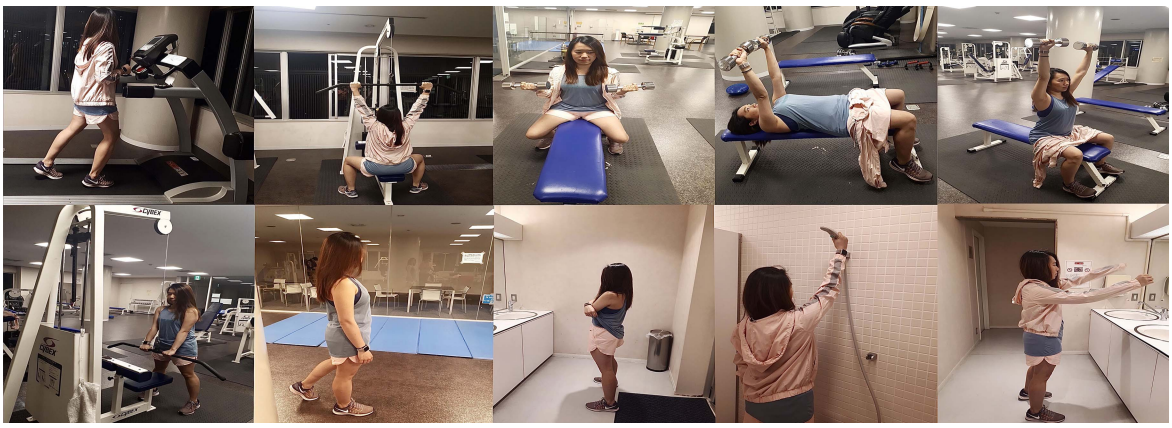


Fig. 1. Snapshots of data collection.

sitting and standing. They compared the efficiency of using clustered KNN and Naïve Bayes. The clustered KNN combines the advantages of the minimum distance and KNN classifiers. The clustered KNN method exhibited much better performance than the Naïve Bayes in terms of accuracy on mobile platforms with limited resources. To the best of our knowledge, most of the related work focuses on classifying isolated activities using smartphones; our novelty contribution is classifying a continuous stream of numerous activities collected from smartwatches by using a deep online model.

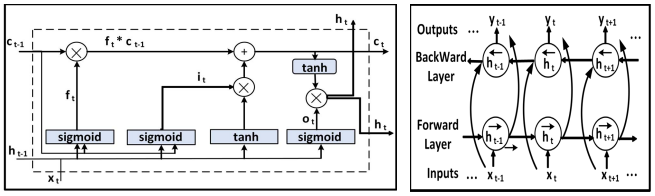
E. Existing Features for Pattern Recognition Techniques

Concerning the feature space, although methods that use raw data directly for training have been proposed, most researchers prefer to apply a preliminary feature extraction stage before the training phase [26]. Three main types of features can be extracted from the IMU signal data: (1) time-domain features such as mean, variance, and standard deviation [27], auto-correlation function [21], interquartile, and ranges; (2) frequency-domain features such as Fourier transform, discrete cosine transform, and wavelet transform [28]; and (3) dimensionality reduction techniques such as PCA and LDA [20]. Additionally, several methods can be used to extract features from a sequence of sensor events, such as the baseline method, sensor window mutual information (SWMI), sensor windows mutual information extension (SWMIex), and sensor windows last state (SWLS) [22].

III. Background

This section briefly reviews LSTM and Bi-LSTM, which are fundamental to the technique developed in this study.

A Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) that is specifically crafted for handling sequential data. They differ from traditional RNNs in the sensing mechanism because they include a flexible gating mechanism. In order to determine the degree to which the constituent recurrent units maintain the preceding state and incorporate the extracted information of the current input data. Furthermore, they include particular provisions to improve gradient flow over sequences and overcome the vanishing gradient



(a) LSTM cell (b) Bi-RNN

Fig. 2. Fig a is an LSTM cell and Fig b is Bi-RNN [31].

problems [29]. LSTMs (being RNNs) are trained using back propagation through time (BPTT) [30]. LSTM equations are explained in detail in [29]. The most prominent advantage of LSTMs is their ability to consider *contextual information* over varying timescales when mapping between input and output sequences through hidden layer units. Therefore, we apply LSTMs in the context of this work in order to differentiate activities based on sequential sensor readings.

A disadvantage of conventional RNNs (and their variants such as the standard LSTMs) is that they can only utilize previous context, while bidirectional RNNs (Bi-RNNs) circumvent this limitation by processing data in both directions with two separate hidden layers, which are then fed forward to the same output layer. As illustrated in Figure 2, a Bi-RNN computes the forward hidden sequence $\overrightarrow{h_t}$, the backward hidden sequence $\overleftarrow{h_t}$, and the output sequence y by considering the original input sequence, i.e., from $t = 1$ to T and the reversed input sequence, i.e., from $t = T$ to 1. Subsequently, the output layer is updated according to the following equations:

$$\overrightarrow{h_t} = H\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \quad (1)$$

$$\overleftarrow{h_t} = H\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \quad (2)$$

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h_t} + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (3)$$

In these equations, W denotes the weight matrices (e.g., W_{xh} is the input-hidden weight matrix), b indicates the bias vectors (e.g., b_h is the hidden bias vector), and H is the hidden layer (sigmoid function). Combining Bi-RNNs with LSTMs yields a bidirectional LSTM [31]. Hence, the cell state will be generated by calculating a weighted sum using both the previous cell

TABLE III
SCENARIOS (SCS) COLLECTED IN CHAR-SW DATASET WHERE THE NUMBER OF STREAMS IN SC 4, 8, 9 IS 40 STREAMS AND IN ALL OTHER SCENARIOS IS 50 STREAMS. SO, THE TOTAL NUMBER OF STREAMS IN THE DATASET IS 470

Sc	Actions Stream
Sc1	Get up, bed making, walk, washing hands, brushing teeth, comb hair, praying, eat Sandwich, pour water, drink.
Sc2	Put off, wear clothes, walk, descend stairs, walk, climb stairs, walk, sit down, using keyboard, writing on paper.
Sc3	Pray, walk, sit down, writing on paper, using keyboard, use Mobile, eat sandwich, drink, using keyboard, walk.
Sc4	Drive, walk, climb, put off, showering, wear clothes, cutting components, flipping, eat with fork/knife, eat with spoon.
Sc5	Stand up, walk, washing dishes, walk, sit down, pour water, drink, use Mobile, stand up, walk.
Sc6	Brush teeth, walk, sleeping, get up, shaking dust, bed making, sweeping, wiping, praying, sit down, use mobile.
Sc7	Put off, wear clothes, Walk, descend stairs, cycling, running, climb stairs, work out, rowing, drink.
Sc8	GYM treadmill, weight back, biceps, chest, shoulders, triceps, walk, put off, showering, wear clothes.
Sc9	walk, descend stairs, cycling, running, climb stairs, walk, put off clothes, wear clothes, washing hands, eat sandwich.
Sc10	Playing on piano, guitar, violin, drawing, dancing, sweeping, walk, sit down, reading, sleeping.

