AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 55(1) (2023) 109-126
DOI: 10.22060/miscj.2023.21984.5308

Remote elderly healthcare: a robust deep learning approach for wearable sensors-based complex activities recognition

F. Serpush[1], M. R. Keyvanpour[2]*, M. B. Menhaj[3]

[1] Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2] Department of Computer Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran
[3] Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT: Human activity recognition system (HARS) is critical for monitoring individuals' health and movements, especially the elderly and the disabled, in different environments. Wearable sensors are valuable tools for information acquisition due to the mobility of people. Researchers have proposed different methods for sensor-based HAR, which face challenges. This paper uses deep learning (DL), which combines the deep convolution neural network (CNN) and long short-term memory (DeepConvLSTM) to extract and select features to recognize high-performance activity. Then, the Softmax-Support Vector Machine (SofSVM) performs the classification and recognition operations. This paper uses a comprehensive nested pipe and filter (NPF) architecture. Filters are usually independent, but some filters can be bidirectional to improve the performance of complex activity recognition. There is two-way communication between convolutional layers and long short-term memory (LSTM). The Opportunity dataset is public and includes complex activities. The results with this dataset display that the proposed work improves the weighting F-measure of HAR. This paper has also shown other experiments; that involve comparing the number of sensors, max-pooling size, and convolutional filter size. According to this dataset, the proposed method weighting F-measure is 0.929. The proposed approach is indeed effective in activity recognition, and the NPF architecture covers all the components of the activity recognition process.

Review History:

Received: Dec. 09, 2022
Revised: Jun. 05, 2023
Accepted: Jun. 19, 2023
Available Online: Oct. 10, 2023

Keywords:

Complex human activity
convolutional neural networks
Long Short-term Memory
Wearable Sensors
Support Vector Machine

1- Introduction

Due to a significant increase in the population, the needs of each individual are not fully met, and medical services are not affordable and available [1]. Human activity recognition (HAR) technology can recognize activities through sensors and computer systems [2-5]. Sensors-based activity recognition (SAR) is categorized into two categories for receiving data: external sensors (ambient) and wearable sensors. In systems based on cameras, devices are installed in fixed locations, and the HAR based on the video data is limited to moving in the described environment. Also, two similar activities may not be recognized when using the camera, and privacy is violated. Wearable technology seems to be critical in realizing and monitoring humans at home. HARS is complex and capable of tracking individuals' status and provides valuable tools to deal with emergencies [6, 7]. Activities refer to complex behaviors [8] consisting of a sequence of actions that can interact with a subject or several subjects. We define SAR according to the following:

$S= (S_1, S_2, \ldots, S_n)$ is the total collection of signals of multiple sensors; S_i is a signal of the ith sensor, and n is the number of wearable sensors. $Win= (win_1, win_2, \ldots, win_k)$ is the total window collection that creates S segments and sends sequentially. Then the samples are created (based on voltage, mean, average, etc.). Features can be extracted based on some methods, and a feature vector $f=(f_1,f_2,...,f_N)$ for every window is formed with weights $w=(w_1, w_2,...,w_N)$. Then some features are selected of f that be named $A= (a_1, a_2, \ldots, a_j)$ that's mean $f \rightarrow A$. A is the classified and recognized activity, and m is the number of classes, meaning $A \rightarrow I_i$. I_i is the recognized activity. Finally, the activity is saved for use in the next activity (if needed).

Providing accurate and appropriate information about people's activities and behaviors is one of the essential computational tasks of HAR [9, 10]. The increasing computational power of machine learning algorithms and neural networks has led to wearable sensor-based HAR becoming popular in many fields, including smart homes [11, 12], healthcare for the elderly [13, 14], medical care, improvement of human interaction with computers, security, training of athlete monitoring, and rehabilitation system [6]. The most critical problems of the HARS are scalability, complex activities, and human behaviors in a complex environment [15, 16]. CNN-Deep learning extracts more valuable features than traditional machine learning techniques. The features learned through deep architecture are reflected as a higher-level abstract representation of low-level raw time-series signals18 ,17]]. However, there

*Corresponding author's email: keyvanpour@alzahra.ac.ir

Table 1. List of abbreviations

abbreviations and the symbols	Descript	abbreviations and the symbols	Descript
ADL	activities of daily living	KNN	K-nearest neighbor
HARS	human activity recognition system	LSTM	long Short-term Memory
HAR-NPF	HAR based on the nested pipes & filter	MLP	Multilayer perceptron
CNN	convolution neural network	NPF	nested pipe and filter
DeepConvLSTM	deep convolution neural network and long short-term memory	PN	participant number
ConvLSTM-SofSVM	deep convolution neural network and long short-term memory-Softmax-support vector machine	RBF	radial basis function
DL	Deep learning	RNN	recurrent neural network
FIR	pass Finite Impulse Response	ReLUs	rectified linear units
HAR	human activity recognition	SAR	Sensors-based activity recognition
HAR-NPF-DL	HAR-NPF and deep learning	SGD	stochastic gradient descent
HARS	human activity recognition system	SofSVM	Softmax-Support Vector Machine
HARS	Human activity recognition system	SVM	Support Vector Machine
IMU	inertia measurement units		

are challenges for CNN in recognizing human activities [19-22]; the most important ones are: 1- CNN processing units are used over time dimensions, 2- Sharing or merging CNN units among different sensors data, and 3- Selecting a small window increase computing cost. LSTM models are favored over recurrent neural networks (RNN) because of a traceable learning structure. LSTM simulations have been used for the variability of activities. The LSTM memory content is appropriate for showing complex relations when it could be in a comprehensive series [18]. Multiclass Support Vector Machine (SVM) is effective in classifying activities [19]. Therefore, this study presents combined deep learning methods to select practical features and a combination of two classifiers (ConvLSTM-SofSVM) for recognizing sensor-based human activities. Filters and their relationship are clearly defined, and each filter represents a component with a specific work in HARS. We provided Table 1 for abbreviations.

Section 2 investigates the previous HAR methods using deep learning. Section 3 presents the proposed method. Section 4 describes the results of the tests. Finally, Section 5 reviews the conclusions and future works.

2- Related Works

HARS shows the primary role [4, 23, 24] in identifying changes in the natural behavior of individuals, especially the elderly [12, 25]. HAR has attracted the consideration of several researchers for checking the elderly remotely [26]. Since HARS are always running, they must continuously improve system performance [27]. Due to reduced costs, ease of use, and the ability to provide continuous monitoring against sensors installed in fixed locations, in recent years, wearable sensors have been used in the HARS [16, 28]. Wearable sensors also have advantages compared to cameras, including 1) the camera has environmental restrictions, 2) Wearable sensors support privacy more than the camera, and 3) It's possible to use several sensors in different parts of the body for better recognition of activity [20]. Wearable sensors include the barometer [16, 29], accelerometer, and gyroscope, which are data acquisition tools [13, 30].

Deep learning approaches such as deep belief networks [23, 31], restricted Boltzmann machines, and conventional neural networks [8] are used as generalized learning methods that extract features directly from input data. These methods increase computational complexity by adding layers to

achieve higher precision [25, 32]. Features extracted using CNN are patterns of activity and non-manual [8, 19, 33]. In addition, these features also increase the precision of the recognition [17, 18].

In [23], RNN, convolution, and deep approaches have been used for AR based on recorded data from wearable sensors. In this work, two LSTM neural networks have been implemented; the first is a deep LSTM containing several recursive and forward unit layers. The latter is bidirectional LSTM, which consists of two adjacent parallel layers that have access to the past and future stages of the current stage. Generally, using more layers is one way to increase performance in the deep neural network. However, in addition to the fact that these networks do not estimate the expected performance, they sometimes lead to subsequent training errors. To solve this problem, a deep residual learning framework is proposed, which involves several residual units. The output of the convolution layers is the input of the next residual unit, and the modified linear unit function activates several composite layers. Finally, the output of the residual unit is obtained.

In [8], in the first stage, the spatial and temporal features are extracted, and then any data type in CNN is fed to extract the spatial attributes. In the second stage, a classifier encodes the binary output vector.

In [34], a general deep structure has been proposed, including convolutional activity and recursive units of LSTM. It is appropriate for wearable multipurpose sensors. This system can apply a sensor fusion and simulate the activity's time dynamics features.

In [35], large-scale data is received from a wearable sensor on the forearm. Time series data appear in image format, and CNN automatically extracts distinctive features [36].

In [24], a deep CNN is proposed for performing high-performance HAR with smartphone sensors [31, 37], in which the intrinsic features of the activity and the 1D time-series signals are extracted. Robust features of raw data are automatically extracted, and the experiments show that as the number of layers increases, the difference in the degree of complexity of the feature decreases. However, its additional layers show more of the connection and the complexity of the features.

