



## IoT System Based on parameter optimization of Deep Learning using Genetic Algorithm

Salwa O. Slim<sup>1\*</sup>      Marwa M. A. Elfattah<sup>1</sup>      Ayman Atia<sup>1,2</sup>  
 Mostafa-Sami M. Mostafa<sup>1</sup>

<sup>1</sup>*Department of Computer Science, HCI-LAB, Faculty of Computers and Artificial Intelligent,  
 Helwan University, Cairo, Egypt*

<sup>2</sup>*October University for Modern Sciences and Arts (MSA)*

\* Corresponding author's Email: [salawaosama@fci.helwan.edu.eg](mailto:salawaosama@fci.helwan.edu.eg)

---

**Abstract:** Nowadays, more and more human activity recognition (HAR) tasks are being solved with deep learning techniques because it's high recognition rate. The architectural design of deep learning is a challenge because it has multiple parameters which effect on the result. In this work, we propose a novel method to enhance deep learning architecture by using genetic algorithm and adding new statistical features. Genetic algorithm is utilized as an enhancing method to get the optimal value parameters of deep learning. Also new statistical features are appended to the features that are extracted automatically from CNN technique. Because the spread of the internet and its significance in our life, we developed Internet of Things (IoT) system. Therefore, we evaluated the performance of the proposed method in its system and found satisfactory results. Moreover, the proposed method was trained on two benchmark datasets (WISDM and UCI) and tested on the dataset, which was collected from IoT system. The results showed that the proposed model improved the accuracy up to 93.8% and 86.1% for user-dependent and independent.

**Keywords:** Human activity recognition, Accelerometer, IoT, Genetic algorithm, Deep learning algorithms, IoT System.

---

### 1. Introduction

The deep learning (DL) is an effective machine learning algorithm for extracting the main features from data that has multiple dimensions. Those features are valuable in case of regression and classification problems [1]. Because DL trains neural networks from start to finish, it can extract features of effective value. In addition, deep generative models are more robust than discriminative models, especially in the over-fitting problem [2]. An open problem is improving the quality of deep learning technology to be effective in user-independent concept. We applied deep learning to HAR because it's significant and it is used in many applications [3, 4]. Also, it has many databases that are available on the internet and are trusted. DL models have many hyper-parameters that affect the entire training process. For example, hyper-parameters can be

divided into two categories: Optimizer hyper-parameters, and Model Specific hyper-parameters [5, 6].

1) Optimizer hyper-parameters related more to the training task and optimization such as learning rate, batch size, and epoch's number. Actually, the execution time of model's epochs may take a long time to reach the ideal state, if the learning rate value is smaller than the ideal value. In contrast, if learning rate value is larger than ideal value, the model will exceed the ideal state. The speed of training process and the number of iterations is influenced by the value of batch size. If the bath size is large, it will affect the training time because it needs more arithmetic to multiply the matrices. And that requires more memory for training process. In contrast, a small batch size of the model leads to a lot of the mistakes in training process that is often an advantage as it does not stop the model at the local minimal [5].

2) Model hyper-parameters are related to the model structure such as layers number, hidden units number, filter size, filters number, activation function, and dropout. The layers between the output and input are hidden layers. The precision is influenced by the number of secret units within a layer and how they organized. Under fitting, a smaller number of units can lead. Dropout parameter is regularization technique that is used to avoid overfitting problem. The activation function is responsible for the non-linear concept in the DL model for training nonlinear prediction boundaries. The rectifier activation function is the most popular [6]. Therefore, each hyper-parameter has a main purpose in the DL architecture and its values affect the final result. Our proposed method provides an appropriate solution for selecting hyper-parameter values by using genetic algorithm. Genetic algorithms (GAs) depend on bioinspired factors, such as selection, crossover, and mutation. Also it has the power to find high-quality solutions to search and optimization problems [7]. This paper's objective is adjusting DL model parameters for improving the accuracy of human activities recognition by genetic algorithm. In this paper, Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) are used as DL models.

To examine the impact of the real-time data in the proposed model performance, Internet of Things (IoT) system was developed. IBM Watson IoT platform (IWIP) was used to design IoT system for sending raw data of human's activities to IBM cloud. Based on sensors embedded in a standard smartphone, we collected 3-axes accelerometers' readings for six human activities: lying, sitting, downstairs, up-stairs, walking, and standing. Moreover, we enhanced the performance of IoT system's classifier by appending new statistical features to the features extracted from deep learning technique. The main contributions of this paper are the following:

Propose a novel method for finding optimized hyper-parameters values that are needed for building an accurate deep learning architecture.

Consolidation of local features extracted from CNN technique along with handcrafted features.

Develop HAR-IoT system for evaluating the performance of the proposed method on it.

The paper is arranged as follows. The motivation and previous work of IoT frameworks and deep learning for human activity recognition are presented in section 2. Section 3 gives outlines of proposed HAR IoT framework design. Experimental preparations, results and discussion are illustrated in section 4. At last, Section 5 displays conclusion and future work.

## 2. Motivations and literature review

The sophisticated issue of recognizing human activity has encouraged the various groups of researchers using different techniques. Some used conventional techniques [8-10] such as Support Vector Machine (SVM), Decision tree, etc. S. Slim et al. [11] preferred to use traditional machine learning for recognizing human activities. While, others preferred to use deep learning technique. For example some researchers used deep learning to improve the performance of HAR systems [12-14]. While others focused on enhancing quality of DL by increasing the information in the training phase [15, 16] or by adapting the sensor data [17, 18]. Also M. Zeng et al. [19] preferred to use DL technique after comparing it with traditional machine learning and finding that DL achieved higher accuracy. S. Matsui et al. [20] evaluated DL technique in the user-independent and dependent concepts and their experiments achieved as desired. While T. Hayashi et al. [21] combined two concerns: improving the DL, by using multi-model signals, and then evaluating it in user-independent concept. Therefore, this section is divided into three parts. The first part presents the last works that compare between deep learning and machine learning techniques. As well as which one is better than the other? And why? The second part presents previous works that aim to improve the deep learning technique. The third part presents the previous works of IoT systems.

### 2.1 Deep and machine learning techniques

There have been numerous studies to compare between two techniques [22-24]. For instance, S. Sarbagya et al. [22] compared between machine learning such as k-Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF) and deep learning such as CNN and RNN. They focused on collecting accelerometer data from single and multiple devices. They stated that DT, logistic regression, and multilayer perception (MLP) techniques failed to recognize the climbing stairs activity with high accuracy. Although the KNN is more accurate than other traditional machine learning such as DT and RF, its accuracy can be affected if the model is supplied during the training phase with little information. Their experiments have shown that deep learning such as CNN is a higher effectiveness than machine learning because it has the ability to extract the most important features of the data without needing an expert. In the light of the mentioned, T. Zebin et al. [23] found that deep learning classifier (CNN) achieved a great improvement in the performance of classification compared to machine

learning classifiers (SVM and MLP). However they tried to enhance the accuracy of SVM and MLP by increasing the number of features extracted, this enhancement didn't give the desired results. Also L. Zhang et al. [25] recommended to use deep learning classifiers (CNN) in real-time activity recognition, which automatically extract appropriate features and then execute classification. They used an android smartphone to collect accelerometer data of seven activities: sitting, standing, upstairs, downstairs, and lying. A recent study by S. Slim et al. [24] concluded that most of recent researches tended to use deep learning techniques in order to identify human activities. According to their study, the overall average accuracy of traditional machine learning algorithms is 83.3%, which is less than the average accuracy of deep learning algorithms that can reach to 94.9%, although the number of studies used traditional machine algorithms are more than those used deep learning. In short, recent literature on recognizing human activity strongly suggests the use of deep learning, especially CNN, over traditional machine learning classifiers. A closer look at the literature of using CNN, however, reveals a number of gaps and shortcomings. For example, I. Andrey et al. [26] indicated that CNN has a problem of capturing the global properties of the signal. While R. Yamashita et al. [27] conducted a review of the basic concepts of CNN and its application, they found that it is hard to capture global and local context of data at the same time. We challenged this problem by capturing both the local and global features. In this paper local features are captured by using CNN but the global by appending a new layer in CNN flatten layer. Also, the architecture design of CNN is a challenge because it has multiple parameters which effect on the result, this is what was addressed in the second part.