TABLE IV
STATISTICS OF SUBJECTS INCLUDED IN CHAR-SW DATASET. 56% OF SUBJECTS ARE FEMALES AND 44% ARE MALES

	Age	Height	Weight
Range	20-55	150-185	55-95
Mean	37	167.5	75
Std	24.75	24.7	28.28

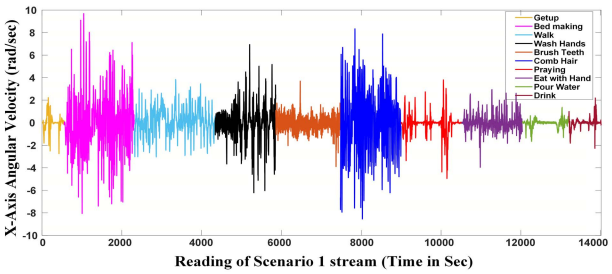


Fig. 3. Shows a visualization of the X-axial angular velocity of Scenario 1 activities while each color represents an activity.

state and the current information generated by the cell [29]. For many sequence modelling tasks, it is beneficial to have access to future and past contexts. However, standard LSTM networks process sequences temporally; they ignore the future context. Bidirectional LSTM networks extend unidirectional LSTM networks by introducing a second layer, where hidden-to-hidden connections flow in the opposite temporal order [32]. The paradigm can exploit information from both the past and the future. In this study, we used Bi-LSTM, as shown in Figure 4.

IV. CHAR-SW DATASET AND SMARTWATCH SENSORS

In developing the CHAR-SW dataset,[1] 25 participants (14 females and 11 males) volunteered to record human activities, as shown in Table III, by wearing an Apple Watch Series 4 on their right wrists. Figure 1[2] shows snapshots of the data collection process. Each participant performed the listed actions serially, as indicated in the scenarios listed in Table III. This dataset contains 470 streams. The statistics of the volunteers are listed in Table IV. The volunteers are from various countries (e.g., Egypt, Japan, Thailand, China, and Australia).

The Apple Watch Series 4 contains a 64-bit dual-core processor and weighs 30 grams. It comes with many sensors, including GPS and Inertial Measurement Unit (IMU) sensors (i.e., accelerometer and gyroscope). The accelerometer is designed to withstand forces up to $32G$. It also includes Wi-Fi and Bluetooth radios and is capable of operating off its inbuilt battery for up to one day. Besides, it is suitable for all skin types [33].

The IMU itself is designed to give full degrees of freedom [34]. The accelerometer and gyroscope sensors are considered to be low-level IMU sensors, permitting the extraction of other derived readings, e.g., rotational displacement [35] by fusing their readings together. Accelerometers sense linear motion/acceleration while gyroscopes sense angular velocity (rad/sec) to measure their rotational motion. The static gravitational component can be isolated from the accelerometer using the gyroscope, and the linear acceleration is obtained by removing the gravitational component from the accelerometer values [36]. The CHAR-SW dataset contains acceleration, angular velocity, rotation displacement and gravity readings - all of which are tri-axial - recorded at a sampling rate of 50 Hz. In [37], it was confirmed that the essential information could be located beyond 50 Hz. Figure 3 shows the x-axis raw stream of angular-velocity of Scenario 1 activities.

V. OFFLINE CHARM-DEEP SYSTEM

This section presents the details of the offline mode of the CHARM-Deep system and its constituent modules.

A. System Overview

Figure 4 shows the system architecture of CHARM-Deep in the offline mode, which consists of a data acquisition module,

[1]The CHAR-SW dataset can be downloaded by contacting the first author through the following link: https://drive.google.com/a/ejust.edu.eg/uc?id=1acGQZ3FUAbOdV8DiyJiiJLGKd7SCzaZE&export=download

[2]The subject gave permission for her photo to be published in this paper.

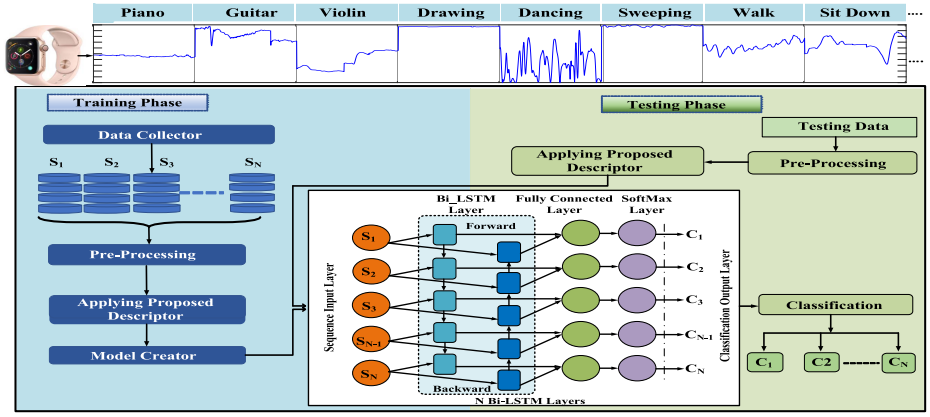


Fig. 4. Shows the offline CHARM-Deep system architecture. The stream signal from Scenario 10 represents the y-axis of the gravity sensor (unit: m/sec^2) to help in visualizing the framework.

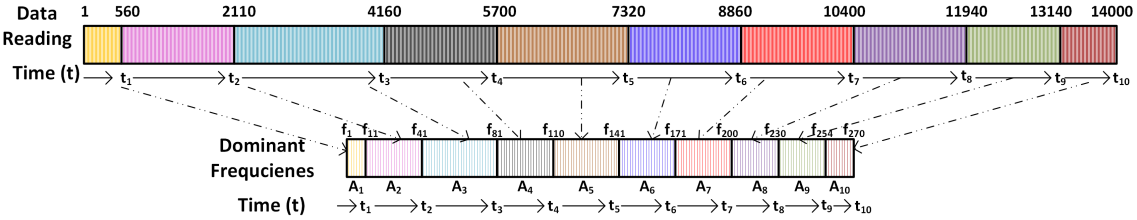


Fig. 5. It shows one of the raw signals as an example and how the new ground truth mapped after applying IFQ.

feature extraction module, model building module, and classification module.

The *data acquisition module* obtains four tri-axial signals from the IMU sensors, as mentioned in Section IV. Such signals are used as inputs to the subsequent feature extraction module, for extracting robust and discriminative features to recognize continuous human activities with high accuracy. During training, the model creator uses the extracted features to train a Bi-LSTM based model. The classification module is then used to estimate the probabilities of human activities classes, given data streams unseen during the training process.

B. Feature Extraction

Feature selection and extraction can enhance the performance (e.g., accuracy, computation time and storage space) of a learning system. We tested many features and descriptors, as shown in Table V, and we have chosen the one that gave the highest accuracy. IFQ-SAME is a novel hand-crafted descriptor which enhances the predictive performance, computation time, and data storage, as will be shown in later sections.

We mainly focused on three main features: autocorrelation coefficients, spectral entropy, and instantaneous frequency. Additionally, we tested a few combinations amongst them and other features to determine the most useful set. Autocorrelation can be represented mathematically as a degree of similarity between a given time series and a lagged version of itself over successive time intervals. It was chosen due to its effectiveness in filtering the data and removing noise [21]. Furthermore, we considered the spectral entropy of the signal, which takes the inherent nature of activity recognition signals into account. The entropy is used to capture the formants or the peakiness of a distribution [38]. For instance, the 'running' activity has a high spectral entropy as its signal is highly oscillatory; in contrast, the 'walking' activity has low spectral entropy.