In [38], the learning of the extreme machine is proposed based on mixed-kernel weighing. Considering that the choice of kernel heavily influences ELM, the combined kernel method is proposed. To deal with imbalances, it used a cost-sensitive method [13]. The main idea of the cost-effective method is that the cost increases with the wrong classification rate.

So, in the next section, we describe the proposed method and architecture of HARS based on the generalization of the issues mentioned.

3- Materials and Methods

HAR is an important research area with many applications [39]. This section outlines the proposed method for HAR based on the nested pipes & filter (HAR-NPF). As shown in "Fig. 1", the HARS contains five key components in which the connection between the main component and the pipe is established. The five main components are preprocessing [40], feature extraction [27], feature selection [41, 42], classification, and propagation [43]. According to "Fig. 1", this architecture is simple in terms of style and includes all the steps to recognize the activity. Each filter can have multiple ports; for example, the feature extraction filter includes two ports. Usually, the filters are independent, but given the simple and complex activities, the feature extraction filter and feature selection filter can be combined to influence one another. The feature extraction component is practical in the weighting F-measure. Feature selection is crucial for complex activities [9].

The classification affects both speed and weighting F-measure [30]. "Fig. 2" reviews each filter individually and the ConvLSTM-SofSVM proposed method on the NPF architecture (HAR-NPF and deep learning (HAR-NPF-DL)). The architecture of the ConvLSTM-SofSVM method includes all of the components mentioned. In the first step (preprocessing), the windowing was created after receiving signals, and the filtering was done. The feature extraction component is introduced in the second step to combine the convolution and pooling. The third step includes LSTM units that are two-way with the second step. The fourth step applies the fully connected layer, softmax, and multiclass SVM to the prepared features vector for classification. Then activity is recognized and sent to the destination in the propagation stage. In the following, we will describe each component in detail separately.

3- 1- Preprocessing

After sensor data acquisition, they should be sent by technologies such as Bluetooth, Wi-Fi, and wireless networks. The development of the human-computer interface and wireless networking in the body has led to the recognition of wearable sensor-based activity, becoming increasingly popular [9, 10]. Regarding "Fig. 3", each sensor receives and sends signals, and sensor data can be multi-channel. The proposed method uses a time window and specifies a label.

The window length is 20 seconds for simple activities and 5 minutes for complex activities. Considering that each example of a complex activity includes several examples of a simple activity, we have considered 20 seconds for simple activity examples and 5 minutes for complex activities. In this article, in addition to convolutional networks, LSTM which is a multi-stage recurrent network (involving several previous examples) we have used in the detection of complex activities that include several simple activities in the sequence. Therefore, we use LSTM to detect complex activity and use a larger window. In the end, we use Softmax to classify activities.

In this paper, since the input data obtained from the sensors are indexed in the time dimension, they can be considered for fixed-length time windows. Although there is no integrated system for defining simple and complex activities, this paper tries to support them. There are some basic definitions that