## 2.2 Parameters optimization

Some authors have driven the further development of deep learning parameter optimization because of challenging problem which arises in deep learning approach is the size of hyper-parameters. For example C. Ronao et al. [28] focused on evaluating DL technique and faced that challenge. While others tried to overcome this challenge by using evolutionary algorithms [29, 30, 31], which are used as a parameter optimization technique for enhancing DL technique. For instance, S. Young et al. [29] proposed a method for optimizing the hyper-parameters of CNN by using genetic algorithm. Nevertheless, they relied on optimizing a specific number of hyper-parameters which are kernel size

and filter number for each of convolution layers. In addition, their work is limited to use image data. B. Qolomany et al. [32] also addressed this problem by using swarm technique to optimize both numbers of neurons and hidden layers for each layer of the deep learning algorithm. And their results reached a good accuracy.

Some of works focused on using genetic algorithm with neural networks in order to optimize the hyper-parameters of shallow networks. Knowing that the size of hyper-parameters of shallow network are not larger than deep networks [30, 32]. E. Cantú-Paz et al [30] performed an empirical evaluation of using genetic algorithm with shallow neural network. Their methodical evaluation of a large type of datasets focused only on small networks (31 hidden units and single hidden layer). A. Fiszlelew et al. [31] proposed a method for training a GA to define the connectivity within a shallow neural network with network performance as the fitness for each population member. They focused on optimizing the structure of connectivity for that network. The experimental results showed that GA was able to create neural networks topologies that, in general work, better than random or fully connected topologies when they learn and classify a new domain specific data. Although studies have been conducted to enhance deep learning hyper-parameters by many authors [29, 30, 31, 32], this enhancement process is limited to images dataset.

As far as we know about prior works which investigated in deep learning parameters optimization and what we have done in this paper:

- Most of the previous research focused on using the genetic algorithm and they got desirable results. Therefore, we used the genetic algorithm in this research.
- All the research focused on improving a small number of parameters, but we addressed this issue by focusing on all parameters.
- All the research focused on applying enhancement process to images datasets and got desired results. But in this paper we used the acceleration data of human activities.

## 3. Applied method

In this section, we illustrate our proposed method and other methods used such as CNN and LISM.

### 3.1 Convolution neural network (CNN)

CNN is a kind of DL technique. It was considered an updated method from multilayer perceptions (MLP). MLP aims to design fully connected network

that links one neuron to all neurons of following layer. CNN performs relatively data pre-processing compared to other traditional machine learning methods. So CNN has a capability to extract the significant features using filters. This means that CNN has a major advantage because it is not dependent on human effort or prior knowledge for designing features. CNN consists of input, hidden, and output layers. The hidden layers have a series of layers that are the following:

- Convolution layer aims to make a network to be deeper with little parameters by decreasing the number of free parameters [33].
- Pooling layer aims to reduce the data dimensions by merging the neuron outputs into one neuron of following layer.
- Flatten layer aims to prepare the data so that dense layer can receive it.
- Dense Layer is a fully connected layers. It acts as same as MLP for labelling the data.

### 3.2 Long short-term memory

Long short-term memory (LSTM) is derived from RNN. Mainly, LISM relies on feedback connections in contrast to ANN. It consists of units and each one is a complex cell. This complex cell has three gates: input, output, and forget gate. Because those gates flow information into the cell and out, the cell can remember the information for a period of time. Therefore it is more suitable to classify a problem based on time series. The most advantage in LISM is a memory unit cell, this cell has the capacity to include a forgotten part of its previous stored memory in addition to the new information [34].

### 3.3 The proposed method

This part illustrates the proposed method, complexity time, and IoT system, which is developed

to evaluate the performance of our proposed method in real live dataset.

#### 3.3.1. IoT system

IoT solutions for recognizing human’s activity are beneficial in several applications [35, 36]. The proposed IoT system aims to collect data in real-time from different subjects for evaluating a novel method based on deep learning. The system uses the smartphone that sends raw data to IoT platform. The Internet is used to transfer sensor data for the subject to cloud storage for ongoing analysis and recognition of this data. There are three main elements:



Figure. 1 IWIP dashboard to visualize acceleration data in real-time

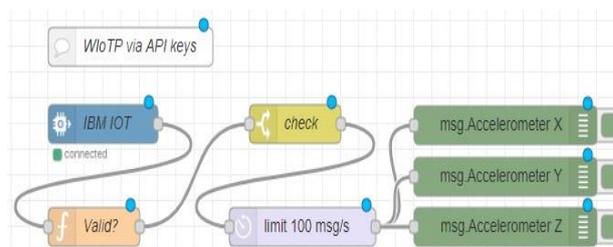


Figure. 2 Data flow process of node-red web application

- Wearable sensor: The collected data, three axes of accelerometer, is sent to a cloud service via WiFi wearable sensor in IoT complaint vision. Smartphone has embedded accelerometer and can be carried conveniently by users and used as IoT sensor.
- Web application: Node.js web application is developed for reading and sending the acceleration data of human activities to the cloud.
- Cloud: It receives raw data from sensors then stores it in the cloudant service (database) for analysis and visualization.

IBM Watson IoT platform (IWIP) was used to send raw data of human’s activities to IBM cloud. Besides, message Queuing Telemetry Transport (MQTT) is a protocol for connecting between smartphone and IWIP. In IWIP, it is applicable to create boards and cards to create and share dashboards that visualize device data in real-time as shown in “Fig. 1”.

Web application is a JSON API that connects smartphone to IBM cloud and Watson IoT Platform. The acceleration data of registered devices is sent to the Watson IoT. We have used Node-Red to write Node.js code as an open source platform. “Fig. 2” shows the flow-code of Node-Red.json and source code in [37]. The first task in Node-Red flow is that takes the input from the IoT device. However the IoT device is the subject’s smartphone that was created and registered using Watson IoT platform services. Then Valid function was created to read the three components of acceleration and acknowledges with true when the acceleration data is correct, or false in

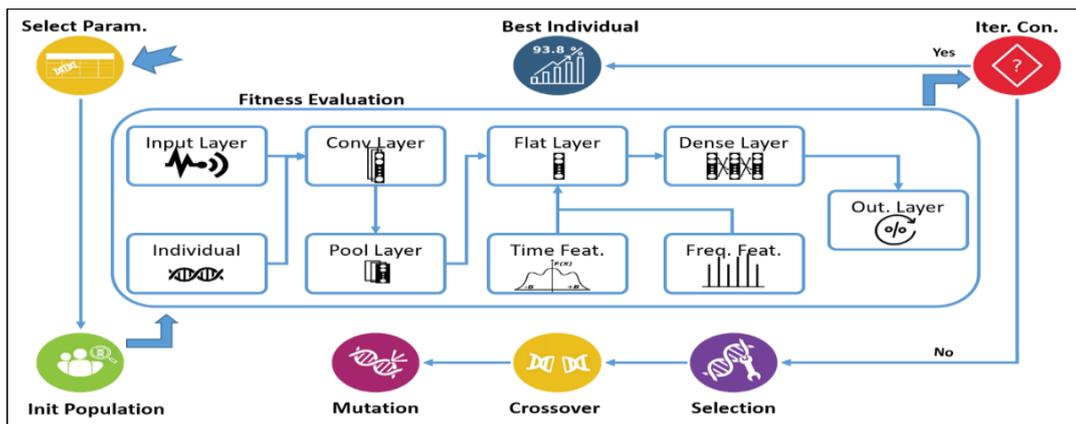


Figure. 3 G-CNN architecture

Table 1. Time and frequency domain features

Time features	Frequency features
<ul style="list-style-type: none"> <li>• Mean</li> <li>• Standard deviation</li> <li>• Median</li> <li>• Largest value</li> <li>• Smallest value</li> <li>• Signal magnitude area</li> <li>• Energy measure</li> <li>• Signal entropy.</li> </ul>	<ul style="list-style-type: none"> <li>• Index of largest magnitude,</li> <li>• Weighted average</li> <li>• Skewness</li> <li>• Kurtosis</li> <li>• Energy of frequency interval within the 64 bins of the FFT.</li> </ul>

a malfunction or error in the registry. Then Check node was used for keeping the data flow when Valid function returned true. Limit node was created to limit the rate of data transmission to the server, the number of limitation is used according to A. Khan et al [38]. Finally, three messages (three components of accelerometers) were sent and saved in Cloudant database.