Additionally, we also used the instantaneous frequency (IFQ) because it can accurately describe the frequency-variant nature of the signal under consideration, and it can define the location of the signal's spectral peak as it varies with time [39]. This physical property allows accurate discrimination of human activities because it is representative of the mono-component signals, where there is only one frequency or a narrow range of frequencies varying as a function of time while it breaks down the multi-component signals into different dominant frequencies [40]. The IFQ is estimated as the first conditional spectral moment of the time-frequency distribution of the input signal. It is calculated in [41] as in Eq.4 where $P(t, f)$ denotes the power spectrum:

$$f_{\text{inst}}(t) = \frac{\int_0^\infty f P(t, f)\ df}{\int_0^\infty P(t, f)\ df} \tag{4}$$

$$[ifq, t] = instfreq_{(function)}(data, f_S) \tag{5}$$

The inputs of the instantaneous frequency ($instfreq$) function in Eq.5 are the long stream of raw data, and the sampling rate ($f_S = 50$) and the outputs are a short vector from the signal's frequencies distribution (ifq) and another vector related to times (t) which correspond to these frequencies [41]. Figure 5 shows how the new ground truth is defined after applying the IFQ to streams of different activities. We matched the return t of the IFQ with the original raw data time according to its sampling rate. Subsequently, the new ground

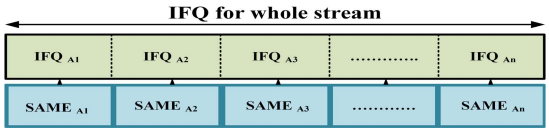


Fig. 6. It indicates the vertical concatenation of IFQ and SAME features while *An* represents $(Activity)_n$.

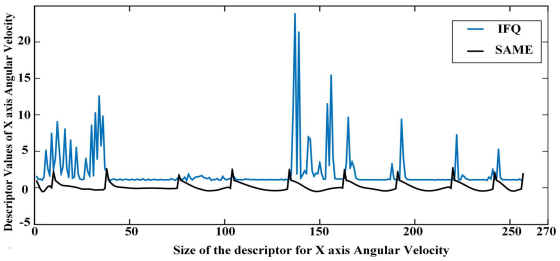


Fig. 7. IFQ-SAME descriptor of a stream of different activities in scenario 1 for the x-axis angular velocity. Blue signal shows the IFQ signal and black shows the SAME signal.

truth (GT) can be mapped from the original GT and redefined easily that was used to learn the activity boundaries.

Feeding the IFQ data and its corresponding GT, in the training phase, to a seq-seq long short term memory network [23] enables each activity to be detected and discriminated. After that, an autocorrelation function with a predefined lag is applied to the raw data. Then, the autocorrelation output is horizontally concatenated with the median and entropy for activities in the stream to form a vector called the 'stream autocorrelation median entropy' (SAME) descriptor. Finally, the IFQ and SAME vectors for each sensor signal are vertically concatenated to shape the final descriptor matrix, called 'IFQ-SAME', as shown in Figure 6.

For example, if the raw signal of the first stream has matrix dimensions of $r \times t$, where r is the number of all sensors axes ($r = 12$), and t has a variable-length depends on the long raw signals length recorded by volunteers. Then, after extracting the IFQ-SAME descriptor, the featured signal dimensions will be $x \times y$, where x is the new dimension of the featured sensors taking into consideration the vertical concatenated descriptor and y is the new variable shorter length of the featured signals after conversion. Using this descriptor, 97.77% of the storage space is reduced relative to the original raw data. Figure 7 shows the descriptor applied on the stream of different activities that contain useful information of each activity while Figure 3 shows the raw data of this stream.

C. Offline CHARM-Deep: Model Building Module

Figure 4 shows the network architecture of CHARM-Deep in offline mode while the internal mapping is many to many. IFQ-SAME features of continuous streams from predefined scenarios are extracted and used as inputs to the seq-seq Bi-LSTM layer, followed by a fully connected layer. Subsequently, the human activities probabilities are obtained in the output layer with a softmax activation function. The number of units in the output layer is set to N, which corresponds to the number of target human activities in the scenario.

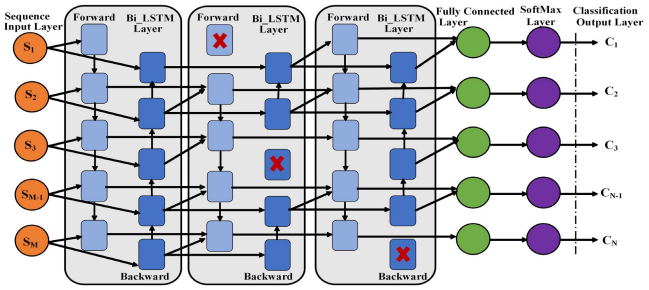


Fig. 8. Crossed-out neurons represent dropped-out.

To avoid overfitting, dropout [42] was applied to the Bi-LSTM layer. The network architecture (i.e., the number of layers, dropout percentage, and activation functions) was empirically determined, as described in Section VII and Appendix. The hyper-parameters (i.e., learning rate, batch size, and epochs) are also mentioned in the evaluation VII section.

Preventing Overfitting: To enhance the model's robustness and further reduce overfitting, CHARM-Deep employs dropout regularization [42], which has been shown to be useful for large networks. It is used to prevent neurons from developing co-dependencies between each other during training. In this technique, neurons are randomly removed from deep neural network layers with a dropout rate during the training phase at different epochs, as shown in Figure 8. Hence, the detached neurons don't contribute to the forward nor the back-propagation passes at the given epoch. The neural network samples different architectures during training, but these architectures share neuron weights. Therefore, the deep neural network is forced to learn more robust data representations that allow the network to generalize rather than memorize. As the dropout technique is parameterized by the global percentage of neurons to be omitted from the candidate network, we performed an experiment to demonstrate the effect of varying this percentage on the proposed method's performance. The results are shown in Appendix IX-A.2.

D. Classification

For classifying a new data stream, the system extracts the IFQ-SAME descriptor from the raw data signals and then feeds it to the CHARM-Deep model. The softmax function is used to output the probability of the feeding features to the network. More formally, assume that we have testing streams $\{x_i\}$, where $1 \leq i \leq N_{ts}$ and N_{ts} is the number of testing streams. Each whole stream of different activities contains features (f) with total number n as in $(x_{if_1}, x_{if_2}, \ldots, x_{if_n})$. The corresponding outcomes from the deep network for the input x_{if_n} are a_{ij}. The softmax function converts the score a_{ij}, where j is the classified activity in the stream corresponding to f_n, into a normalized probability score as follows:

$$P\left(a_{ij}\right) = \frac{e^{a_{ij}}}{\sum_{j=1}^{n} e^{a_{ij}}} \tag{6}$$

The probabilistic $P(a_{ij})$ for the initial outputs a_{ij} or a_N is denoted as $P = [P_1, P_2, ..., P_N]$, where a_N is the corresponding activity for each feature in the whole stream scenario.

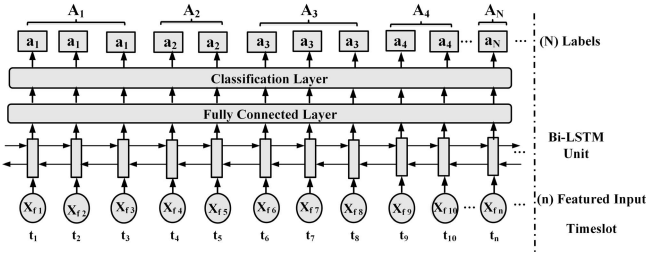


Fig. 9. Classification framework of CHARM-Deep. A_N represents the whole sequence of activity (N) recognized.

The goal is to obtain the N^{th} activity as output that has the maximum probability for each given input feature set (x_f) as indicated in this equation $a_N^* = argmax[P(a_N \mid x_f)]$. Finally, the classification output layer consists of a number of units corresponding to the number of activities N in the scenario stream. Its goal is gathering each sequence stream of the same activity outputs a_N as a final class (A_N) as visualized in Figure 9. We used cross-entropy as a loss function for optimizing a classification predictive model.