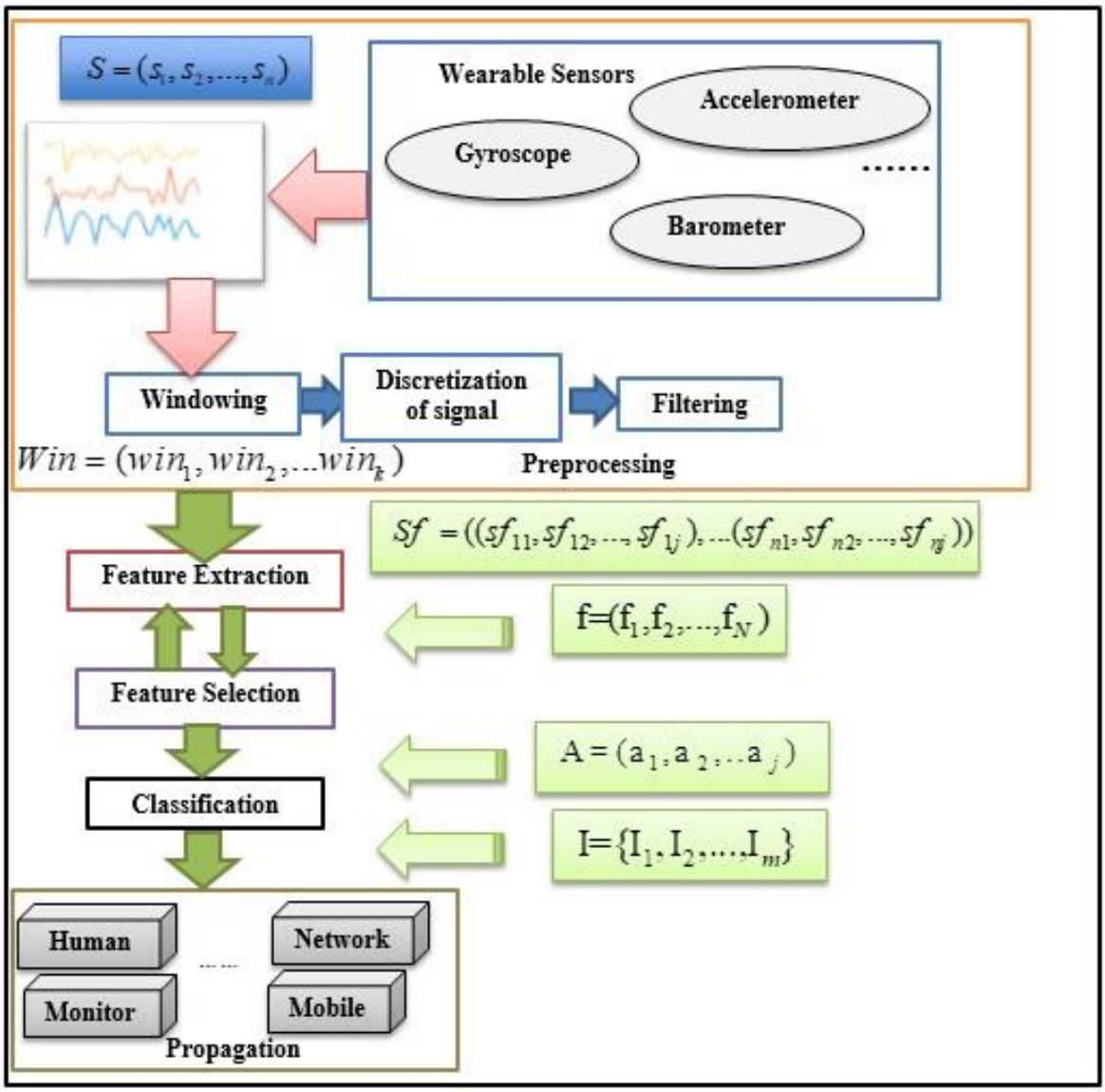

Fig. 1. NPF architecture for the recognition of sensor-based human activity

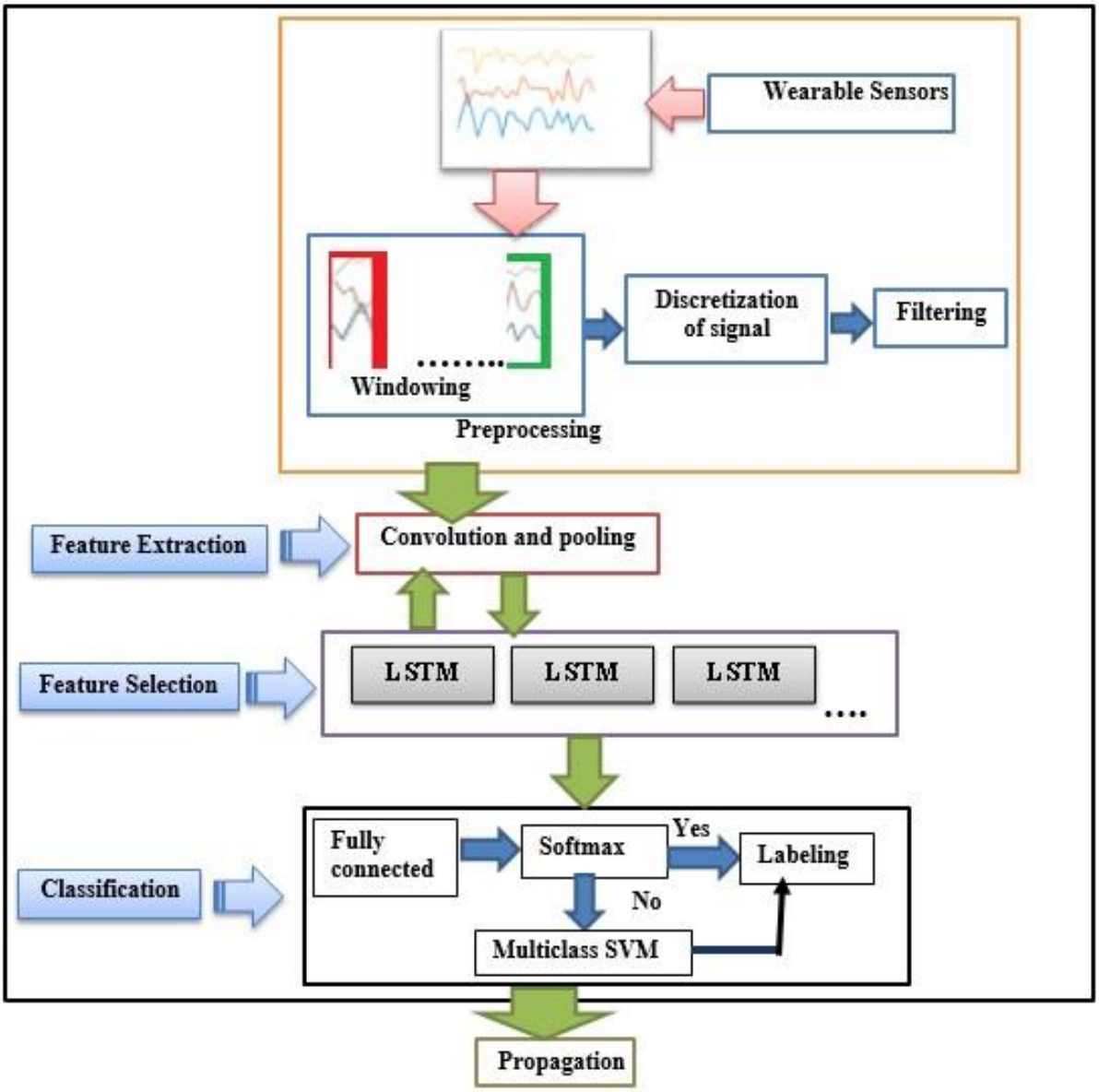

Fig. 2. The proposed deep learning method on the HAR-NPF architecture: HAR-NPF-DL (ConvLSTM-SofSVM)

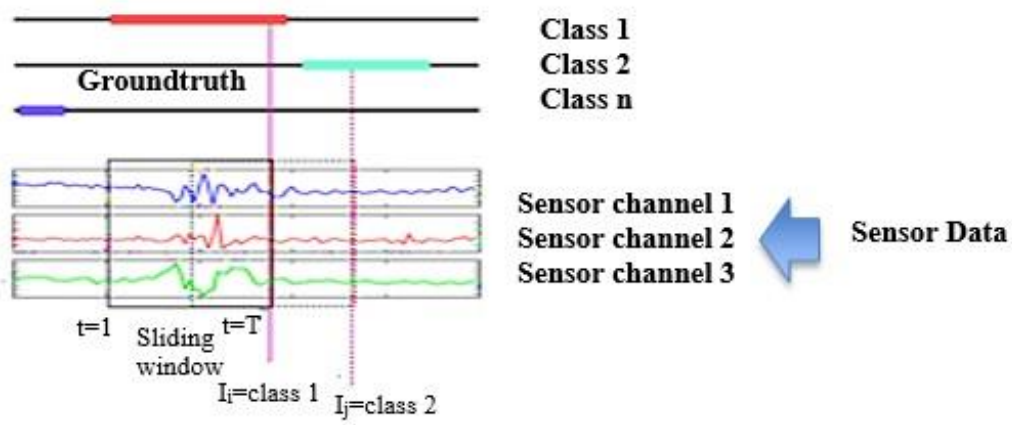

Fig. 3. Segmenting the data with a sliding window [34]

the following are mentioned.

A window has three tuples win_i= (sf, t_s, t_e) where sf specifies input data and t_s and t_e are the beginning and the end of the time window, and sf= ((sf_{11}, sf_{12}, …, sf_{1z}), …, (sf_{n1}, sf_{n2}, …, sf_{nz})) is the input vector of the gyroscope sensor, accelerometer sensor data, and manual features, respectively.

Low-level activities such as running, walking, and sitting, which involve several movements, are simple activities. A simple activity usually involves a constant length of the window (L_s is the window length of simple activity) [1, 44].

Complex activities (high-level) sometimes involve several simple activities [45] (working, eating, shopping, etc.). Complex activity has a window of constant length (L_c is the window length of complex activity).

Because complex activities are more extended and inhomogeneous, their window length must be longer than simple activities to obtain features and recognize activity [20, 23]. After the windowing, the signal is discretized, and features such as mean and average are extracted. In the next stage, noises are eliminated. Therefore, a range should be considered for information received; whenever the data is outside this range must be identified as noise and removed.

Acceleration sensors can be affected by several noises, including electronic noise or noise caused by the diffusion process and distortion in the transmission signals. These noises exist in all frequency components, and if we want to remove those parts that are noisy, we will face challenges. Therefore, it is not appropriate to use traditional filter methods. In this paper, the noise is removed using a two-stage sequential filtering method that is a combination of band-pass Finite Impulse Response (FIR) and a wavelet filter, which is the values affected by data loss and random noise, problems that are very common in wireless acceleration sensors are eliminated.

For filtering, the signal must first be decomposed into wavelet basis functions, and then noise removal is performed. To eliminate the wavelet noise, the threshold value must be determined. If the value of the wavelet coefficients is less than this threshold, it will take a value of zero, which can be seen in the following Eq. 1 [46].

$$f(s)\begin{cases} s & if\ |s|>\lambda \\ 0 & otherwise \end{cases} \quad (1)$$

λ is the specific value for the threshold.

Also, if the wavelet coefficients are below the threshold value, they are shrunk, and when the coefficients are above the threshold value, they are scaled that represented as:

$$f(s)=\max\left(0,1-\frac{\lambda}{|s|}\right) \quad (2)$$

In this paper, we have used the square root log method to determine the threshold.

After the data is ready, they are sent to the next component that extracts the features. Algorithm 1 presents a preprocessing step in HARS.

3- 2- Feature Extraction

It can be said that the feature extraction filter provides local prominence for the signals in the lower layers (to determine the disposition of any fundamental movement in human activities) and the outstanding samples of the signal in the higher layers (to specify the combination of some simple moves) [4].

Algorithm 1: preprocessing component for SAR

Input: $S= (S_1, S_2, \ldots, S_n)$, n wearable sensors

Output: Final samples with feature vectors ($sf= ((sf_{11}, sf_{12}, \ldots, sf_{1z}), \ldots, (sf_{n1}, sf_{n2}, \ldots, sf_{nz}))$)

Algorithm:

1-Segmentation of signals: $Win= (win_1, win_2, \ldots, win_k)$ (windowing) [20-23]

2-Discretization of signal (voltage, average, mean, etc.)

3- Filtering (Noise removal process (Eliminate frequencies that are outside of the range for each activity) [46].)

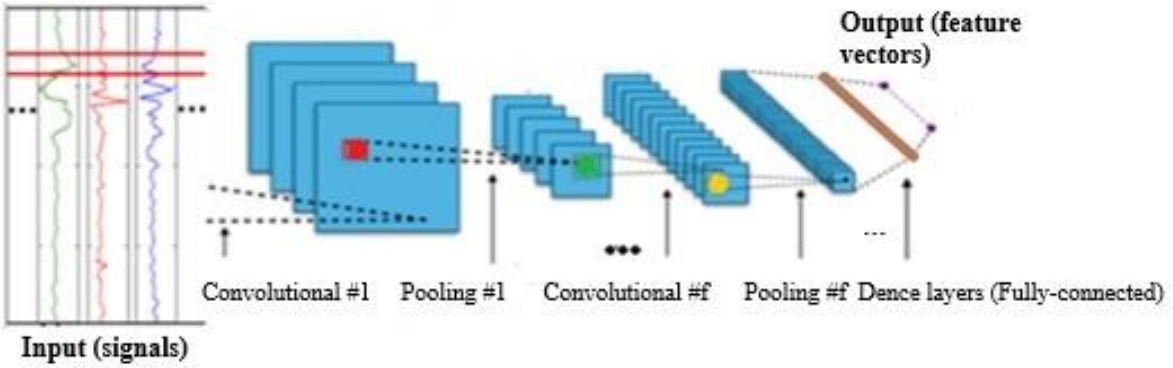

Fig. 4. DeepCNN process for the features extraction

Each layer may have many pooling or convolutional operators (which various parameters characterize them) [47]; thus, several patterns from different aspects have been considered by CNN [48], and "Fig. 4" shows how these operators are used in the DeepCNN method.