3.3.2. Classifier methodology

In the classification process, we used a proposed novel method based on deep learning. Our proposed method aims to find optimized hyper-parameters for DL models by using genetic algorithm. We target two models of DL techniques: CNN and LSTM, it is kind of RNN. However, we updated the architecture of CNN by combining some statistical features to CNN flatten layer. This combination is intended to add global features of signal. I. Andrey et al. [26] indicated that CNN has a problem of capturing the global properties of the signal.

While R. Yamashita et al. [27] mention that CNN is hard to capture global and local context of data at the same time. “Table. 1” shows time and frequency features used.

Algorithm 1: The pseudocode of GA with RNN

```

1: Selecting Parameters of LISM for creating individual
2: Generating initial Population
3: Function Fitness Evaluation (population)
   |   3.1: split dataset to train and test datasets
   |   3.2: while 1 to population size do training
   |       LISM algorithm by using training dataset
   |       fitness value= accuracy of LISM algorithm
   |       by using testing dataset
   |       end
   |   end
   |   4: if Critical is satisfied? then
   |       4.1: return Best Individual
   |       4.2: Selection end
   |   else
   |       5: Crossover
   |       6: Mutation
   |   end
Go To Step 3
    
```

R. Patton et al. [39] genetic algorithm structure was used but with different fitness function. Our fitness function aims to get the accuracy of the CNN architecture in testing data. The pseudo code of a genetic algorithm with LSTM is shown in Algorithm 1. Also genetic algorithm with CNN are shown in “Fig. 3”.

Our proposed method is divided into six phases: 1) Selecting the parameters, it aims to determine the form and the size of individuals. The gene in individual is a deep learning parameter, which needs to be optimized. Our proposed method has capability to select all or specific parameters to be optimized. Therefore all CNN parameters, such as input layer size, filter number for every convolution layer, epochs number, convolution layers number, batch size number, learning rate, filter size, and finally the pooling layers number have been prepared to act as genes. Also, this proposed method was applied on RNN. Therefore the parameters of RNN, such as

hidden nodes number, layers number, epoch's number, and finally batch size number, were prepared to act as genes to be optimized.

2) Initial population, it is the second phase in genetic algorithm to generate the parameters which were selected in the previous phase.

3) Fitness evaluation, the fitness value for each individual is calculated by computing the accuracy of the network for testing data after training it. Before calculating fitness, we must first train the network according to Individual values, contains selected parameters to optimize, using training data. The steps of CNN training are shown in "Fig. 3". The first layer in CNN is input layer - that is taken from training dataset. Followed by convolution, pooling, flatten, dense, and the output layers respectively. In this work, we proposed to add some another general handcrafted features to flatten layer such as signal entropy, smallest and largest values, energy measure, standard deviation, signal magnitude area, smallest value, largest value, median absolute deviation, and mean in time domain features. Also, some handcrafted frequency domain features have proposed to be added such as largest magnitude index, energy of 64 frequency bins (it is interval between samples in frequency bins) of each window, skewness, kurtosis, Weighted average to calculate a mean frequency. Those features were used in several researches and introduced acceptable results. Such as I. Andrey et al. [26] used some statically features such as mean, variance, sum of the absolute values and the histogram for improving the accuracy of DL. While Davila et al. aimed to enhance the classification by extracting the kinematics feature such as signal magnitude. Some of researches extracted both time and frequency domain features [10, 40, 41] and their results were acceptable. For example M. Saeed et al. [10] improved the classifier's accuracy with percentage 4%. In time domain they extracted mean, standard deviation, median, correlations, and signal magnitude vector. In frequency domain they extracted energy of each axis in different frequency bands, magnitude of specific range, and weighted average. As well as D. Anguita et al. [41] used the most common features for recognizing human activity. Those features were

mean, standard deviation, median, large and small values, signal magnitude, signal entropy, correclation coefficient, largest frequency component, skewness, kurtosis, energy of frequency interval, and angle between two vectors. S. Slim et al. [24] also worked a survey and stated the most used features, which give desired results, are those mentioned in research D. Anguita et al.[41]. After getting the trained network, the fitness value is calculated by using the testing dataset on the trained network for getting the fitness value or the accuracy.

4) The critical satisfaction is checked, if valid, the best individual is generated and the optimization process will be finished. The critical satisfaction is determined by the number of iterations. If the critical satisfaction is invalid, the selection process will begin. There are various ways for the selection process [7] but roulette wheel technique is selected. The critical satisfaction is checked, if valid, the best individual is generated and the optimization process will be finished. The critical satisfaction is determined by the number of iterations. If the critical satisfaction is invalid, the selection process will begin. There are various ways for the selection process [7] but roulette wheel technique is selected. When an individual's fitness value is high, its choice increases. In crossover phase, the probability of selecting an individual is quoted from J. Jebari et al. [42].

5) Crossover, Parental chromosomes are divided at a randomly specified crossover point when conducting a crossover. Thereafter, by combining 1st part of parent1 to 2nd part of the parent2, a new child genotype is formed. We used a single point of crossover. 6) Mutating is the last phase. The permutation representation is used in this work, therefore it is preferred to use point mutation [7, 42]. The point mutation is executed by randomly choosing gene in the individual and then mutating it.

The mutation of the gene is executed by selecting a random value in specific range. The range values of CNN parameters according to many surveys [24, 43] in HAR field are displayed in "Table. 2". Following to mutation process the fitness evaluation phase is executed again according to new population

**3.3.3. Time complexity**

In our proposed model, GA and CNN are combined so that we calculate the complexity for each one. GAs runs in iterations. Initially, a set of solutions *S* are generated randomly (*S* is called a population). The costs of the solutions of *S* are computed. Some operations are done over the solutions in each iteration such as crossover, mutation, selection, and fitness function. The best *K*

Table 2. The mutation range values for CNN parameters

Parameters	Start	End
Segment Size	150	200
Filters Number	100	200
Layers Number	1	10
Epochs	100	200
Batch size	100	250

solutions in  $S$  are kept and we continue as previous. After the last iteration, the best solution is assigned. Therefore the time cost of an iteration depends on the inner operations (e.g. crossovers, mutation, finding best  $k$  distinct solutions, generating random solutions, calculating cost of the solutions of  $S$ ). The time cost can be deduced using the formula of GA complexity [44]:

$$\text{Time Complexity of GA} = O(S \times I \times O(F) \times ((p_c \times O(C)) + (p_m \times O(M))) \quad (1)$$

Here  $I$  is iteration number,  $F$  is the fitness function,  $P_c$  is probability of crossover,  $P_m$  is probability of mutation,  $C$  is crossover function, and finally  $M$  is mutation function. Besides, the execution time of a GA depends on the number of iterations [45] we will ignore the cost of iteration number because it is fixed in our model. If we ignore the complexity of mutation and crossover function because they are constant, that really simplifies GA complexity to:

$$\text{Time complexity of GA} = O(S \times O(F)) \quad (2)$$

Therefore our genetic method depends mainly on the cost of fitness function which was illustrated in Fig. 3 for CNN. All convolution layers time complexity is calculated by [46]:

$$\text{Time complexity of CNN} = O(\sum_{l=1}^d n_{l-1} \times f_l^2 \times n_l \times m_l^2) \quad (3)$$

Here  $l$  is a convolution layer index, and  $d$  is a convolution layers number (depth).  $n_l$  is a width (filters number) of  $l$ -th layer.  $n_{l-1}$  is used to determine input channels numbers of the  $l$ -th layer.  $f_l$  is the filter length.  $m_l$  is output feature map size. Note this time complexity, though with a different scale, refers to both training and testing time. The training is approximately three times testing time (forward propagation takes one and backward propagation takes two) [46]. This time of genetic and CNN is executed in the training phase after that we don't need this time in testing phase. Therefore we can conclude the time complexity of our proposed method in the Eq. (4).

$$\text{Time Complexity of Proposed Method} = O(S \times O(\sum_{l=1}^d n_{l-1} \times f_l^2 \times n_l \times m_l^2)) \quad (4)$$

## 4. Datasets

Our experiments were conducted on two public Datasets, WISDM [47] and UCI [41], which were

used for training the proposed model. Moreover, we undertake the empirical evaluation using live dataset collected in IoT system.