VI. Online CHARM-Deep Model

In the online mode: real-time data (i.e., 3D acceleration, angular velocity, rotation displacement, and gravity) - are collected by a smartwatch - synchronized with iPhone. Then, the data are sent from the iPhone to the PC server via WiFi. The proposed hierarchical model is manipulated on the server. It relies on two Bi-LSTM units called 'Online CHARM-Deep Model'. The First Bi-LSTM unit recognizes the data over small non-overlapped windows in real-time (Small Window Bi-LSTM). The second called (Large Window Bi-LSTM) to detect and recognize the data of the accumulated windows, as shown in Figure 10, while these windows are concatenated non-overlapping small windows as will explain later in detail.

The large window expands and shrinks based on a decision taken according to the probability of the detected patterns of both small and large windows with different thresholds as described in algorithm 1 and Figure 10. The threshold are ($T_1 = 0.9$, $T_2 = 0.8$, $T_3 = 0.7$) empirically estimated by the greedy algorithm [43]. Algorithm 1 shows the online CHARM-Deep technique while all the variables are described in footnote.[3]

During training: all individual or isolated activities are trained offline by applying the proposed descriptor IFQ-SAME for each activity. Then, the extracted featured data are fed to the Bi-LSTM network with dropout 80% and ELU as an activation function. However, during the testing and classification:

[3]Note: SW and LW refer to small and large window respectively. SW_{Feat}, LW_{Feat} refer to the featured signal after applying the descriptor to both the small and large windows respectively. 'Start' and 'EnD' are the start and end index of the raw data inside the window. $Start_L$, End_L, and $Pred_L$ are the start index, end index, predicted class of the featured signal inside the large window respectively. $MaxP_L$ is the maximum probability estimated for the different actions. The same meaning for variables with subscript 'S' instead of 'L' which is related to the small window. The 'Convert' function returns the original index of the raw data from the index of the featured data. The 'Next' function calls the next small window received online.

ALGORITHM 1 Online CHARM-Deep

Input: Raw data of the received window in real time
Output: Classification of the received window
switch *Online CHARM-Deep State* **do**
  case *Initialize*
    LW=SW=100;
    Start = 1;
    End =size(LW);
  end
  case *Predict and Cut*
    for $Start$ **to** End **do**
      $LW_{Feat} = Descriptor(LW)$;
      $SW_{Feat} = Descriptor(SW)$;
      $[Start_L, End_L, Pred_L, MaxP_L] = Predict(LW_{Feat})$;
      $[Start_S, End_S, Pred_S, MaxP_S] = Predict(SW_{Feat})$;
      if $MaxP_L >= T_1$ **then**
        $Start_L = End_L$; (Predicted and Cut)
      else if $MaxP_L <= T1$ **then**
        $Start_L = Start_L$; (Expand, add next SW)
        $End_L = End_L + size(SW_{Feat})$;
      else if $(Pred_L == Pred_S)\&(MaxP_L >= T_2)$ **then**
        $Start_L = End_L$; (Predicted and Cut)
      else if $(Pred_L != Pred_S)\&(MaxP_S >= T_3)\&(MaxP_L <= T_3)$ **then**
        $Start_L = End_L - SW$; (Shrink and Cut)
      $Start = Convert(Start_L)$;
      $End = Convert(End_L)$;
      $SW = Next(SW)$;
    end
  end
endsw

a predefined small window with size (100), a hyper-parameter tuning, has been chosen according to the greedy algorithm in [43]. Then, the IFQ-SAME descriptor is extracted from the small window data and classified using $Predict$ function as shown in algorithm 1. The input of this function is the featured data, and the outputs are (the start and end indices of the featured data, the predicted activity, the maximum probability).

Then, the server will receive the following small windows and will apply the same previous prediction step. A concurrent prediction step will be applied on an accumulated window (from the current and previous small window) to predict this whole new large pattern. Because the existence of a slightly complete pattern will increase its probability to match the right activity. Frequently, the algorithm compares the probability of the accumulated windows with a threshold of $T1$ until the condition is satisfied. Then, it cuts the large data window, removes it from the online stream and names it. Then, the algorithm resets the stream and receives new data. The mechanism of predicting every new small window is to detect the beginning pattern of another activity as early as possible. Thus, the small window can be excluded from the accumulated window. In this case, the prediction probability

TABLE V

ACCURACY OF DIFFERENT DESCRIPTORS APPLIED TO CHAR-SW (SC1), ARUBA, TULUM DATASETS USING EPOCHS = 200. THESE FEATURES FEED TO CHARM-DEEP WITH ONE BI-LSTM LAYER, WITHOUT ANY DROPOUT. THE MEASURED TIME OF TRAINING AND TESTING IS IN SECONDS. THE TOTAL ACCURACY IS CALCULATED AS THE AVERAGE OVER ALL ACTIVITIES ACCURACY IN THE SCENARIO

Datasets	Classification Accuracy and processing time using different descriptors											
	Autocorr		Pentropy		IFQ		IFQ+Pentropy		SAME		IFQ-SAME	
	Acc%	Time	Acc%	Time	Acc%	Time	Acc%	Time	Acc%	Time	Acc%	Time
CHAR-SW	81.5	23	91.8	29	92.5	32	93.2	38	92.7	35	95.4	42
Aruba	90.3	57	87.2	56	90	53	88.6	57	90.45	60	90.9	62
Tulum	88.2	53	85.8	50	89.9	52	87.3	53	90	55	91.5	58

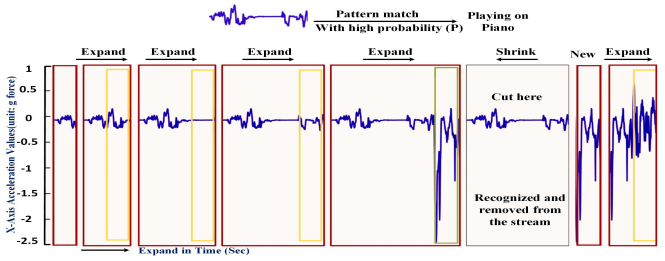


Fig. 10. Shows the online CHARM-Deep Model. Yellow rectangular represents the small window, and the expanding red rectangular is the large window. The green small rectangular describes detecting a new pattern of another activity different from the previous one.

of the large window - contains little different activity at the end - will decrease. This mechanism strengthens the idea that two different activities entered in the same window should be observed and detected.

The large window will shrink by cutting the last small window of the different activity from it. Then, the newly defined large window will be removed from the stream and labeled. Finally, the algorithm resets the stream and start receiving new data.

In this approach, we use a single Bi-LSTM rather than cascade Bi-LSTM for a faster execution time due to its importance in the real-time processing of the online mode.

VII. EVALUATION

In order to demonstrate the effectiveness of CHARM-Deep in both the offline and online modes, we performed several experiments to investigate the different aspects of the system. Sections VII-A, VII-B VII-C, and Appendix IX-A, IX-B are concerned with the offline mode of the system, while Section VII-D discusses the online mode. We consider three datasets of interest, CHAR-SW, Aruba and Tulum, in all evaluations, except when otherwise specified.

In the evaluations of the offline mode, we focus on selecting the most suitable feature representation and network architecture. Note that when one particular effect/parameter is studied, we fixed the other settings, as shown in section VII-A, VII-B, and Appendix IX-A. The offline mode of CHARM-Deep in different scenarios is investigated in section VII-C.1. Besides, it was compared with state-of-the-art HAR systems using two publicly-available datasets, Aruba and Tulum, to demonstrate its effectiveness as indicated in section VII-C.2. Finally, the online mode of CHARM-Deep is evaluated by comparing different descriptors with raw data and with other related techniques.