A basic interpretation is obtained when these operatives apply the same parameters to local signals (or their mappings) [37]. The output of the convolutional layer is visible in Eq. (3).

$$c_i^{l,j} = \sigma(b_j + \sum_{m=1}^{M} w_m^j sf_{i+m-1}^{0j}) \quad (3)$$

l is the index layer, σ the activation function, b_j the bias for the j_{th} character, M is the size of the kernel/filter and w_m^j is the weight of the map feature, and m is the index of the filter [37]. Each time multiply the two corresponding elements, considering a new element. This article uses max-pooling, a 5 * 5 window [44]. In general, max-pooling is the maximum value of a set of adjacent data and represents them as outputs which can be seen in Eq. (4). There are other ways for pooling, including calculating the average.

$$p_i^{l,j} = \max_{r \in R}(c_{i \times T+r}^{l,j}) \quad (4)$$

Here, R is the pooling size, and T is the pooling stride. In the deep neural network architecture, there are some convolutional and pooling layers. Convolutional layers employ rectified linear units (ReLUs) to compute the feature maps. ConvLSTM has a significant influence on the precision and weighting F-measure of HAR. The CNN is used to HAR using the extracted dynamic features. Although more computational and memory is required, CNN provides higher precision for recognition with dynamic features than conventional methods such as Multilayer perceptron

(MLP) and SVM or K-nearest neighbor (KNN) [37, 49-51]. In the proposed method, CNN is used in combination with LSTM. Dense layers perform a nonlinear transformation

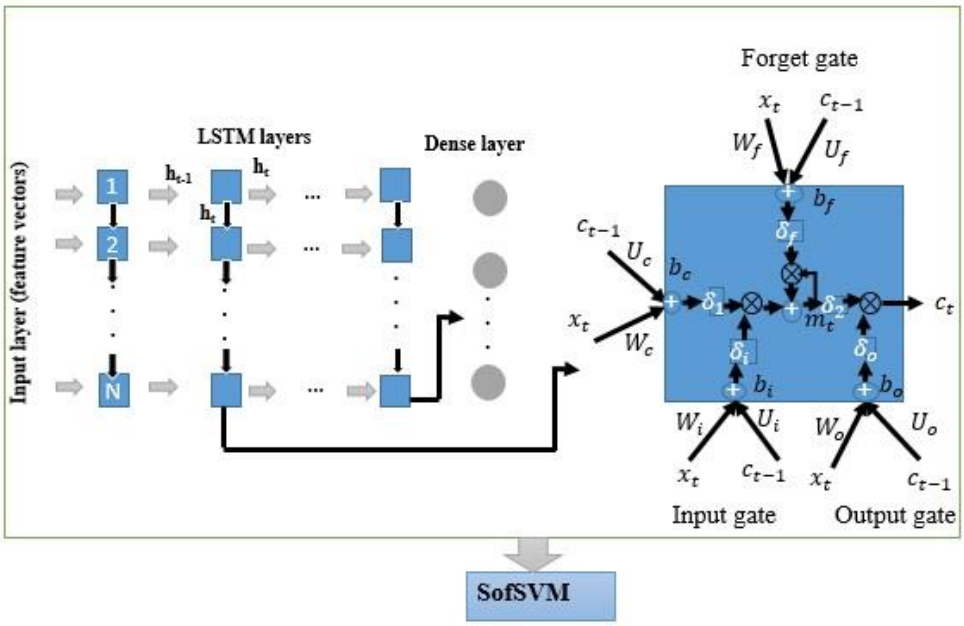

Fig. 5. Deep LSTM for feature selection in HARS

(Relu), and the units of these layers are LSTM cells. In the final $f = [f_1, ..., f_N]$ are extracted features of this filter.

3- 3- Feature Selection

In this architecture, CNN and LSTM interact with each other. LSTM cells are designed to manage the effect of gradient descent. Each LSTM cell holds the current situation in the memory location [18]. Regularly, learning is done in cells, and output is based on the current input and the history of past internal conditions; This system can store information in many steps [23]. Each LSTM unit can contain multiple memory cells that store information for a short time interval [34]. There are gateway units whose job is to regulate the contents of the memory cell and control the flow of information inside and outside the cell. When the length of the temporal input is considerable, there is an issue of spurious gradient [52] updates in the RNN, and LSTM fixes this problem.

This feature allows many of these layers to be assembled to learn the dynamics of complex input in different ranges. In "Fig. 5", the LSTM layers are described. The LSTM unit contains input, memory, and output, in which x_t is the input cell, m_t is the memory cell, and c_t is at the time of t. $\oplus$ is element-wise addition, $\otimes$ is element-wise multiplication, W_* and U_* are internal weight .matrices, b_* is the bias vector and δ_* is activation function that * is the member of the set {i, o, f}. δ_1, δ_2 are the internal activation function in the input cell and the memory, respectively.

Therefore, the value of h_t from each LSTM cell at each t period is updated, as in Eq. (5) [34].

$$
\begin{aligned}
i_t &= \delta_i (W_{ai} a_t + W_{hi} h_{t-1} W_{ci} c_{t-1} + b_i) \\
f_t &= \delta_f (W_{af} a_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f) \\
o_t &= \delta_o (W_{ao} a_t + W_{ho} h_{t-1} + W_{co} c_t + b_o) \\
h_t &= o_t \delta_h (c_t)
\end{aligned}
\tag{5}
$$

i, f, o, and c are the input gate, the freight gate, the output gate, and the cell activation vectors, all of which are the same size, and h are hidden values and σ are nonlinear functions. The a_t is the input to the memory cell layer at time t. $W_{ai}, W_{hi}, W_{ci}, W_{af}, W_{hf}, W_{cf}, W_{ac}, W_{hc}, W_{ao}, W_{ho}, W_{co}$ (W_* and U_* according to "Fig. 5") are the matrixes of weight and relationships, which W_{ai} is the input-input gate matrix and W_{ai} is the hidden-input gate matrix. b_i, b_f, b_c *and* b_o are bias vectors.

Each LSTM unit sends output in two directions (bidirected) and receives input from two directions. Multiple layers can be considered, and complex activities involving simple activities can be detected more effectively by involving previous signal sequences. Theoretically, recursive neural networks must deal with long-term dependencies to solve the exploding gradient problem and vanishing that the LSTM solves these two challenges. After extracting the property by pooling and convolution, this paper investigates the feature selection layer, including the LSTM units [31].

Algorithm 2: classification component for HAR with wearable sensors

Input: $A=\{a_1,a_2,..,a_n\}$ ($f \rightarrow A$)

Output: $I=\{I_1,I_2,...,I_m\}$ (classes or labels for activities, and m is the number of classes $A \rightarrow I_i$)

Algorithm:

1- Fully connected is applied for feature mapping (in Eq. (6)).

2- Softmax classifier is used on the z^{l+1} (in Eq. (7)) [54].

3- If softmax succeeds, I_i will be returned and saved [3].

4- Else, the SVM classifier (in Eq. (8) to Eq. (11)) is applied, and I_i will be returned and saved [55].

These units interact with the lower and upper layers. For this reason, the system's efficiency increases, and the label selection for complex activities is effective [45]. The training is done in stages; the subject's activity predicts the first and then group activity done in the second stage by the LSTM layer [18]. At each stage, the memory content of the first layer of LSTM contains specific information that results from subject performance and past performance changes. It is used to describe the whole activity of the group when the memory content includes data from all the subjects present in the environment. Therefore, combining the deep CNN model and the first layer of LSTM can be used to extract complex features and time features [34]. Consequently, it is concluded that different strategies can be used to collect these features anywhere in the environment and at any reasonable time [53].

3- 4- Classification

The label associated with the time window will be specified in the classification component in Algorithm 2 presented. The fully connected layer is first applied in the proposed method, and then the softmax classifier is located practically

after the last layer in the deep neural network. After several convolutional and pooling layers, fully connected layers are applied [54], which integrates the output of the feature mappings seen in Eq. (6).

$$z^{l+1} = x_i^{lo} w_f \tag{6}$$

The z-vector contains a not-normalized log probability, and w_f is a weight vector. The results from the deep CNN-LSTM layers are a feature vector $A^l = [a_1,...,a_l]$, which is sent as input to softmax. According to [37], the softmax classifier formula is in Eq. (7) [3].

$$soft\max = \frac{p(c \mid A)\arg\max_{c \in C} \times \exp(A^{L-1} w^L + b^L)}{\sum_{k=1}^{N_C} \exp(A^{L-1} w_k)} \tag{7}$$

c is the activity class, L is the last-level index, and Nc is the number of activity classes. After the forward release, wrong values are generated that should be updated and get the error to a minimum through training by stochastic gradient descent (SGD) on several training data. Backpropagation balances weights by calculating the gradient of convolution weights. After the class of activity has been specified by softmax, it is possible to belong to two or more classes then SVM is used. We defined a threshold. SVM is used when the most significant probability of softmax is lower than the threshold. Since SVM is more effective than other classifiers in dealing with small data sets and high-dimensional spaces, this article uses it in specific conditions in addition to softmax. Finding the optimal hyperplane that separates the feature space is a special idea of SVM. Therefore, a kernel is applied, and the samples are transferred to a space with high dimensions if the samples are not linearly separable. Then the boundary between the classes is specified, and similar activities are recognizable. Data patterns from different locations and individuals sometimes vary for the same activity. In order to enhance the recognition performance in the HARS, the knowledge of the previous source has been used when learning new examples [56, 57]. Therefore, SVM is a classifier used to obtain an optimal hyperplane, categorizing data into two classes. The two classes have expanded to several classes to solve this problem [57, 58]. While the hyperplane's distance from the class boundary is large, the weighting F-measure increases and reduces the risk of incorrect classification from the two training and validation datasets [59]. SVM dramatically reduces the complexity of time and memory. The size of the support vectors with this method can be limited, and the implementation time and test will be reduced [57].