### 4.1 WISDM dataset

Smartphone was used for collecting the acceleration and gyroscope sensor data. The dataset was recorded from twenty-nine volunteer subjects. The accelerometer data every 50ms was collected for the subject's activities: walking, sitting, jogging, ascending, standing, and descending [47].

### 4.2 UCI dataset

Samsung Galaxy S II smartphone was used for collecting six activities daily life (Walk, Sit, Stand, Lie, walk and downstairs, walk and upstairs). UCI dataset [41] recorded accelerometer and gyroscope sensors data with sampling rate 50 Hz for 30 subjects.

### 4.3 Live dataset

With the aim of testing our model from a substantive point of view, we have to get a live dataset. Hence, a web application was created for collecting the data. Rather than constructing a mobile application, creating a web application may be a more appropriate way. To begin with, web application helps volunteers to collect the data in a comfortable way. The volunteers who have portable can get into web application by means of the internet browser and it'll adjust to whichever gadget the volunteer is viewing them on. Besides, web application isn't local to a specific framework, and ought not to be downloaded or installed just navigate to the URL [48]. Also constructing application is functional and not expensive. Browsing the web application means a new device. That creates it simpler to urge get to a more prominent sum of data. In addition, it roles a fundamental stage for our future work of constructing a HAR framework. We obtained 6 typical activities, including lie, upstairs, downstairs, sit, and stand in a normal environment from 6 healthy subjects. For the purpose of unifying the Data gathering, the subjects have to use a mobile placed in front pants leg pocket. Also three axes accelerometer was collected from android system. Because of the operating system, it is difficult to stabilize the sampling rate of the accelerometers at specific value. According to our knowledge, 100Hz sampling rate is an acceptable value. A. Khan et al [38] found that 100Hz is the optimal sampling rate in a massive comparison that collected seven public and huge datasets.

The segmentation process was applied to split all acceleration signals to 2 seconds with overlap 50%.

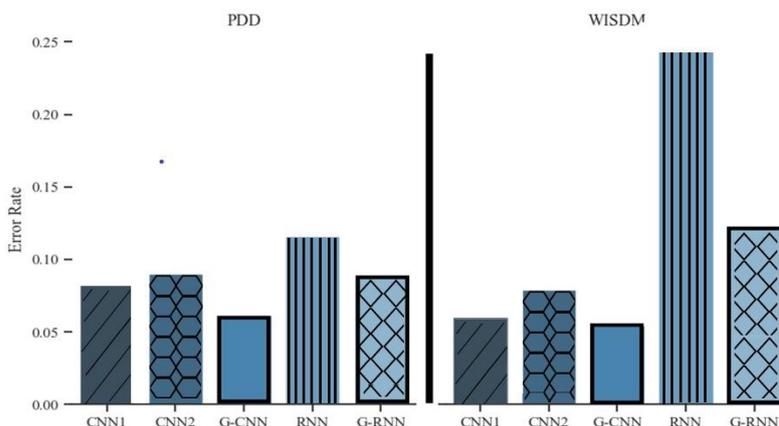


Figure. 4 Comparative error rate of our proposed method with other methods

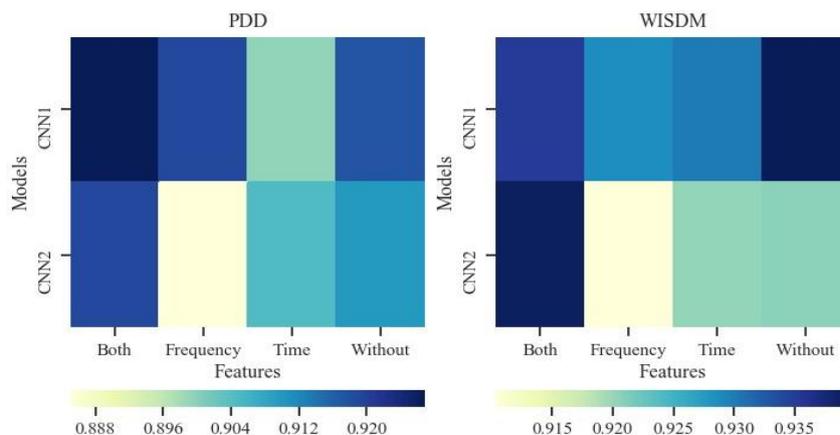


Figure. 5 Comparative accuracy of combining time or/and frequency features with CNN extracted features

Because 100Hz sampling rate is used, every sample is a matrix with the size of 200x3 (2 seconds \* sampling rate \* 3D acceleration). In noise reduction phase, 20 Hz cutoff frequency was applied for both 3rd order low-pass butter worth filter and a median filter. Just like David et. al performed in UCI dataset [41] and achieved a high accuracy.

### 5. Experimental results and analysis

In order to test optimization parameters method, genetic algorithm, and editing CNN flatten layer of our proposed method, we have conducted two experiments. The first experiment objective is to show the ability of genetic algorithm for finding a good DL architecture when it is compared with different shallow deep learning. The second experiment objective is to compare between time and frequency statistical features, that were combined with CNN flatten layer. In order to see how external data affects the proposed model, we performed the third experience. The experiments are completely based on using genetic with CNN with 2s segment size and for both CNN1 and CNN2, 1.28S segment size was used according to the module of H. Nweke et al [43]. The first and the second experiment used

both UCI and WISDM datasets with percentages 80%, 20% as training, validation data respectively. In the third experiment, UCI is used as a training dataset and tested the model by live dataset.

#### 5.1 Optimal parameters

This experiment was performed to prove the capability of GA for enhancing the performance of deep learning architecture. The proposed method was compared with two different architectures for shallow CNN [43]. As well as genetic algorithm is used to optimize RNN parameters. RNN with genetic algorithm was evaluated by comparing it with LSTM-RNN architectures.

“Table. 3” displays the parameters of the two shallow CNN architectures. “Table. 4” displays the

Table 3. Two different CNN architectures values

Parameters	CNN1	CNN2
Input Size	128	128
No filters	100000	100000
Batch size	200	200
Filter size	12	16
No layers	4	1

Table 4. The parameters value of RNN architectures

Parameters	Values
No units	32
Learning rate	0.0025
Lambda loss amount	0.0015
Batch size	1500
No iterations	300
Input size	128

properties of LSTM-RNN which was used.

Fig. 4 shows the error rate of Genetic with CNN (G-CNN), two different shallow CNN architectures, CNN1 and CNN2, Genetic with RNN (R-CNN), and shallow RNN architecture.

According to this comparison results, we found that error rate of proposed model is less than CNN1 and CNN2 to both UCI and WISDM. As well as genetic with RNN (G-RNN) error rate is less than RNN for two datasets. Therefore the outcomes indicate that GA was able to get optimized output for CNN and RNN parameters. Besides, results are analyzed from different points of view such as the number of layers affects efficiency of CNN [49].