All the experiments were executed using a single NVIDIA GeForce RTX 2020Ti GPU. In the experimental models, the optimal hyper-parameters that are selected by a greedy algorithm [43] are as follows: the learning rate is 0.01; batch Size is 32; max epochs are 200; and hidden neurons numbers in the four cascade Bi-LSTM are 150, 100, 75, 75, respectively. Also, 70% of the data are randomly chosen for training while the remaining 30% is used for testing.

A. Different Descriptors Feed to Offline CHARM-Deep

Table V shows the accuracy obtained when using different descriptors with a Bi-LSTM network with a single hidden layer as a classifier. We consider the activities in Scenario 1 for the CHAR-SW dataset. For the Aruba and Tulum datasets, we consider all their respective activities. The IFQ-SAME descriptor can be seen to obtain the highest predictive accuracy across all the datasets: CHAR-SW 95.4%, Aruba 90.9%, and Tulum 91.5%, compared to all the other descriptors considered.

We also investigated the processing time by measuring the amount of time the training and testing processes take. We use inbuilt MATLAB commands for measuring these times. We obtain a reduction in the processing time of 86.45% relative to the time taken for using the Bi-LSTM classifier directly on the raw data. It is also noteworthy to state that the use of the IFQ-SAME descriptor obtains savings in the amount of storage needed during network training by up to 97.77%, compared to the space required by the raw data itself. The average size of the whole raw stream is 1045.2 Kb. In contrast, the featured stream is 23.28 Kb. This shows the importance of using effective features rather than using raw data, and in particular, highlights the benefit of our proposed descriptor.

Additional evaluation metrics are done on CHAR-SW dataset (Sc1) as indicated in Appendix IX-B. It includes the confusion matrix and the following metrics (i.e., precision, recall, specificity, and F-score).

B. Effect of Network Architectures on Accuracy

1) Effect of Changing the Number of Layers Without Dropout: We also explored the effect of adding more Bi-LSTM layers on the approach and presented the results in Table VI. The table includes the effect of layers without dropout and with the recommended activation function (as mentioned in

TABLE VI
ACCURACY RESULTS FROM ADDING BI-LSTM LAYERS TO THE NETWORK USING IFQ-SAME DESCRIPTOR AND WITHOUT DROPOUT USING THE RECOMMENDED ACTIVATION FUNCTION IN APPENDIX IX-A.1

Total No. Layers	**5**	**6**	**7**	**8**
No. of Bi-LSTM Layers	**1**	**2**	**3**	**4**
CHAR-SW Accuracy%	95.4	94.7	96	92.2
Aruba Accuracy%	90.9	93.34	93.46	93.3
Tulum Accuracy%	91.5	91.61	93.79	93.15

TABLE VII
OFFLINE COMPARISON WITH DIFFERENT CHARM-DEEP NETWORK ARCHITECTURES USING DROPOUT AND ELU AS THE ACTIVATION FUNCTION FOR CHAR-SW DATASET

Models	**Accuracy**	
	Raw Data	Featured (IFQ-SAME)
Single LSTM	79.8%	89.71%
Single BiLSTM	83.3%	95.8%
Cascade (2 LSTM)	84 %	91.29 %
Cascade (2 BiLSTM)	86.2%	94.8%
Cascade (3 LSTM)	87.67%	94.3 %
Cascade (3 BiLSTM)	88%	96.2%

Appendix IX-A.1). It can be seen that increasing the number of layers affects the accuracy, reaching the best value (on all the considered datasets) at three hidden Bi-LSTM layers (total number is seven layers with input, fully connected, softmax, and classification layers). By increasing the number of layers beyond this value, the accuracy starts to decrease as the network begins to overfit the input data. We attribute the differences in results for each model to the fact that the models and the data were subjected to stochastic noise.

2) Comparison With Different Offline CHARM-Deep Models With Dropout: Experiments in Table VII shows the comparative performance of different offline CHARM-Deep models using the CHAR-SW dataset. The effectiveness of using both Bi-LSTM and LSTM, in addition to their cascading versions were investigated using raw and featured data (IFQ-SAME). The hyper-parameters were empirically chosen, i.e., the dropout rate of 80% and ELU as an activation function, as described in Appendix IX-A. We observed that using three hidden Bi-LSTM layers with empirically chosen dropout yielded the highest accuracy; thus, this is the recommended version for the offline mode of the CHARM-Deep architecture. This study also shows that the average accuracy enhancement between the featured (IFQ-SAME) column and the raw data column is about 8.87% while, the average accuracy increasing by using Bi-LSTM rather than LSTM is approximately 2.9%.

C. Comparative Evaluation

1) Evaluating Other Scenarios: Table VIII shows the accuracy of ten other scenarios of the CHARM-SW dataset using our proposed Bi-LSTM based approach. The Scenarios details are mentioned in Table III. The experimental results shown in Table VIII were obtained by selecting the best parameters as determined from the previous experiments: a batch size of 32, 200 training epochs, three hidden Bi-LSTM layers, and a dropout ratio of 80%. It is clear that the highest accuracy of 97.2% is achieved in Scenario 8, whereas the lowest accuracy of 94.2% (which is still reasonably high) is produced at Scenario 4.

2) Comparing With State-of-the-Art Techniques and Evaluating Other Public Datasets: We also perform comparative evaluations between the proposed technique and other state-of-the-art techniques as applied to the Aruba and Tulum datasets (described in II-B). Figure 11 shows how the accuracy of CHARM-Deep compares to the other techniques (mentioned in [22]). As shown, the total accuracy enhancement obtained through the use of CHARM-Deep with the proposed descriptor relative to the other approaches ranges from 7% to 9%, in addition to reducing the execution time of the training and testing processes by approximately 80%. The combination of the best tuning parameter and network architecture used in the CHARM-Deep model afforded the highest accuracy. In the Aruba dataset, the best model was achieved using a dropout percentage of 60% and ELU as the activation function, with three Bi-LSTM layers. On the other hand, the best model for the Tulum dataset used a dropout percentage of 60%, a leaky-Relu as the activation function, and three Bi-LSTM layers.

D. Online CHARM-Deep Evaluation

We evaluate our novel technique of online CHARM-Deep VI by comparing the predictive performance of the proposed method with different classifiers against using the raw data. The average accuracy is calculated by comparing all the final predicted and removed activity from the stream with the ground truth. Besides the classification accuracy, we also evaluated the performance of the online mode of CHARM-deep in terms of execution times, i.e., how long it takes to recognize activity in the testing phase as indicated in Table IX. Ideally, for online scenarios, the recognition process needs to happen as early as possible (i.e., on the fly) so as to provide as smooth a user experience as possible.

It can be observed that the proposed method with IFQ-SAME achieves the highest accuracy of 91%. It exceeds the performance of other descriptors by 4% to 21% and the raw data by 6%. It is notably, from Table IX that the average prediction time of the classified window with IFQ-SAME is higher than the average prediction time using the IFQ descriptor by 39%. Conversely, it is also lower than the use of raw data by 39.5%. It can be concluded that IFQ-SAME is a good descriptor for the online prediction as it achieved the highest accuracy. However, it takes 1 second extra from the fastest descriptor with lower accuracy. We are of the opinion that this value is an acceptable trade-off for the accuracy obtained using the proposed method.