The SVM classifier function is seen in Eq. (8) [55].

$$f(x)=\sum_{i=1}^{l} y_i a_i K(x_i,x)+b \tag{8}$$

The kernel function $K(x_i,x)$ is used to measure the training vector (x_i,x). y_i is labels. There are several kernel functions that we used radial basis function (RBF) kernel. RBF can be seen in Eq. (9).

$$K(x_l,x)=\exp\left\{-(X-X_l)^2/\sigma^2\right\} \tag{9}$$

σ is the variance and hyperparameter. $(X-X_l)^2$ is for Squared Euclidean Distance.

Since RBF is highly efficient57 ,55]], we use this kernel in this paper. The objective function must be minimized to obtain the optimal classification function, as seen in Eq. (10).

$$\begin{aligned} &Minimize\ \frac{1}{2}\|(W\ \|_2)^2+C\sum_{i=1}^{N}\epsilon_i \\ &subject\ to\ y_i\left(W^T\varphi(x_i)+b\right)\geq 1-\epsilon_i\ ,\forall \mathrm{i} \\ &\epsilon_i\geq 0\ \ i=1,\ldots,N \\ &\varphi: R^M\rightarrow R^S, W\in R^S\ \ and\ b\in R \end{aligned} \tag{10}$$

Parameter C is a penalty parameter that the user can set. This parameter specifies the degree to which defect variables are involved in determining the margin. **ε is the slack** variable. Data samples are presented in an M-dimensional space (R^M) and $y_i\in\{+1,-1\}$. Multiclass difficulty can be addressed by well-stabled methods such as one-vs-one. The following equations are obtained using Lagrange coefficients and simplifying (Eq. (11) and Eq. (12)).

$$\begin{aligned} L(\alpha)=&\sum_{i=1}^{N}\alpha_i- \\ &\frac{1}{2}\sum_{i=1}^{N}\sum_{J=1}^{N}\alpha_i\alpha_j y_i y_j\left\langle\varphi(x_i),\varphi(x_j)\right\rangle \end{aligned} \tag{11}$$

$$\sum_{i=1}^{N} a_i y_i=0,\quad a_i\geq 0,\quad i=1,\ldots,N \tag{12}$$

α is the dual variable. See [55, 57] for a more in-depth mathematics analysis of SVM.

4- Experiments and results

In this section, the dataset and the required tools are described. Then the evaluation criteria, implementation process based on the dataset, and the determination of the hyperparameters are described. Finally, the tests are done, and the results are discussed.

4- 1- Dataset and tools

The Opportunity dataset is public and robust and includes complex activities derived from several wearable sensors [3, 44], which characteristics are shown in Table 2. This dataset contains 6 hours of activity data from 4 participants (participant number (PN)).

Different sensors are placed in various positions of the participants' bodies to record activities, and the participants perform their activities of daily living (ADL). For this dataset, five commercial inertia measurement units (IMU), two commercial inertia sensors, and 12 Bluetooth accelerometer sensors have been used. Each IMU consists of a 3D accelerometer, a 3D gyroscope, and a 3D magnetic sensor. Each participant performed 5 ADL sessions and one practice session. During each ADL session, participants perform limited activities. Participants repeat a predetermined set of 17 exercises in the training session 20 times. Data can be grouped, depending on the complexity, into high-level activities, medium-level activities (such as arm movements), and low-level activities (left and right movements, use of objects) [26]. The Opportunity dataset is classified, and a sequence of activities can be observed. Simple activities (locomotion) include "walking," "standing," "sitting," etc. Complex activities (high level) include "coffee time," "sandwich time," etc., and gestures (mid-level) have a "close drawer," "drinking from a cup," etc. "Fig. 6" shows how the sensors were connected and the location of the connection [60]. Software tools used for implementation are described in Table 3.

4- 2- Evaluation criteria and test method

F-Measure considers the precision and recall for each class and provides a better evaluation of performance than accuracy. Then, we used weighting F-measure (F1) criteria for the performance evaluation of the proposed method and comparison with state-of-the-art methods [13, 23, 28]. F1 can be seen in Eq. (13).

$$F_1=\sum_i 2*w_i\frac{precision_i\times recall_i}{precision_i+recall_i} \tag{13}$$

i is the class index and w_i is the ratio of class i in all samples ($w_i=n_i/N$ where n_i is the number of samples from the i-th class and N is the total number of samples), the $precision_i$ is the total number of the samples of class i which

Table 2. Details of the Opportunity Dataset

PN	Channels Number	Sensors		Frequency	Simple Activities (Locomotion)	Complex Activities (high level)	Gestures (mid-level)	Action (low level)
		Type	Number					
4	113	commercial RS485-networked XSense inertial measurement units (IMU)	5	30Hz	Stand, Walk, Sit, Lie	Relaxing, Coffee time, Early morning, Cleanup, Sandwich time	Open Door 1 Open Door 2 Close Door 1 Close Door 2 Open Fridge Close Fridge Open Dishwasher Close Dishwasher Open Drawer 1 Close Drawer 1 Open Drawer 2 Close Drawer 2 Open Drawer 3 Close Drawer 3 Clean Table Drink from Cup Toggle Switch Null	Actions: reach, release, grasp, open, etc. Objects: fridge, door, glass, etc.
		commercial InertiaCube3 inertial sensors	2					
		Bluetooth acceleration sensors	12					
		3D accelerometer	1					
		3D gyroscope	1					
		3D magnetic	1					

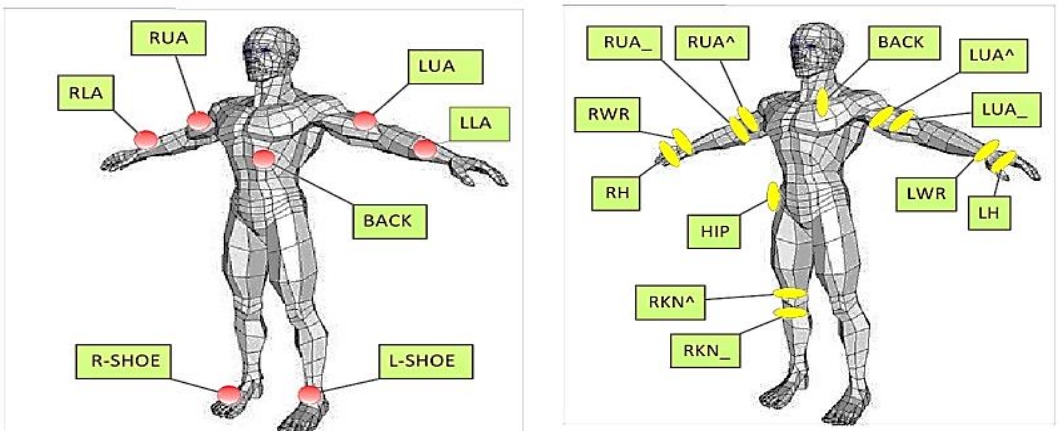

Fig. 6. The location of the wearable sensor in the Opportunity dataset (Red: IMU sensors; Yellow: 3-axis accelerometers) [34, 60]

Table 3. Software tools used to implement

Tools	Type
Operating System	**Windows 10**
Programing language	**Python 3.6**
IDE	**Jupyter(Notebook) 5.6.0**

Table 4. Parameters in the ConvLSTM-SofSVM method

Number of classes	Number of filter convolutional	Size of filter convolutional	Size of max-pooling	Number of units LSTM	Number of samples
18	64	5*5	5*5	128	676713
		4*4	4*4		
		3*3	3*3		
		2*2	2*2		
		1*1	1*1		

are correctly predicted on total number predicted samples, namely precision is defined as $\frac{\text{TP}}{\text{TP}+\text{FP}}$ [13, 23, 28]. $recall_i$ is the total number of samples of class i that are correctly predicted on the total number of the correct samples, namely recall is defined as $\frac{TP}{TP+FN}$ [34, 60]. TP, FP are the number of true and false positives, respectively, and FN matches the number of false negatives.