As G. Montavon et al [50] illustrated that when CNN layers number increases, the accuracy increases because more layers aid to extract more features. However there is limitation to increase the layers number more than a certain point, it tends to overfit the data. Therefore CNN1 error rate is less than CNN2 because the number of layers for CNN1 is more than CNN2. Also, G-CNN accuracy is better than the others (CNN1, CNN2, RNN, and G-RNN) and the different architectures of CNN (CNN1, CNN2, G-CNN) are better than the different architectures of RNN (RNN, G-RNN) and that was proved by the experiments of N. Hammerla et al. [51].

Moreover, we found that GA obtained the optimal value of segment size was 2 second. This result is matched with study of O. Baños et al [52]. How the segment size affects the efficiency of HAR systems has been studied By O. Baños et. al. From their assessment, the interval 1-2s provides the finest trade-off between recognition accuracy and speed. Because our genetic algorithm focused on finding the highest accuracy regardless of time, it found two seconds is the optimal segment size value. To investigate the performance of G-CNN throughout genetic iterations, the best and average individual accuracy of each iteration were presented in "Fig. 6". We found that the accuracy is almost constant after finding the best individual accuracy. Therefore the experiment in twenty two iterations was stopped. However, the individuals' accuracy is random in the first iterations and the best individual is found in

iteration eighteen and thirteen to G-CNN and G-RNN respectively. We can summarize our proposed method contributes over others [14, 29, 31-32], as two reasons. The first: GA algorithm is able to find the best values of hyper-parameters that improved the accuracy of the proposed method than in study [14]. The second: we focused on improving the values of the most of hyper-parameters and the proposed method is not limited to a specific number of hyper-parameters such as [29, 31-32].

## 5.2 Statistical features

To prove how the adding statistical features to flatten layer affects CNN performance, it is required to find the most appropriate features for both time and frequency domains.

For this purpose, we investigated the most public features that were used in many researches and achieved high results as mentioned before in section methodology. Those features were presented in "Table. 1". To show significantly how those features affect CNN accuracy, we compared time and frequency statistical features. The results of this comparison are shown in "Fig. 5". Even though deep learning has a great power to extract the features automatically [43], we appended handcrafted features to CNN flatten layer. Besides, our results show a significant improvement in CNN accuracy. That's consistent with the results of I. Andrey et al. [26]. They were capable of getting a good CNN accuracy when they stacked the output of flatten layer with additional statistical features. Those features are variance, mean, histogram of data channel and sum of absolute values. Our results show a significant improvement in using features of both time and frequency. As well as the least accuracy when using frequency features in both UCI and WISDM datasets. On the opposite side, when we combined both domain features in WISDM dataset, we found that CNN1 without combining any features accuracy is better than when combining both domain features. Because CNN1 layers number is large and that means the huge features were extracted, no need to combine

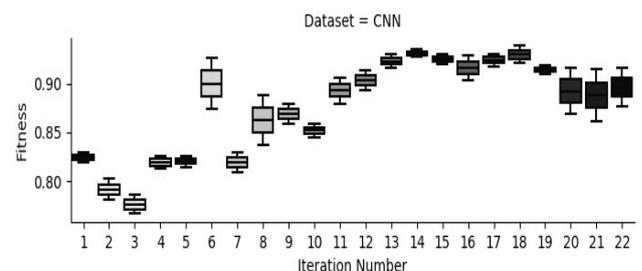


Figure. 6 Best and average individuals' accuracy for each G-CNN and G-RNN iterations.

more features in CNN flatten layer to increase the accuracy. In short, when we added a new layer to CNN, we could improve the accuracy. Because this layer can extract global features which is a drawback of CNN.

### 5.3 User-independent

The main purpose of this experiment was to test our proposed model on a live dataset that does not depend on the training data (User-Independent). This experiment is based on trained models (G-CNN, CNN1 and CNN2) that depend on UCI dataset in the previous experiments.

To investigate the purpose of this experiment, we compared G-CNN with CNN1 and CNN2, the results are shown in “Fig. 7”.

We can conclude that our experiments results show three significant different point of views:

- GA was able to find the optimized values of CNN. Because G-CNN accuracy is better than CNN1, CNN2 in both user dependent and independent approaches. In independent real dataset, CNN1, CNN2 and G-CNN accuracy is approximately 83%, 82%, and 86% respectively. Otherwise, the accuracy of CNN1, CNN2, G-CNN in the dependent dataset are 92%, 91%, 94%.
- CNN1 is better than CNN2 in both user dependent and independent approaches. That is similar to deduction of M. Sebastian et al. [18]. While they compared CNN with two layers and CNN with three layers by using public dataset (PAMP2 [53]) and their dataset, they found that the accuracy of CNN with large number of layers is better than the other in both datasets.
- All three techniques in user dependent are better than in user independent approach because the three techniques have never been trained in the subjects' activities before. That's consistent with the several experiments [15, 54-55].

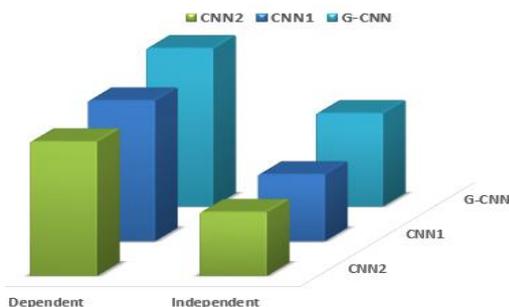


Figure. 7 Dependent and independent accuracy for G-CNN, CNN1, CNN2

For further clarification, H. Ponce et al [54] compared 19 techniques in case of user dependent and independent, and found that their average accuracy was 93.3% and 80.9%, respectively. In the same study, L. Bai et al [55] compared C4.5 and random forest techniques with their proposed model based on deep learning and they found that those techniques accuracy is higher in dependent than independent dataset with range 2.2%, 2.8%, and 14% respectively. Further study about user independent, some of researches used leave-one-subject-out (L1O) technique such as B. Almaslukh et. al [15]. C. Almaslukh et al. collected ADL activities (walk, jump, lie, sit, downstairs, upstairs, stand and run) from 15 volunteers using smartphones (Samsung Galaxy S4). They used CNN techniques for comparing the accuracy between subject-dependent and subject-independent and found that subject-dependent accuracy is higher than subject-independent with range 7%.

To analyze the outcome in more detail, “Fig. 8” displays the confusion matrix for CNN1 and our proposed model. After a closer analytical look, we found that around the diagonal, that includes the true positive, false predictions are approaching in CNN1 and G-CNN. According to confusion matrix results, our model has a significant improvement in the accuracy of lying and walking activities reach to 93.3 % and 96.6% respectively. In a similar study, M. Sebastian et. al [18] found that lying and walking activities are the highest accuracy up to 83% and 84% respectively even though their percentage are less than ours. In contrast, Y. Liang et al [56] found that the lowest accuracy of activities is walking.

However, there are trends in many studies to suggest that there are two problems in activities confusions [12, 16-19, 56-57]: Group1 (walking, up and downstairs) and Group2 (sit, stand). For more explanation, experiments results of J. Lee et al. [57] showed that 6% of ascending activity was incorrectly classified as walking but 5% of walking activity was incorrectly classified as ascending stairs. As well as A. Murad et. al [12] used UCI dataset for evaluating RNN and they found that there is misclassification between Group1 activities and Group2. It is not surprising that G-CNN and CNN are once again struggling to recognize a sitting from the standing activity. Many researchers [12, 16, 17, 18] contend that this misclassification could be a common issue.