The online mode of CHARM-Deep is also compared with state-of-the-art techniques for online activity recognition, as reported in [25]: Naive Bayes, KNN, CNN, and clustered KNN as verified in figure 12. The online CHARM-deep using the IFQ-SAME descriptor outperforms the existing approaches by a percentage ranged from 3.5% to 11% when we applied the same windowing technique at Section VI. The accuracy for online CHARM-deep, Clustered KNN, CNN, KNN and Naive Bayes with the IFQ-SAME descriptor is 91%, 87.2%, 86%, 85% and 80%, respectively. In contrast, the accuracy

TABLE VIII

ACCURACY OF THE RECOMMENDED VERSION OF OFFLINE CHARM-DEEP (THREE BI-LSTM LAYERS, DROPOUT 80%, ELU AS ACTIVATION FUNCTION) USING IFQ-SAME DESCRIPTOR FOR DIFFERENT SCENARIOS IN CHAR-SW DATASET

Scenarios (SC)	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
Accuracy%	96.2	95.8	96.8	94.2	97	94.3	95.2	97.2	95.7	96.4

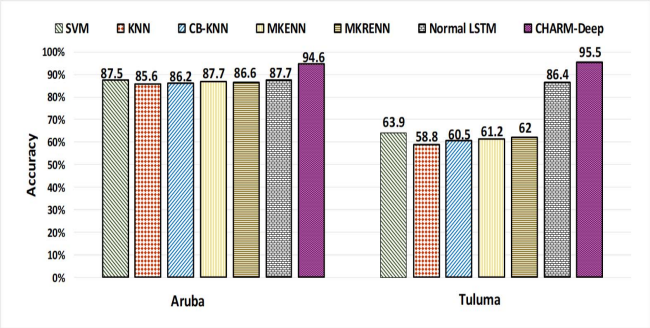


Fig. 11. Accuracy of the offline CHARM-Deep comparing with the state-of-the-art techniques which are applied for Aruba and Tulum datasets. Using SWLS feature extraction for SVM, KNN, CB-KNN, MKENN, and MKRENN approaches verified in [22].

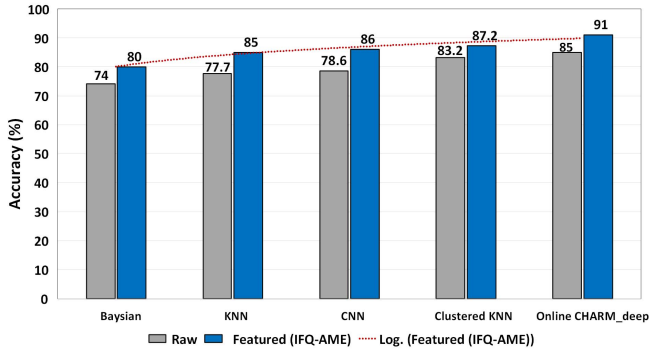


Fig. 12. Accuracy of the online CHARM-Deep comparing with the existing techniques like Baysian, KNN, CNN, and clustered KNN [25]. Accuracy is calculated by averaging over the accuracy's of the predicted and removed activity from the stream.

TABLE IX

THE EXPERIMENTAL RESULTS OF ONLINE CHARM-DEEP IN TERM OF AVERAGE ACCURACY AND AVERAGE EXECUTION TIMES IN SECOND OF RECOGNIZING ACTIVITY AS EARLY AS POSSIBLE IN THE FLY ACCORDING TO THE LARGE SEVERED WINDOW

Features	Raw	Autocorr	AME	Pentropy	IFQ	IFQ-SAME
Acc%	85	70	82	74	87	91
Time	4.3	2.4	2.56	3.53	1.87	2.6

obtained using the same classifiers (in the same order as above) as applied to raw data are 85%, 83.2%, 78.6%, 77.7% and 74%, respectively. These results indicate the supremacy of the proposed approach for online activity recognition relative to the other extant approaches considered.

VIII. DISCUSSION AND ANALYSIS

In this section, we summarize conclusions obtained from the results and their subsequent discussion in the preceding subsections.

Considering the offline CHARM-Deep for continuous human activity recognition, experimentally, we observed that using a combination of the best tuning parameters and the network architecture (i.e., three hidden Bi-LSTM layers, an efficient dropout, a specific activation function) besides using IFQ-SAME descriptor afforded the highest accuracy for the datasets in the study.

It can be inferred that using effective handcrafted features can enhance system performance while also minimizing the need for a large training set. It also increases the accuracy and reduces the execution time and storage space. In the future work, we will try to adjust an automatic setting of the dropout and activation function parameters to work fine enough for all different datasets but these parameters are specified in this study which gave the highest accuracy for every dataset.

More discussion details about the CHAR-SW dataset through the offline CHARM-Deep model: we measured the effect of using features on the processing time. We evaluated that the execution time of training and testing, using IFQ-SAME descriptor and the recommended version of three cascaded Bi-LSTM layers on single GPU is around 1.2 minutes while the time of using raw data is 10 minutes. So, the featured model executes faster than the raw model by 88%. Generally, using GPU instead of CPU has effectiveness in the processing time. For instance, the time of using raw data on CPU is about 455 minutes (7.6 hour) while the time of using raw data on GPU is 10 minutes. We conclude that the raw model on GPU is faster than the raw model on CPU by 97.8% or by 45.5 times. Around 90% saving in the processing time has achieved when comparing the featured model between CPU and GPU.

Appendix IX-B draws the confusion matrix of scenario 1 as an example. It also represents the analysis of scenario 1, which indicates that bed-making activity has achieved the highest classification accuracy in the stream. Followed by washing hands, drink from a glass and get up from the bed. It also indicates that "pour water" activity has the lowest accuracy, and it is misclassified with "drink from a glass" as both of them have a similar repetitive pattern.

Figure 11 shows the comparison of experimenting with Aruba and Tulum datasets using the offline CHARM-Deep and the state-of-the-art techniques like SVM, KNN, CB-KNN, MKENN and MKRENN that are verified in [22]. From the Tulum dataset, we conclude that the MkRENN classifier achieves higher accuracy than all KNN classifiers. Also, it can even perform better than SVM. What can be gained mostly from using different variants of KNN is that the classification performance is close or better than the ones obtained by SVM without going through the learning phase.

The SVM in the training phase becomes unfeasible when the size of the available data is large. However, as there

isn't a model to build at KNN and its variants, the test phase depends on the size of the training set and on the capacity of the algorithm to determine the neighbors quickly without having to go through the entire dataset. It is a costly phase in terms of time and memory in case of a huge dataset. A multi-classes Exemplar-based Nearest Neighbors (MkRENN) classifier was proposed in [22] to overcome the high computational cost of the SVM training phase without the need to learn a model for each activity. The offline CHARM-Deep with IFQ-SAME descriptor outperforms all the previous studies for both Aruba and Tulum datasets in terms of accuracy, execution time, and storage space.

In the online CHARM-Deep, we chose, at the network architecture, a single Bi-LSTM layer rather than cascaded Bi-LSTM to make the execution time faster. The processing time is a significant factor, especially for the online mode. It is evident that using the proposed method with IFQ-SAME has achieved the highest accuracy of 91% and the average prediction time of the classified window is 2.6 second.

We observed that the error rates reduced by 4.2% to 12.1% if we comparing the featured data of online CHARM-deep with other approaches like Clustered KNN, CNN, KNN and Naïve Bayes. Naïve Bayes is highly dependent on the system parameters. Compared to Naïve Bayes, on average, clustered KNN achieved a much better classification performance, around 87.2% accuracy with K parameter equals 3.

IX. Conclusion and Future Work

We first collected the CHAR-SW dataset followed by the CHARM-Deep models for offline and online modes, which are deep neural networks approach for recognizing a continuous stream of human activities collected by streaming IMU signals from smartwatches. We presented the details of the system and demonstrated its applicability, by preprocessing the raw data before feeding it to the proposed Bi-LSTM architecture. Furthermore, we investigated effective regularization techniques to increase the system robustness and reduce overfitting.