It used 557963 data for the train and 118750 data for the test based on the Opportunity dataset (0.83 data for the train and 0.17 data for the test) [13, 34]. The number and size of the parameters related to the feature extraction and the feature selection for the architecture and the proposed method can be seen in Table 4. We extracted and selected features using a deep model (ConvLSTM in HAR-NPF-DL proposed method). SofSVM gives the high weighting F-measure in classifying some activities. We used categorical cross-entropy (loss function), mini-batch gradient descent, and the RMSProp. The sum of the probabilities assigned to each class is one in the softmax classifier. Therefore, the possibility of error increases in similar or complex activities. Thus, after the recognition of activity by softmax, it is possible to belong to two or more classes; SVM is used when the most significant probability of softmax is lower than the threshold.

4- 3- Results and discussion

The nested pipes & filter system uses a combination of convolution and pooling to extract the feature, which is deep CNN. Then, it uses recursive LSTM units and holds several steps to recognize complex activities. Classification operations with SVM and softmax are performed, and eventually, propagation. The architecture is HAR-NPF, and after applying the ConvLSTM-SofSVM proposed method in the architecture, we called the architecture HAR-NPF-DL. The following will describe the experiments in sequence.

4- 3- 1- Experiment based on sensor number

The selection of appropriate and optimal locations of sensors in different body positions and their number are studied by researchers.

It is tried that the HARS is the most efficient with the minimum number of sensors. We compare the deep learning methods in terms of F1 criteria based on the number of sensors, as shown in Table 5 (in both DeepConvLSTM [34] and hybrid [61] methods); when the number of sensors increases (it rises than a specific value), F1 decreases. The DeepConvLSTM method [34] combines convolutional and dense layers. LSTM cells are used in dense layers. Convolutional layers do not include a pooling operation. Classification in this method is done using Softmax. According to the ConvLSTM-SofSVM method, in contrast to the two methods mentioned, when the number of sensors is 19, it has a higher weighting F-measure and is more effective in complex AR. F1 in the DeepConvLSTM method [34] is more than in other methods.

The sensor number is 10 (Skoda dataset); therefore, we understand that the weighting F-measure may decrease when the sensor number increases. In [61], when the sensor number is 19 (Opportunity dataset) and the channel number is 109, F1 is less than when the number of sensors is 3 (UniMiB-HARS dataset).

To increase the amount and variety of data, one of the critical factors is the number of sensor channels, which directly affects the variety. In this paper, due to the particular importance of the extracted features in recognizing the activity, this component is considered, and the feature extraction method is studied. According to the evaluations, feature extraction with convolution and feature selection based on LSTM is suitable for sensor signals. On the other hand, the layers are different in dealing with the signals of various sensors. It can be concluded that the higher number of sensors interrupts the identification of a unique feature in the data of each activity. Also, the location of the sensors significantly impacts the detection, provided that the user does not feel uncomfortable in the location selected for the sensor.

Table 5. Comparison SAR previous methods with the ConvLSTM-SofSVM method based on the number of sensors

Activity classification	Sensor Number	F1
DeepConvLSTM[34]	19	**0.9015**
	10	**0.958**
Hybrid[61]	19	**0.915**
	3	**0.7365**
ConvLSTM-SofSVM (ours)	**19**	**0.929**

4- 3- 2- Test2: Experiment based on the size of the pooling

We will evaluate different sizes of the pooling window. Based on this assessment, can regulate the parameters better. The effects of different size pooling windows and even when they do not have pooling have been reviewed and evaluated.

This paper used max-pooling between convolutional layers and LSTM units [23, 34]. The fully connected layers and softmax are placed at the end. This work evaluates Pooling sizes from 1(1*1) to 5 (5*5). The results are shown in "Fig. 7". The results are better when max-pooling (size> 1*1) is used. Max-pooling only reduces the size of the convolutional output. The pooling layer can preserve the invariant features; therefore, an additional layer that performs max-pooling is stacked over the convolutional layer to reduce the sensitivity of the output. Scale-invariant feature protection is another crucial distinguishing of CNN, which the max-pooling layer achieves [44]. In the max-pooling layer, features from the convolutional layer are split into several parts.

The max operation is applied to output the values [62]. Then a fully connected network is followed by the max-pooling layer. It can be concluded that the task of convolution is to extract the features while the pooling layer recombines extracted features in a more meaningful way. So these two types of layers extract special features throughout the input window. The use of pooling dramatically reduces the number of parameters that can be trained and contributes to high precision during network training. Due to sequential pooling procedures in each layer, the output becomes increasingly sensitive to small input changes, which ultimately help identify unique time-domain activity for classification. A higher level of abstraction can be achieved by accumulating several pooling layers [63]. This paper evaluates the sensitivity of various pooling window sizes, and the best result will be obtained when the size is 3. According to the Opportunity dataset, the proposed method weighting F-measure is 0.929.

4- 3- 3- Test3: Convolutional Kernel Size

Convolution layers are capable more than other statistical and machine learning techniques feature learning and temporal dependence detection in HARS.

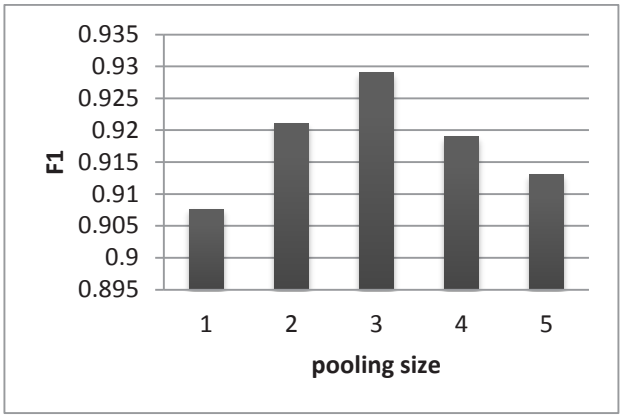

Fig. 7. The effect of pooling size on F1 based on the Opportunity dataset

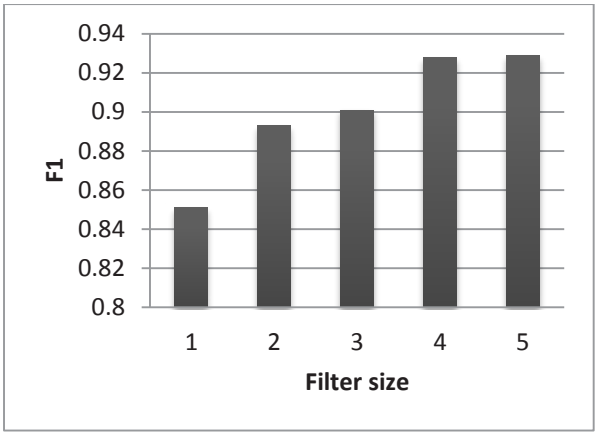

Fig. 8. Effect of convolution kernel size on F1 based on Opportunity dataset

Table 6. Results of the evaluation of the proposed method and other deep methods on the Opportunity dataset

Methods	F_1
CNN-LSTM-ELM [13]	0.918
LSTM Ensemble [26]	0.72
CNN-2 [69]	0.9152
LSTM-S [23]	0.912
Baseline CNN [34]	0.864
AROMA [45]	0.927
LR-ConvLSTM [70]	0.875
LSTM	0.893
DeepConvLSTM-Softmax	0.9076
ConvLSTM-SofSVM (ours)	**0.929**

The number of filters and the filter size significantly impact the performance of layers. The convolutional layer is more efficient for processing large time dimension sequences than sliding windows [64]. We examined the sizes 1 to 5 for the convolutional kernel, which is shown in "Fig. 8". For the Opportunity dataset, size 5 works the best for filtering, so this size works better for this dataset and the proposed method. In kernel sizes 1 to 5, F1 changes from 0.851 to 0.929. To obtain local dependencies over time and spatial, kernels are effective in both the convolution layer and the pooling on CNN.

CNN approaches only reflect a 1D kernel, which is not suitable for the usability of multiple sensors. With a 1D kernel, CNN only can take local dependency over time. Still, it misses the dependence between signals from different axes of the same sensor and sensors in various positions [65-68]. Therefore, CNN with a 2D kernel is more suitable for HAR with multiple sensors. Before the final model training, the hyper-parameter of the convolution kernel size must be determined for all layers we use in our proposed method.