One of the core strengths in our proposed model is its ability to classify between walking and down stairs activities with accepted accuracy however, it is a common misclassification problem of Group1 activities. On the contrary, our proposed model failed to classify distinctly between standing and sitting

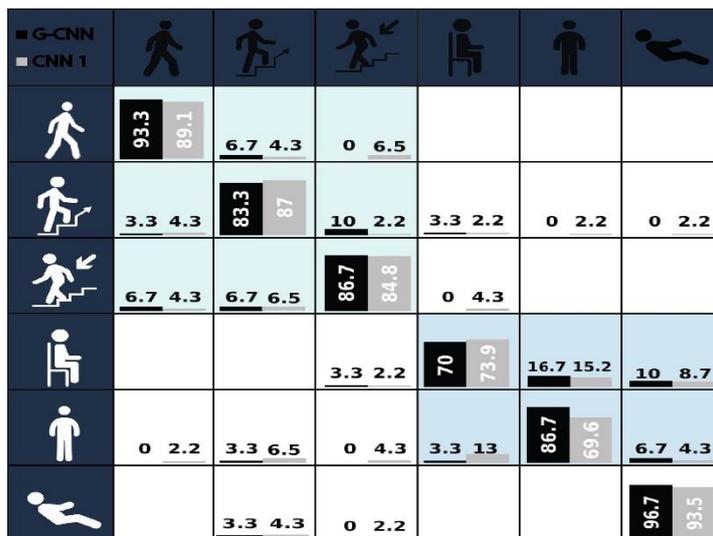


Figure. 8 Confusion matrix of CNN1 and G-CNN of independent real dataset

Table 5. Comparison between our proposed model and other researches

Ref	Algorithm	Accuracy	Dataset	Testing Techn.
*[13]	CNN	94.7%	UCI[41]	Dependent
<b>Our Method</b>	<b>G-CNN</b>	<b>94.5%</b>	<b>WISDM[47]</b>	<b>Dependent</b>
<b>Our Method</b>	<b>G-CNN</b>	<b>93.8%</b>	<b>UCI[41]</b>	<b>Dependent</b>
[14]	CNN	93.8%	Their dataset[14]	Dependent
[15]	CNN	93.5%	RealWorldHAR[58]	Dependent
[54]	Nearest Shrunken Centroids	93.3%	Turk[59]	Dependent
[60]	Hierarchical Continuous HMM	93.18%	UCI[41]	Dependent
[61]	Stacked Autoencoders + SVM	92.16%	UCI[41]	Dependent
[57]	SVM	92.1%	Their Dataset[57]	Dependent
[61]	PCA-SVM	91.82%	UCI[41]	Dependent
[62]	HMM	91.76%	UCI[41]	Dependent
[55]	RandomForest	91.54%	HHAR[63]	Dependent
[54]	Model Averaged ANN	91%	Turk[59]	Dependent
[19]	CNN	90.88%	Actitracker[64]	Dependent
[40]	RandomForest	90.3%	Their Dataset[40]	Dependent
[55]	KNN	90.21%	HHAR[63]	Dependent
[28]	CNN	90%	UCI[41]	Dependent
[65]	DTW	89%	UCI[41]	Dependent
[66]	Handcrafted features + SVM	89%	UCI[41]	Dependent
[55]	RandomForest	88.75%	Their Dataset[55]	Independent
[16]	CNN	88.56%	PAMAP[53]	Dependent
[20]	CNN	88.2%	Their Dataset[20]	Dependent
[19]	PCA-ECDF	87.85%	Actitracker[64]	Dependent
[17]	CNN	87.82%	UCI[41]	Dependent
[55]	RNN	87%	HHAR[63]	Dependent
[55]	C4.5	86.91%	HHAR[63]	Dependent
[67]	Convolutional LSTM	86.78%	PAMAP[53]	Dependent
[54]	AdaBoost	86.6%	Turk(L1O)[59]	Independent
[19]	CNN	86.19%	Skoda[68]	Dependent
<b>Our Method</b>	<b>G-CNN</b>	<b>86.1%</b>	<b>Our Dataset</b>	<b>Independent</b>
[18]	CNN	85.5 %	PAMAP[53]	Dependent
[69]	Handcrafted features + Dropout	85.36%	WISDM[47]	Independent

[19]	log-linear Markov Random Field	85.21	Actitracker[64]	Dependent
[54]	Multivariate Adaptive Regression	85.15%	Turk(L1O)[59]	Independent
[70]	CNN	85.1%	OPP[71]	Dependent
[15]	CNN	85%	RealWorldHAR(L1O)[58]	Independent
[55]	C4.5	84.63%	Their Dataset[55]	Independent
[13]	CNN_EF	84.57%	PAMAP[53]	Dependent
[19]	PCA-ECDF	83.78%	Skoda[68]	Dependent
[72]	HMM	83.51%	UCI[41]	Dependent
[20]	CNN	83.5%	Their Dataset[20]	Independent
[40]	KNN	83.5%	Their Dataset[40]	Dependent
[96]	Handcrafted features + RF	83.46%	WISDM[47]	Independent
* [26]	CNN	82.76%	UCI[41]& WISDM[47]	Independent
[54]	SVM with Radial Basis Function	81.33%	Turk(L1O)[59]	Independent
[54]	Rule-Based Classifier	81.23%	Turk(L1O)[59]	Independent
[73]	SVM	80.4%	OPP[71]	Dependent
[54]	C4.5-Decision Trees	80.07%	Turk(L1O)[59]	Independent
[70]	MV	79.5%	OPP[71]	Dependent
[54]	Naive Bayes	79.06%	Turk(L1O)[59]	Independent
[40]	Naive Bayes	79%	Their Dataset[40]	Dependent
[69]	Handcrafted features + RF	77.81%	UCI[41]	Independent
[70]	DBN	77.8%	OPP[71]	Dependent
[69]	Handcrafted features + Dropout	76.26%	UCI[41]	Independent
[8]	SVM	76.04%	OPP[71]	Dependent
[55]	KNN	75.73%	Their Dataset[55]	Independent
[19]	PCA-ECDF	75.63%	OPP[71]	Dependent
[19]	log-linear Markov Random Field	75.36%	Skoda[68]	Dependent
[54]	Model Averaged ANN	75.04%	Turk(L1O)[59]	Independent
[18]	CNN	75%	Their Dataset[18]	Independent
[54]	k-Nearest Neighbors	74.91%	Turk(L1O)[59]	Independent
[19]	CNN	74.9%	OPP[71]	Dependent
[19]	log-linear Markov Random Field	73.58%	OPP[71]	Dependent
[55]	RNN	73%	Their Dataset[55]	Independent
[73]	INN	72.3%	OPP[71]	Dependent
[40]	SVM	69.9%	Their Dataset[40]	Dependent
[18]	RF	64%	Their Dataset[18]	independent
[54]	Artificial Neural Networks	61.38%	Turk(L1O)[59]	Independent
[26]	BasicFeatures+RF	46.5 %	UCI[41]& WISDM[47]	Independent
[26]	KNN	38.47%	UCI[41]& WISDM[47]	Independent
[26]	PCA+RF	38.2 %	UCI[41]& WISDM[47]	Independent

activities because those activities have a high variance in z-axis and our model is based on three axes acceleration. Therefore, we think the quality can be enhanced by appending additional features especially for Z-axis to CNN flatten layer.

### 5.4 Comparison with similar methods

The main focus of the experiment was to prove the effectiveness of our proposed model by comparing it with a few existing strategies, “Table. 5” displays the results. To investigate CNN accuracy statistically we calculated the average of CNN accuracy of user dependent and independent in “Table. 5” researches. Thus, we found that CNN’s

Table 6. The comparison between our and I. Andrey et. al study [26]

Property	Our	Andrey
Training	UCI	WISDM
Testing	Live	UCI
Dependency	independent	independent
Technique	G-CNN	CNN
Accuracy	86.1%	82.76%

average accuracy is 81.69% and 77.33 to user-dependent and independent and that is matched with the second experiment (User-Independent). For more analysis, we arrange the accuracy of this researches

in descending order and found that our proposed model is the second one. Because of C. Ronao et. al [13] (market by star in “Table. 5”) trained CNN on a larger dataset, their CNN accuracy is the highest. There is a similarity between our study and other researches for example the study of I. Andrey et. al [26] (market by star in “Table. 5”). Table. 6” shows the comparison between our study and Andrey et. al study. However our proposed model obtained higher accuracy than Andrey et. al.