Upon implementing our offline system, an accuracy of 94.2% to 97.2% was achieved in classifying the entire stream of different activities in all scenarios. This accuracy was better than those of other LSTM-based techniques [23] with raw data by 14% at the least per each scenario. In addition, offline CHARM-Deep with featured data reduced the processing time of training and testing by 86% compared with the normal LSTM applied to raw data, based on the same installation and tuning parameters. Finally, we compared the offline CHARM-Deep system with other techniques using two public datasets that demonstrated the degree of enhanced accuracy.

In order to provide an automated monitoring system for different human needs, an online system that performs activity recognition from free movement device like the smartwatch is required. Most of the techniques used in the literature are not suitable enough to build high efficient online system. In this paper, we proposed and evaluated the online CHARM-Deep on CHAR-SW dataset, which outperforms the state-of-the-art techniques with accuracy reaching to 91%.

In the future, we plan to collect a dataset of daily human-human interactions including handshakes and scrimmages, as well as that of sports activities such as boxing, by collecting data using multiple smartwatches.

TABLE X
Effect of Changing Activation Function on Accuracy for CHAR-SW, Aruba, and Tulum Dataset Using the Recommended Version of 3 Bi-LSTM Layers Without Dropout

Activation function	tanh	relu	Leaky-relu	clipped-relu	Elu
CHAR-SW Acc%	90.2	91.83	94.68	95	96
Aruba Acc%	90.4	92.33	91.71	93.09	93.46
Tulum Acc%	90.1	93.12	93.79	91.43	90.0

TABLE XI
The Effect of Changing Dropout Percentage for CHAR-SW, Aruba, Tulum Datasets Using the Recommended Version of Three Bi-LSTM Layer and Activation Function

Dropout %	Accuracy		
	CHAR-SW%	Aruba%	Tulum%
0	96	93.4	93.79
10	93.5	91	93.43
20	95	92.46	91.8
30	94.66	91.26	94.31
40	94.58	94.4	93.35
50	95.71	94.14	94.25
60	94.38	94.6	95.5
70	95.58	92.04	94.02
80	96.2	90.17	94.31
90	95.56	86.26	85.94

Appendix

A. Effect of Changing Network Architecture

1) Effect of Changing Activation Function on Accuracy: Table X shows the effect of changing the network's activation function on the predictive accuracy of the proposed method based on the three Bi-LSTM layers (the recommended version). The activation function decides whether (and how) the neuron output should be fired/activated. The table indicates that the ELU (i.e., Exponential Linear Unit) activation function achieved the highest accuracy on the CHAR-SW dataset, with an accuracy of 96.2%. An accuracy of 93.46% is obtained for the Aruba dataset. However, on the Tulum dataset, it is obvious that the Leaky-relu (Leaky Rectified Linear Unit) activation function [44] yields the highest accuracy at 93.79%. In our dataset, we recommend using an exponential linear unit (ELU) [45] as an activation function, which is a generalization of the ReLU [46] that uses a parameterized exponential function. This helps to mitigate the vanishing gradient problem [47].

2) Effect of Changing Dropout Percentage: We assessed changing the network architecture on the predictive performance of the proposed method based on the three Bi-LSTM layers. Table XI shows the effect of increasing the dropout percentage on performance. Regarding the CHAR-SW dataset, it shows that the best value is achieved at 80% dropout, where the accuracy is 96.2%. The best accuracy is 94.6%, 95.5% at dropout percentage 60% for the Aruba and Tulum datasets, respectively. This is due to the fact that as the network

TABLE XII
CONFUSION MATRIX OF SCENARIO 1 ACTIVITIES WHILE *Ci* IS THE ORDER OF ACTIVITIES IN THIS SCENARIO

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	154	3	0	0	0	0	0	0	0	0
C2	0	506	2	0	0	0	0	0	0	0
C3	0	5	601	10	0	0	0	0	0	0
C4	0	0	0	448	5	0	0	0	0	0
C5	0	0	0	10	433	4	0	0	0	0
C6	0	0	0	0	3	446	14	0	2	0
C7	0	0	0	0	0	5	417	30	0	0
C8	0	0	0	0	0	0	2	442	10	0
C9	0	0	0	0	0	0	0	2	162	22
C10	0	0	0	0	0	0	0	0	2	130

TABLE XIII
RESULTS OF THE INDIVIDUAL PREDICTED ACTIVITIES OF SCENARIO 1 IN PERCENTAGES (%)

Classes	Accuracy	Precision	Recall	specificity	F-score
Get UP	99	98.36	99	98.89	98.17
Bed Making	94.96	98	94.96	98	96.46
Walk	96	97.95	96	99.66	96.96
Wash hands	95.33	92.56	94.16	99.45	94.95
Brush Teeth	93.2	94.7	93.2	98.24	93.21
Comb Hair	95.33	94.72	95.33	96.56	94.52
Praying	97.33	94.16	97.33	98.34	95.72
Eat with Hand	94.31	94.3	94.31	95.78	94.3
Pour Water	96.67	95	96.67	99.69	95.86
Drink	96.41	94.87	96.41	97.79	95.64

size increases, dropout regularization becomes significant in avoiding overfitting. A high dropout adds more noise to the training process, which reduces overfitting. However, at a small dropout, the model becomes more complex, which tends to overfit the training data.

B. Confusion Matrix (CHAR-SW Dataset (Sc1))

In this appendix, we provide more details on the confusion matrix of scenario 1 (CHAR-SW dataset) as shown in Table XII. This scenario has chosen as an example to measure the following metrics (i.e Precision, Recall, Specificity and F-score) as in Eq., 7, 8, 9, 10 and 11 which are evaluated in Table XIII. The metrics evaluated the offline CHARM-Deep after applying the IFQ-SAME descriptor. All other scenarios in the dataset are evaluated also using the same metrics. The equations have some abbreviations such as TP refers to the true positive, FP is the false positive, TN is the true negative and FN is the false negative. N is the data reading number of each activity inside the stream.

$$Accuracy = \frac{TP}{N} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{10}$$

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

REFERENCES

[1] S. Katz *et al.*, "Multidisciplinary studies of illness in aged persons II: A new classification of functional status in activities of daily living," *J. Chronic Dis.*, vol. 9, no. 1, pp. 55–62, 1959.

[2] R. Elbasiony and W. Gomaa, "A survey on human activity recognition based on temporal signals of portable inertial sensors," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* Cham, Switzerland: Springer, 2019, pp. 734–745.

[3] A. Papagiannaki *et al.*, "Recognizing physical activity of older people from wearable sensors and inconsistent data," *Sensors*, vol. 19, no. 4, p. 880, 2019.

[4] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *Proc. 23th Int. Conf. Archit. Comput. Syst.*, 2010, pp. 1–10.

[5] S. A. Mohammed and W. Gomaa, "Exploration of unknown map for safety purposes using wheeled mobile robots," in *Proc. ICINCO*, 2017, pp. 359–367.

[6] S. A. Mohammed and W. Gomaa, "Exploration of unknown map for gas searching and picking up objects using Khepera mobile robots," in *Proc. ICINCO*, 2016, pp. 294–302.

[7] J. Yang, "Toward physical activity diary: Motion recognition using simple acceleration features with mobile phones," in *Proc. 1st Int. Workshop Interact. Multimedia Consum. Electron.*, 2009, pp. 1–10.

[8] E. M. Tapia, "Using machine learning for real-time activity recognition and estimation of energy expenditure," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2008.

[9] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, "Collecting and disseminating smart home sensor data in the casas project," in *Proc. CHI Workshop Develop. Shared Home Behav. Datasets Adv. HCI Ubiquitous Comput. Res.*, 2009, pp. 1–7.