4- 3- 4- Test4: Experiment of ConvLSTM-SofSVM

This section compares the ConvLSTM-SofSVM proposed method with other deep methods, as shown in Table 6. The feature extraction and selection were performed using convolution filters, pooling filters, and LSTM units [32, 34]. The probability of error in the classification is significantly reduced. Due to the Opportunity dataset being a dataset of complex activities, this method can recognize complex activities. This dataset is imbalanced, and Deep ConvLSTM acts effectively against it.

Also, using Deep ConvLSTM and a low number of sensors, HARS has high performance detecting upper body activities. On the other hand, we used the advantage of SVM with a proper kernel and softmax. Therefore, the proposed ConvLSTM-SofSVM method is more efficient than other deep models. Table 6 shows ConvLSTM-SofSVM improved by 1.1% compared to the CNN-LSTM-ELM [13]. Our method has been enhanced by 2.2%, 3.6%, and 5.4% compared to the DeepConvLSTM-Softmax, LSTM, and LR-ConvLSTM

methods, respectively.

The proposed method in this paper has a higher weighting F-measure than other methods. Baseline CNN [34] offers acceptable results in locomotion recognition but encounters errors in gesture recognition.

AROMA has a higher efficiency than both previous methods for recognizing simple and complex activity, but the labels have two levels and limitations. Baseline CNN offers acceptable results in locomotion recognition but encounters errors in gesture recognition. AROMA is also more effective than traditional methods for recognizing simple and complex activities, but the labels have two levels and limitations. "Clean up" and "coffee time" are worse than the other three complex activities. The "Cleanup" activity joins to be classified as "sandwich time" by mistake because it is similar to a sequence of kitchen activities. "Coffee time" activities also tend to be classified as "sandwich time" by mistake because they are both based on the simple activity of "sitting." Moreover, some "coffee time" examples are known as "early morning." This mistake could be that these two activities prepare food and have common parts, such as pouring water [45]. There are no considerable efficiency changes for most activities when rectifying the sequence length. However, shorter gestures seem to advantage from being completely involved in the sequence detected by the model. Several short gestures include "Open Drawer 1", "Close Drawer 1", "Open Drawer 2", and "Close Drawer 2". While the motion length is longer than the sequence length, DeepConvLSTM can only develop a classification result based on a partial view of the temporal unfolding of the features within the sequence [34]. The main challenges of HARS are the nature of human activity, which includes complex features, changing its pattern in different environments, performing various activities similarly, extracting knowledge, and monitoring immediately. These challenges can lead to reduced recognition accuracy. Machine learning methods and deep neural networks have been effective in solving these challenges but still have problems detecting complex activities. Therefore, this paper presented a combined deep neural network method to provide better performance for the HAR. However, results show that ConvLSTM-SofSVM can find worthy efficiency.

5- Conclusion

This paper used a coherent and systematic architecture (HAR-NPF) that includes independent filters, but some filters are two ways to detect complex activities. We also presented the ConvLSTM- SofSVM method, which combines a deep convolutional neural network and bi-directional LSTM units to extract and select features. The SofSVM classifier was proposed for labeling the activity. We performed different experiments using varying convolution and max-pooling filters and then examined them. We also analyzed the number of sensors for detection. The results show that the proposed method has improved the weighting F-measure. AR in the smart healthcare system has attracted the attention of many researchers who can monitor subjects through the environment and body with sensors. In future work, it is possible to improve the architecture to predict the next activity and to be able to recognize the events with more accuracy and weighting F-measure. We also will use fuzzy and semantic systems to improve deep networks in future studies and examine different datasets in this field.

References

[1] Keyvanpour, MR., Khanbani, N., Aliniya, Z. (2021). Detection of individual activities in video sequences based on fast interference discovery and semi-supervised method. Multimedia Tools and Applications, 80, 13879-13910.

[2] Kharati, E., Khalily-Dermany, M., & Kermajani, H. (2019). Increasing the Value of Collected Data and Reducing Energy Consumption using Network Coding and Mobile Sinks in Wireless Sensor Networks. AUT Journal of Modeling and Simulation, 51(1), 3-14.

[3] Ezatzadeh, S., Keyvanpour, M. R., & Shojaedini, S. V. (2021). A human fall detection framework based on multi-camera fusion. Journal of Experimental & Theoretical Artificial Intelligence, 1-20.

[4] Serpush, F., Menhaj, M. B., Masoumi, B., & Karasfi, B. (2022). Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System. Computational Intelligence and Neuroscience, 2022.

[5] Hassan, M. M., Huda, S., Uddin, M. Z., Almogren, A., & Alrubaian, M. (2018). Human activity recognition from body sensor data using deep learning. Journal of medical systems, 42(6), 1-8.

[6] Xu, S., Zhang, L., Huang, W., Wu, H., & Song, A. (2022). Deformable Convolutional Networks for Multimodal Human Activity Recognition Using Wearable Sensors. IEEE Transactions on Instrumentation and Measurement, 71, 1-14.

[7] Qi, W., Su, H., Yang, C., Ferrigno, G., De Momi, E., & Aliverti, A. (2019). A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone. Sensors, 19(17), 3731.

[8] Kuncan, F., Kaya, Y., Tekin, R., & Kuncan, M. (2022). A new approach for physical human activity recognition based on co-occurrence matrices. The Journal of Supercomputing, 78(1), 1048-1070.

[9] Ahad, M. A. R., Antar, A. D., & Ahmed, M. (2021). Deep learning for sensor-based activity recognition: recent trends. IoT Sensor-Based Activity Recognition, 149-173.

[10] Razzaq, M. A., Cleland, I., Nugent, C., & Lee, S. (2020). SemImput: Bridging Semantic Imputation with Deep Learning for Complex Human Activity Recognition. Sensors, 20(10), 2771.

[11] Suto, J., Oniga, S., Lung, C., & Orha, I. (2020). Comparison of offline and real-time human activity recognition results using machine learning techniques. Neural computing and applications, 32(20), 15673-15686.

[12] Miranda, L., Viterbo, J., & Bernardini, F. (2022). A survey on the use of machine learning methods in context-aware middlewares for human activity recognition. Artificial Intelligence Review, 55(4), 3369-3400.

[13] Sun, J., Fu, Y., Li, S., He, J., Xu, C., & Tan, L. (2018). Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors. Journal of Sensors, 2018.

[14] de la Concepción, M. Á. Á., Morillo, L. M. S., García, J. A. Á., & González-Abril, L. (2017). Mobile activity recognition and fall detection system for elderly people using Ameva algorithm. Pervasive and Mobile Computing, 34, 3-13.

[15] Alemdar, H., Van Kasteren, T. L. M., & Ersoy, C. (2017). Active learning with uncertainty sampling for large scale activity recognition in smart homes. Journal of Ambient Intelligence and Smart Environments, 9(2), 209-223.

[16] Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., & Chouireb, F. (2020). Fall detection and human activity classification using wearable sensors and compressed sensing. Journal of Ambient Intelligence and Humanized Computing, 11(1), 349-361.

[17] Inoue, M., Inoue, S., & Nishida, T. (2018). Deep recurrent neural network for mobile human activity recognition with high throughput. Artificial Life and Robotics, 23(2), 173-185.

[18] Serpush, F., & Rezaei, M. (2021). Complex human action recognition using a hierarchical feature reduction and deep learning-based method. SN Computer Science, 2(2), 1-15.

[19] Rajabi, M., & Khaloozadeh, H. (2020). Long-term prediction in Tehran stock market using a new architecture of Deep neural networks. AUT Journal of Modeling and Simulation, 52(2), 4-4.

[20] Mukherjee, D., Mondal, R., Singh, P. K., Sarkar, R., & Bhattacharjee, D. (2020). EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. Multimedia Tools and Applications, 79(41), 31663-31690.

[21] Hassan, M. M., Ullah, S., Hossain, M. S., & Alelaiwi, A. (2021). An end-to-end deep learning model for human activity recognition from highly sparse body sensor data in internet of medical things environment. The Journal of Supercomputing, 77(3), 2237-2250.

[22] Thakur, D., & Biswas, S. (2021). Feature fusion using deep learning for smartphone based human activity recognition. International Journal of Information Technology, 13(4), 1615-1624.

[23] Hammerla, N. Y., Halloran, S., & Plötz, T. (2016, July). Deep, convolutional, and recurrent models for human activity recognition using wearables. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (pp. 1533-1540).

[24] Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. Expert systems with applications, 59, 235-244.

[25] De Vita, A., Russo, A., Pau, D., Di Benedetto, L., Rubino, A., & Licciardo, G. D. (2020). A partially binarized hybrid neural network system for low-power and resource constrained human activity recognition. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(11), 3893-3904.

[26] Guan, Y., & Plötz, T. (2017). Ensembles of deep lstm learners for activity recognition using wearables. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(2), 1-28.