## 6. Conclusion

Identifying people’s activities is a huge challenge. Therefore, we focused on improving deep learning performance to be more powerful for classifying human activity data in real time. A genetic algorithm was used for optimizing CNN and RNN hyper-parameters. As well as some handcrafted features were combined with features, that were extracted automatically by deep learning technique. The proposed model was evaluated by using two datasets: public dataset (UCI) and real dataset. We trained and tested the proposed model by using UCI and real dataset respectively. We developed IoT system to collect live dataset for evaluating the proposed method in user independent dataset. Our experiments have shown that the accuracy is enhanced for both user-dependent and independent with percentage 93.8% and 86.1% respectively. Future research should be devoted to the development of HAR-IoT system to identify the humans ‘activities in real-time. Besides, we are going to solve to enhance the proposed classification method by appending Z-axis features on CNN flatten layer for solving the problem activities of upstairs and downstairs confusion.

## References

- [1] R. Salakhutdinov, “Learning deep generative models”, *Annual Review of Statistics and Its Application*, Vol. 2, No. 1, pp.361–385, 2015.
- [2] G. Hinton, “A practical guide to training restricted Boltzmann machines”, *In Neural Networks: Tricks of the Trade*, Springer, pp. 599–619, 2012.
- [3] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciarri, M. Mordonini and I. De Munari, “IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment”, *IEEE Internet of Things Journal*, Vol. 6, No. 5, pp. 8553-8562, 2019.
- [4] J. Lee and J. Kim, “Energy-Efficient Real-Time Human Activity Recognition on Smart Mobile Devices”, *Advanced Technologies for Mobile IoT and Cyber-Physical Systems*, 2016.
- [5] T. Yu and H. Zhu, “Hyper-Parameter Optimization: A Review of Algorithms and Applications”, *ArXiv*, 2020.
- [6] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization”, In: *Proc. of the International Conf. on Neural Information Processing Systems*, NY, USA, pp. 2546–2554, 2011.
- [7] M. Mitchell, “An Introduction to Genetic Algorithms”, *Cambridge, MA: MIT Press*, ISBN 9780585030944, 1996.
- [8] J. Davila, A. Cretu, and M. Zaremba, “Wearable Sensor Data Classification for Human Activity Recognition Based on an Iterative Learning Framework”, *Sensors*, Vol. 17, No. 6, pp. 1287, 2017.
- [9] Consortium Publications. Available online: <http://www.opportunity-project.eu/publications> (accessed on 1 June 2017).
- [10] M. Saeed, J. Pietilä, and I. Korhonen, “An Activity Recognition Framework Deploying the Random Forest Classifier and A Single Optical Heart Rate Monitoring and Triaxial Accelerometer Wrist-Band”, *Sensors (Basel, Switzerland)*, Vol. 18, No. 2, 2018.
- [11] S. Slim, A. Atia, and M. M. Mostafa, “An Experimental Comparison between Seven Classification Algorithms for Activity Recognition”, In: *Proc. of the 1st International Conf. of Springer on Advanced Intelligent System and Informatics*, Beni Suef, Egypt, pp. 37-46, 2015.
- [12] A. Murad and J. Pyun, “Deep Recurrent Neural Networks for Human Activity Recognition”, *Sensors*, Vol. 17, No. 11, 2017.
- [13] C. Ronao and S. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks”, *Expert Systems with Applications*, Vol. 59, pp. 235-244, 2016.
- [14] Y. Chen and Y. Xue, “A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer”, In: *Proc. of IEEE International Conf. on Systems, Man, and Cybernetics, SMC*, pp. 1488–1492, 2015.
- [15] B. Almaslukh, AM. Artoli, and J. Al-Muhtadi, “A Robust Deep Learning Approach for Position-Independent SmartphoneBased Human Activity Recognition”, *Sensors*, Vol. 18, No. 11, pp.3726, 2018.
- [16] P. Kasnesis, C. Patrikakis and I. Venieris, “PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion”, In: *Proc. of*

- SAI Intelligent Systems Conf.*, Springer, Cham, pp. 101–19, 2018.
- [17] X. Han, J. Ye, J. Luo and H. Zhou, “The Effect of Axis-Wise Triaxial Acceleration Data Fusion in CNN-Based Human Activity Recognition”, In: *Proc. of IEICE Transactions Conf. on Information and Systems*, Vol. E103.D, No. 4, pp. 813–24, 2020.
- [18] M. Sebastian, S. Philip, R. Attila, H. Michael, S. Rainer, and D. Robert, “CNN-based sensor fusion techniques for multimodal human activity recognition”, In: *Proc. of ACM International Symposium Conf. on Wearable Computers*, pp. 158-165, 2017.
- [19] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, Zhu J, P. Wu, and J. Zhang, “Convolutional Neural Networks for human activity recognition using mobile sensors”, In: *Proc. of IEEE International Conf. on Mobile Computing, Applications and Services*, pp. 197-205, 2014.
- [20] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, “User adaptation of convolutional neural network for human activity recognition”, In: *Proc. of European Signal Processing Conf. (EUSIPCO), IEEE, Kos*, pp. 753-757, 2017.
- [21] T. Hayashi, M. Nishida, N. Kitaoka, T. Toda, and K. Takeda, “Daily Activity Recognition with Large-Scaled Real Life Recording Datasets Based on Deep Neural Network Using Multi-Modal Signals”, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 101, No. 1, pp. 199-210, 2018.
- [22] S. Sarbagya, Z. Chaoyang, and Z. Zhaoxian, “Comparative Study of Machine Learning and Deep Learning Architecture for Human Activity Recognition Using Accelerometer Data”, *International Journal of Machine Learning and Computing*, Vol. 8, No. 6, pp. 577-582, 2018.
- [23] T. Zebin, P. J. Scully, and K. B. Ozanyan, “Human activity recognition with inertial sensors using a deep learning approach”, In: *Proc. of IEEE Conf. on Sensors*, Orlando, FL, pp. 1-3, 2016.
- [24] S. Slim, A. Atia, M. Elfattah, and M. M. Mostafa, “Survey on Human Activity Recognition based on Acceleration Data”, *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 10, No. 3, 2019.
- [25] L. Zhang, X. Wu, and D. Luo, “Real-time activity recognition on smartphones using deep neural networks”, In: *Proc. of 2015 IEEE 12th Intl Conf. on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf. on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf. on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pp. 1236-1242, 2015.
- [26] Andrey, “Real-time human activity recognition from accelerometer data using Convolutional Neural Networks”, *Applied Soft Computing*, Vol. 62, pp. 915-922, 2018.
- [27] R. Yamashita, M. Nishio, R. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology”, *Insights Imaging*, Vol. 9, No. 4, pp. 611–629, 2018.
- [28] C. Ronao and S. Cho, “Evaluation of Deep Convolutional Neural Network Architectures for Human Activity Recognition with Smartphone Sensors”, *Yonsei University*, pp. 858-860, 2015.
- [29] S. Young, D. Rose, T. Karnowski, S. Lim, and R. Patton, “Optimizing deep learning hyperparameters through an evolutionary algorithm”, In: *Proc. of the Workshop Conf. on Machine Learning in High-Performance Computing Environments*, pp. 1-5, 2015.
- [30] E. Cantú-Paz and C. Kamath, “An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 35, No. 5, pp. 915–927, 2005.
- [31] A. Fiszlelew, P. Britos, A. Ochoa, H. Merlino, E. Fernández, and R. García-Martínez, “Finding optimal neural network architecture using genetic algorithms”, *Advances in Computer Science and Engineering Research in Computing Science*, Vol. 27, pp. 15–24, 2007.
- [32] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta, and D. Benhaddou, “Parameters optimization of deep learning models using Particle swarm optimization”, In: *Proc. of International Wireless Communications and Mobile Computing Conf. (IWCMC), IEEE, Valencia*, pp. 1285-1290, 2017.
- [33] H. Aghdam and E. Heravi, “Guide to convolutional neural networks: a practical application to traffic-sign detection and classification”, *New York, NY: Springer*, Vol. 10, pp. 978-973, 2017.
- [34] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [35] C. Rodriguez, D. Castr, W. Coral, J. Cabra, N. Velasquez, J. Colorado, D. Mendez, and L. Trujillo, “IoT system for Human Activity Recognition using BioHarness 3 and Smartphone”, In: *Proc. of International Conf. on Future Networks and Distributed Systems*