[10] S. S. Intille *et al.*, "Using a live-in laboratory for ubiquitous computing research," in *Proc. Int. Conf. Pervas. Comput.* Berlin, Germany: Springer, 2006, pp. 349–365.

[11] M. Abdu-Aguye and W. Gomaa, "Robust human activity recognition based on deep metric learning," in *Proc. 16th Int. Conf. Informat. Control, Automat. Robot. (ICINCO)*, 2019, pp. 656–663.

[12] M. Abdu-Aguye and W. Gomaa, "VersaTL: Versatile transfer learning for IMU-based activity recognition using convolutional neural networks," in *Proc. 16th Int. Conf. Inform. Control, Automat. Robot. (ICINCO)*, 2019, pp. 507–516.

[13] M. G. Abdu-Aguye and W. Gomaa, "Competitive feature extraction for activity recognition based on wavelet transforms and adaptive pooling," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[14] S. Ashry, R. Elbasiony, and W. Gomaa, "An LSTM-based descriptor for human activities recognition using IMU sensors," in *Proc. 15th Int. Conf. Inform. Control, Automat. Robot. (ICINCO)*, vol. 1, 2018, pp. 494–501.

[15] *WSU CASAS Smart Home Project Aruba, Tulum Datasets*. Accessed: 2011. [Online]. Available: http://ailab.wsu.edu/casas/datasets/

[16] *IDTechEx*. Accessed: 2019. [Online]. Available: https://www.idtechex.com/en/reports/wearable-technology/52

[17] S. Liu. (May 22, 2019). *Smartwatches—Statistics & Facts*. [Online]. Available: https://www.statista.com/topics/4762/smartwatches/

[18] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[19] K. Soomro, H. Idrees, and M. Shah, "Online localization and prediction of actions and interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 459–472, Feb. 2019.

[20] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 1602–1607.

[21] W. Gomaa, R. Elbasiony, and S. Ashry, "ADL classification based on autocorrelation function of inertial signals," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 833–837.

[22] N. Yala, B. Fergani, and A. Fleury, "Towards improving feature extraction and classification for activity recognition on streaming data," *J. Ambient Intell. Hum. Comput.*, vol. 8, no. 2, pp. 177–189, Apr. 2017.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[24] G. Yavuz *et al.*, "A smartphone based fall detector with online location support," in *Proc. Int. Workshop Sens. App Phones*, Zürich, Switzerland, 2010, pp. 31–35.
[25] M. Kose, O. D. Incel, and C. Ersoy, "Online human activity recognition on smart phones," in *Proc. Workshop Mobile Sens., Smartphones Wearables to Big Data*, vol. 16, 2012, pp. 11–15.
[26] S. Ashry and W. Gomaa, "Descriptors for human activity recognition," in *Proc. 7th Int. Japan-Africa Conf. Electron., Commun., Comput. (JAC-ECC)*, Dec. 2019, pp. 1–4.
[27] O. Kilinc, A. Dalzell, I. Uluturk, and I. Uysal, "Inertia based recognition of daily activities with ANNs and spectrotemporal features," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 733–738.
[28] J. Margarito, R. Helaoui, A. M. Bianchi, F. Sartor, and A. G. Bonomi, "User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 4, pp. 788–796, Apr. 2016.
[29] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[30] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.
[32] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2016, pp. 207–212.
[33] (Jan. 6, 2020). *Apple Watch Series 4—Technical Specifications*. [Online]. Available: https://support.apple.com/kb/SP778?viewlocale=en_US&locale=en_US/
[34] J. Barton, A. Gonzalez, J. Buckley, B. O'Flynn, and S. C. O'Mathuna, "Design, fabrication and testing of miniaturised wireless inertial measurement units (IMU)," in *Proc. 57th Electron. Compon. Technol. Conf.*, 2007, pp. 1143–1148.
[35] M. Perlmutter and S. Breit, "The future of the MEMS inertial sensor performance, design and manufacturing," in *Proc. DGON Intertial Sensors Syst. (ISS)*, Sep. 2016, pp. 1–12.
[36] *Motion Sensors Explainer*. Accessed: Aug. 30, 2017. [Online]. Available: https://www.w3.org/TR/motion-sensors/
[37] T. Provot, X. Chiementin, E. Oudin, F. Bolaers, and S. Murer, "Validation of a high sampling rate inertial measurement unit for acceleration during running," *Sensors*, vol. 17, no. 9, p. 1958, 2017.
[38] A. M. Toh, R. Togneri, and S. Nordholm, "Spectral entropy as speech features for speech recognition," *Proc. PEECS*, vol. 1, p. 92, May 2005.
[39] *Instantaneous Frequency Estimation and Localization*. Accessed: 2003. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/instantaneous-frequency/
[40] Z. M. Hussain and B. Boashash, "Adaptive instantaneous frequency estimation of multicomponent FM signals using quadratic time-frequency distributions," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1866–1876, Aug. 2002.
[41] *Estimate Instantaneous Frequency*. Accessed: 2019. [Online]. Available: https://www.mathworks.com/help/signal/ref/instfreq.html
[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[43] S. K. Debnath *et al.*, "A review on graph search algorithms for optimal energy efficient path planning for an unmanned air vehicle," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 743–749, 2019.
[44] *Leaky*. Accessed: 2019. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.leakyrelulayer.html/
[45] *Elu*. Accessed: 2019. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.elulayer.html?searchHighlight=elu&s_tid=doc_srchtitle/
[46] *Relu*. Accessed: 2019. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relulayer.html/
[47] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: http://arxiv.org/abs/1511.07289

Sara Ashry received the M.Sc. degree from CSE Department, Egyptian Ministry of Higher Education in 2017. She is currently pursuing the Ph.D. degree with the Computer Science and Engineering (CSE) Department, Egypt-Japan University of Science and Technology (E-JUST), a Research Fellow with Waseda University, Japan. She is also an Assistant Researcher with Electronic Research Institute (ERI), Egypt. She granted a summer school program at Halmstald University, Sweeden, in July 2017. Her research interest includes artificial intelligence and its applications, multiagent systems, machine learning, smart cities, and IMU sensors application. She received the Scholarship for M.Sc. and Ph.D. from the Egyptian Ministry of Higher Education.

Tetsuji Ogawa (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 2000, 2002, and 2005. He is currently a Professor with School of Science and Technology, Waseda University. He has been an Associate Professor with Waseda University and Egypt-Japan University of Science and Technology (E-JUST) from 2012 to 2015. His research interests include stochastic modeling for pattern recognition, speech enhancement, speech and speaker recognition and human activity recognition. He is a member of the Information Processing Society of Japan (IPSJ) and the Acoustic Society of Japan (ASJ). He received the Awaya Prize Young Researcher Award from the ASJ in 2011 and Yamashita SIG Research Award from the IPSJ in 2013. Under his supervision a lot of researchers and postgraduate students in his own research lab at Waseda University.

Walid Gomaa (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from Alexandria University in 2000 and 2002, respectively, and the Ph.D. degree in computer science from Maryland University, USA, in 2007. He is currently a Professor with the Computer Science and Engineering Department, E-JUST, and Alexandria University, Egypt. He granted a Postdoctoral position with Loria Lab, Nancy, France, between 2008 till 2009. He was the Head of the CSE Department, E-JUST. His research interests include artificial intelligence and its applications, multiagent systems, machine learning, logic-based-artificial intelligence, bio-informatics, smart city, and speech recognition. He participated in many projects funded by ITIDA, INRIA, NII, France, and Tokyo. Under his supervision a lot of researchers and postgraduate students in his own research lab at E-JUST University.