[27] Wang, L. (2016). Recognition of human activities using continuous autoencoders with wearable sensors. Sensors, 16(2), 189.

[28] Munoz-Organero, M., & Ruiz-Blazquez, R. (2017). Time-elastic generative model for acceleration time series in human activity recognition. Sensors, 17(2), 319.

[29] Semwal, V. B., Gupta, A., & Lalwani, P. (2021). An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition. The Journal of Supercomputing, 77(11), 12256-12279.

[30] Paraschiakos, S., Cachucho, R., Moed, M., van Heemst, D., Mooijaart, S., Slagboom, E. P., ... & Beekman, M. (2020). Activity recognition using wearable sensors for tracking the elderly. User Modeling and User-Adapted Interaction, 30(3), 567-605.

[31] Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. Future Generation Computer Systems, 81, 307-313.

[32] Zhao, Y., Yang, R., Chevalier, G., Xu, X., & Zhang, Z. (2018). Deep residual bidir-LSTM for human activity recognition using wearable sensors. Mathematical Problems in Engineering, 2018.

[33] Kim, E. (2020). Interpretable and accurate convolutional neural networks for human activity recognition. IEEE Transactions on Industrial Informatics, 16(11), 7190-7198.

[34] Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), 115.

[35] Jiang, Y., Hernandez, V., Venture, G., Kulić, D., & K. Chen, B. (2021). A data-driven approach to predict fatigue in exercise based on motion data from wearable sensors or force plate. Sensors, 21(4), 1499.

[36] Lee, S. M., Yoon, S. M., & Cho, H. (2017, February). Human activity recognition from accelerometer data using Convolutional Neural Network. In 2017 ieee international conference on big data and smart computing (bigcomp) (pp. 131-134). IEEE.

[37] Ronao, C. A., & Cho, S. B. (2015, November). Deep convolutional neural networks for human activity recognition with smartphone sensors. In International Conference on Neural Information Processing (pp. 46-53). Springer, Cham.

[38] Wu, D., Wang, Z., Chen, Y., & Zhao, H. (2016). Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. Neurocomputing, 190, 35-49.

[39] Hassani, S. A., Mobaraki, H., Bayat, M., & Mafimoradi, S. (2013). Right place of human resource management in the reform of health sector. Iranian journal of public health, 42(1), 56.

[40] Davila, J. C., Cretu, A. M., & Zaremba, M. (2017). Wearable sensor data classification for human activity recognition based on an iterative learning framework. Sensors, 17(6), 1287.

[41] Bazgir, O., Habibi, S. A. H., Palma, L., Pierleoni, P., & Nafees, S. (2018). A classification system for assessment and home monitoring of tremor in patients with Parkinson's disease. Journal of medical signals and sensors, 8(2), 65.

[42] Nazari, J., Fathi, P. S., Sharahi, N., Taheri, M., Amini, P., & Almasi-Hashiani, A. (2022). Evaluating Measles Incidence Rates Using Machine Learning and Time Series Methods in the Center of Iran, 1997–2020. Iranian Journal of Public Health, 51(4), 904.

[43] Muzammal, M., Talat, R., Sodhro, A. H., & Pirbhulal, S. (2020). A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks. Information Fusion, 53, 155-164.

[44] Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., & Liu, Y. (2021). Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. ACM Computing Surveys (CSUR), 54(4), 1-40.

[45] Peng, L., Chen, L., Ye, Z., & Zhang, Y. (2018). Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(2), 1-16.

[46] Davila, J. C., Cretu, A. M., & Zaremba, M. (2017). Wearable sensor data classification for human activity recognition based on an iterative learning framework. Sensors, 17(6), 1287.

[47] Yao, R., Lin, G., Shi, Q., & Ranasinghe, D. C. (2018). Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. Pattern Recognition, 78, 252-266.

[48] Kautz, T., Groh, B. H., Hannink, J., Jensen, U., Strubberg, H., & Eskofier, B. M. (2017). Activity recognition in beach volleyball using a deep convolutional neural network. Data Mining and Knowledge Discovery, 31(6), 1678-1705.

[49] Suto, J., & Oniga, S. (2018). Efficiency investigation of artificial neural networks in human activity recognition. Journal of Ambient Intelligence and Humanized Computing, 9(4), 1049-1060.

[50] Liu, H., & Wang, L. (2018). Gesture recognition for human-robot collaboration: A review. International Journal of Industrial Ergonomics, 68, 355-367.

[51] Kang, J., Kim, J., Lee, S., & Sohn, M. (2020). Transition activity recognition using fuzzy logic and overlapped sliding window-based convolutional neural networks. The Journal of Supercomputing, 76(10), 8003-8020.

[52] Abedini, F., Menhaj, M. B., & Keyvanpour, M. R. (2019). An MLP-based representation of neural tensor networks for the RDF data models. Neural Computing and Applications, 31(2), 1135-1144.

[53] Shu, X., Zhang, L., Sun, Y., & Tang, J. (2020). Host–parasite: Graph LSTM-in-LSTM for group activity recognition. IEEE transactions on neural networks and learning systems, 32(2), 663-674.

[54] Reining, C., Niemann, F., Moya Rueda, F., Fink, G. A., & ten Hompel, M. (2019). Human activity recognition for production and logistics—a systematic literature review. Information, 10(8), 245.

[55] Taha, Z., Musa, R. M., Majeed, A. P. A., Alim, M. M., & Abdullah, M. R. (2018). The identification of high potential archers based on fitness and motor ability variables: A Support Vector Machine approach. Human movement science, 57, 184-193.

[56] Serpush, F., & Keyvanpour, M. (2014). QEA: a new systematic and comprehensive classification of query expansion approaches. Journal of Computer & Robotics, 7(1), 1-17.

[57] Chen, Z., Zhu, Q., Soh, Y. C., & Zhang, L. (2017). Robust human activity recognition using smartphone sensors via CT-PCA and online SVM. IEEE transactions on industrial informatics, 13(6), 3070-3080.

[58] Erdaş, Ç. B., & Güney, S. (2021). Human activity recognition by using different deep learning approaches for wearable sensors. Neural Processing Letters, 53(3), 1795-1809.

[59] Tran, D. N., & Phan, D. D. (2016, January). Human activities recognition in android smartphone using support vector machine. In 2016 7th international conference on intelligent systems, modelling and simulation (isms) (pp. 64-68). IEEE.

[60] Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. D. R., & Roggen, D. (2013). The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, 34(15), 2033-2042.

[61] Li, F., Shirahama, K., Nisar, M. A., Köping, L., & Grzegorzek, M. (2018). Comparison of feature learning methods for human activity recognition using wearable

sensors. Sensors, 18(2), 679.

[62] Tuncer, T., & Ertam, F. (2021). Novel tent pooling based human activity recognition approach. Multimedia Tools and Applications, 80(3), 4639-4653.

[63] Zebin, T., Scully, P. J., Peek, N., Casson, A. J., & Ozanyan, K. B. (2019). Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition. IEEE Access, 7, 133509-133520.

[64] Yao, R., Lin, G., Shi, Q., & Ranasinghe, D. C. (2018). Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. Pattern Recognition, 78, 252-266.

[65] Esmaeili, V., Mohassel Feghhi, M., & Shahdi, S. O. (2022). Automatic Micro-Expression Recognition using LBP-SIPl and FR-CNN. AUT Journal of Modeling and Simulation, 54(1), 5-5.

[66] Shojaedini, S. V., Morabbi, S., & Keyvanpour, M. (2018). A new method for detecting p300 signals by using deep learning: hyperparameter tuning in high-dimensional space by minimizing nonconvex error function. Journal of medical signals and sensors, 8(4), 205.

[67] Wang, L., & Liu, R. (2020). Human activity recognition based on wearable sensor using hierarchical deep LSTM networks. Circuits, Systems, and Signal Processing, 39(2), 837-856.

[68] Keyvanpour, M., & Serpush, F. (2019). ESLMT: a new clustering method for biomedical document retrieval. Biomedical Engineering/Biomedizinische Technik, 64(6), 729-741.

[69] Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., & Zhang, J. (2014, November). Convolutional neural networks for human activity recognition using mobile sensors. In 6th international conference on mobile computing, applications and services (pp. 197-205). IEEE.

[70] Guo, Y., Wu, Z., & Ji, Y. (2017, August). A hybrid deep representation learning model for time series classification and prediction. In 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM) (pp. 226-231). IEEE.

HOW TO CITE THIS ARTICLE

F. Serpush, M. R. Keyvanpour, M. B. Menhaj, Remote elderly healthcare: a robust deep learning approach for wearable sensors-based complex activities recognition, AUT J. Model. Simul., 55(1) (2023) 109-126.

DOI: 10.22060/miscj.2023.21984.5308