- (*ICFNDS*), Cambridge, United Kingdom, pp. 40-62, 2017.
- [36] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb, "IoT based mobile healthcare system for human activity recognition", In: *Proc of Learning and Technology Conf.*, Jeddah, pp. 29-34, 2018.
- [37] Source Code Available on: <https://github.com/SalwaOsama/Genetic-With-DeepLearning>.
- [38] A. Khan, N. Hammerla, S. Mellor, and T. Plötz, "Optimising Sampling Rates for Accelerometer-Based Human Activity Recognition", *Pattern Recognition Letters, Elsevier BV*, Vol. 73, pp. 33-40, 2016.
- [39] R. Patton, B. Beckerman, and T. Potok, "Analysis and Classification of Mammography Reports Using Maximum Variation Sampling", In *Genetic and Evolutionary Computation: Medical Applications*, pp. 112-131, 2010.
- [40] L. Nhac, T. Nguyen, T. Ngo, T. Nguyen, and H. Nguyen, "Mobile Online Activity Recognition System Based on Smartphone Sensors", In: *Proc. Of International Conf. on Advances in Information and Communication Technology*, pp. 357-366, 2017.
- [41] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones", In: *Esann*. Bruges, Belgium, Vol. 3, pp. 24-26, 2013.
- [42] Jebari and M. Madiafi, "Selection Methods for Genetic Algorithms", *International Journal of Emerging Science*, Vol. 3, No. 4, pp. 333-344, 2013.
- [43] H. Nweke, Y. Teh, M. Al-Garadi, and U. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges", *Expert Systems with Applications*, Vol. 105, pp. 233-261, 2018.
- [44] Z. Yuri, F. Elena, and C. Dmitry, "Two-step classification method based on genetic algorithm for bankruptcy forecasting", *Expert Systems with Applications*, Vol. 88, pp. 393-401, 2017.
- [45] P. Oliveto, J. He, and X. Yao, "Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results", *International Journal of Automation and Computing*, Vol. 4, No. 3, pp. 281-293, 2007.
- [46] H. Kaiming and S. Jian, "Convolutional Neural Networks at Constrained Time Cost", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5353-5360, 2014.
- [47] R. Kwapisz, G. Weiss, and S. Moore, "Activity Recognition using Cell Phone Accelerometers", In: *Proc. of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, Washington DC, Vol. 12, No. 2, 2010.
- [48] S. Slim, <https://iot-solution-tutorial-wise-serval.eugb.mybluemix.net/>, (accessed on Jan 2020).
- [49] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning", *arXiv preprint arXiv:1611.02167*, 2016.
- [50] G. Montavon, M. Braun, and K. Müller, "Kernel Analysis of Deep Networks", *Journal of Machine Learning Research (JMLR)*, Vol. 12, No. 9, 2011.
- [51] N. Hammerla, S. Halloran, and T. Plötz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables", *arXiv preprint arXiv:1604.08880*, 2016.
- [52] O. Banos, J. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window Size Impact in Human Activity Recognition", *Sensors Basel*, Switzerland, Vol. 14, No. 4, pp. 6474-6499, 2014.
- [53] Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring", In: *Proc. of IEEE 16th Int. Symposium on Wearable Computers*, pp. 108-109, 2012.
- [54] H. Ponce, L. Miralles-Pechuán, and M. Martínez-Villaseñor, "A Flexible Approach for Human Activity Recognition Using Artificial Hydrocarbon Networks", *Sensors*, Vol. 16, No. 11, p. 1715, 2016.
- [55] Bai, C. Yeung, C. Efstratiou, and M. Chikomo, "Motion2Vector: unsupervised learning in human activity recognition using wrist-sensing data", In: *Proc. of the 2019 ACM International Joint Conf. on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 537-542, 2019.
- [56] Y. Liang, X. Zhou, Z. Yu, and B. Guo, "Energy-Efficient Motion Related Activity Recognition on Mobile Devices for Pervasive Healthcare", *Mobile Networks and Applications*, Vol. 19, No. 3, pp. 303-317, 2014.
- [57] J. Lee and J. Kim, "Energy-Efficient Real-Time Human Activity Recognition on Smart Mobile Devices", *Mobile Information Systems*, 2016.
- [58] T. Szttyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition", In: *Proc. of the 2016 IEEE International Conf. on Pervasive Computing*

- and Communications*, Sydney, Australia, pp. 1–9, 2016.
- [59] Barshan and M. Yükses, “Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units”, *The Computer Journal*, Vol. 57, No. 11, pp. 1649–1667, 2014.
- [60] Ronao and S. Cho, “Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models”, *International Journal of Distributed Sensor Networks*, Vol. 13, No. 1, 2017.
- [61] Y. Li, D. Shi, B. Ding, and D. Liu, “Unsupervised feature learning for human activity recognition using smartphone sensors”, In: *Proc. of Second International Conf. on Mining Intelligence and Knowledge Exploration*, pp. 99–107, 2014.
- [62] Ronao and S. Cho, “Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models”, In: *Proc. of 10<sup>th</sup> International Conf. on Natural Computation*, pp. 681–686, 2014.
- [63] A. Stisen, H. Blunck, S. Bhattacharya, T. Prentow, M. Kjærgaard, A. Dey, T. Sonne, and M. Jensen, “Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition”, In: *Proc. of the 13th ACM Conf. on Embedded Networked Sensor Systems*, ACM, pp. 127–140, 2015.
- [64] J. Lockhart, G. Weiss, J. Xue, S. Gallagher, A. Grosner, and T. Pulickal, “Design considerations for the wisdom smart phone-based sensor mining architecture”, In: *Proc. of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, ACM, pp. 25–33, 2011.
- [65] S. Seto, W. Zhang, and Y. Zhou, “Multivariate time series classification using dynamic time warping template selection for human activity recognition”, In: *IEEE Symposium Series on Computational Intelligence*, pp. 1399–1406, 2015.
- [66] Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine”, In: *International Workshop on Ambient Assisted Living*, pp. 216–223, 2012.
- [67] Ordóñez and D. Roggen, “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”, *Sensors*, Vol. 16, No. 1, pp. 1–25, 2016.
- [68] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, “Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection”, In: *Proc. of European Conf. on Wireless Sensor Networks*, Springer, Berlin, Heidelberg, pp. 17–33, 2008.
- [69] B. Kolosnjaji and C. Eckert, “Neural network-based user independent physical activity recognition for mobile devices”, In: *Proc. of International Conf. on Intelligent Data Engineering and Automated Learning*, Springer, Cham, pp. 378–386, 2015.
- [70] J. Yang, M. Nguyen, P. San, X. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition”, In: *Proc. of the 24th International Joint Conf. on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, pp. 25–31, 2015.
- [71] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Forster, and O. Troster, “Collecting complex activity datasets in highly rich networked sensor environments”, In: *Proc. of Seventh International Conf. of IEEE on Networked Sensing Systems (INSS)*, pp. 233–240, 2010.
- [72] Y. Kim, B. Kang, and D. Kim, “Hidden Markov model ensemble for activity recognition using tri-axis accelerometer”, In: *Proc. of 2015 IEEE International Conf. on Systems, Man, and Cybernetics*, pp. 3036–3041, 2015.
- [73] Cao, M. Nguyen, C. Phua, S. Krishnaswamy, and X. Li., “An integrated framework for human activity classification”, In: *Proc. of ACM International Conf. on Ubiquitous Computing*, pp. 331–340, 2012.