Neuroscience Informatics 2 (2022) 100078

Contents lists available at ScienceDirect

Neuroscience Informatics

www.elsevier.com/locate/neuri

Original article

Deep CNN hyperparameter optimization algorithms for sensor-based human activity recognition

Saeid Raziani[a], Mehran Azimbagirad[b,c,*]

[a] Department of Computer Engineering, Razi University, Kermanshah, Iran
[b] Faculty of Medicine and Health Sciences, University of Western Brittany, Brest, France
[c] Laboratory of Medical Information Processing (LaTIM), UMR1101 INSERM, Brest, France

ARTICLE INFO

Article history:
Received 4 May 2022
Accepted 12 May 2022

Keywords:
Deep learning
Human activity recognition
Convolutional neural network
Metaheuristic algorithms

ABSTRACT

Human activity recognition (HAR) is an active field of research for the classification of human movements and applications in a wide variety of areas such as medical diagnosis, health care systems, elderly care, rehabilitation, surveillance in a smart home, and so on. HAR data are collected from wearable devices which include different types of sensors and/or with the smartphone sensor's aid. In recent years, deep learning algorithms have been showed a significant robustness for classifying human activities on HAR data. In the architecture of such deep learning networks, there are several hyperparameters to control the model efficiency which are mainly set by experiment. In this paper, firstly, we introduced one dimensional Convolutional neural network (CNN) as a model among supervised deep learning for an online HAR data classification. In order to automatically choose the optimum hyperparameters of the CNN model, seven approaches based on metaheuristic algorithms were investigated. The optimization algorithms were evaluated on the HAR dataset from the UCI Machine Learning repository. Furthermore, the performance of the proposed method was compared with several state-of-the-art evolutionary algorithms and other deep learning models. The experimental results showed the robustness of using metaheuristic algorithms to optimize the hyperparameters in CNN.

1. Introduction

Aging is one of main concern of the countries due to the growing population of vulnerable groups such as the elderly or cardiovascular or diabetic patients. Therefore, monitoring the physical health of these people and the amount of energy they consume during the day is crucial. Thus, acquiring information about their physical activities helps the medical centers to screen the level of activity, check their health status and recovery process, prescribe medication accordingly, and totally improving the quality of their life which affects the entire community. Consequently, monitoring and detecting such activities are conducting HAR researches [1,2].

Due to the diversity and complexity of human activities, different methods have been used to collect HAR data. In general, such methods can be divided into two groups; sensors and cameras [3]. Collecting HAR data using camera is very common because of the simplicity of implementation in numerous researches on such data. In this method, the cameras need to be installed in a fixed place and capture video of the user's movements. Several video games and computer applications are constructed using this method [4]. One of the disadvantages of this method is that the camera should be installed in a special place and the user must always be in the field of view of the camera. Another drawback is the cameras cannot be used everywhere, for instance in the bedroom or private spaces which concerns for privacy violations. Furthermore, providing sufficient brightness during the night might increase the cost of recent technique.

In order to overcome the aforementioned problems, sensors are more convenient and lower-cost option than the cameras for HAR data collection. Various sensors are available for this purpose such as gyroscopes, electromyography, magnetometers accelerometer, pressure sensors body and compasses [5]. Also, valuable information about people's lifestyle and a wide range of activities such as walking, sitting, running, lying down can be captured using these sensors, by putting them on special parts of the body [6]. As long as the sensors have been developing, the acquired data need to be interpreted precisely using robust tools such as neural network algorithms.

According to Data Reportal, over than 2.9 billion smartphone and smartwatch users were in 2020 which they all include several

* Corresponding author at: 22 Av Camille Desmoulins, 5th floor, Latim, 29200 Brest, France.
E-mail address: mehran.azimbagirad@univ-brest.fr (M. Azimbagirad).

https://doi.org/10.1016/j.neuri.2022.100078

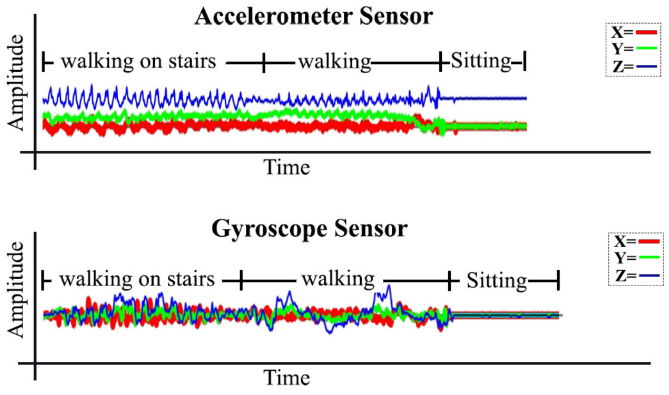

Fig. 1. Accelerometer and Gyroscope sensors in a smartphone (Samsung S7) for three human activities, walking on stairs, normal walking and sitting. X-Y-Z are three directions for signal acquisition.

sensors to collect the motion data such as magnetometer sensors, accelerometers and gyroscopes. They are collecting motion data [7] and transfer them to supporting apps for monitoring the activities of their owners. They offer a great number of advantages including user-friendliness and portability, support for various communication protocols such as WIFI and Bluetooth, convenient and fast processing power, as well as no need for special infrastructure. The aforementioned advantages of smart devices have made them very popular for potential researches [8].

Accelerometer and gyroscope are two most used built-in smartphone sensors. Accelerometer is employed to detect the orientation of the phone while the gyroscope, or gyro for short, adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist (see Fig. 1). The raw signals collected by such sensors usually are pre-processed to clean the noise and artifacts. Later by performing several feature extraction methods, they can be classified using statistical and machine learning models such as Multilayer perceptron, Support vector machine, Decision Tree, Markov models, etc. [9]. Two drawbacks of these traditional methods are the preprocessing steps on the data and also using hand-crafted feature engineering techniques to reduce the feature space. Basically, manual extracting features from sensor data due to the similarity between activities (e.g., sitting or standing) is a difficult and time-consuming process that requires high skill and knowledge [10].

In recent decades, deep learning (DL) algorithms as a subbranch of machine learning have had a successful performance in many fields such as machine vision, natural language processing and speech recognition [11]. A DL such as Convolutional Neural Network (CNN) can significantly reduce the difficulty of selecting appropriate features in traditional methods by automatically extracting abstract features using hidden multi-layers [12–14]. Each DL model has its own learning process settings to learn from the data and improve its performance. These settings are associated to the hyperparameters in DL models which have a huge impact on the training time, cost of calculations and performance of the models [15]. The main problem is how to choose the optimum set of hyperparameters since each hyperparameter has different effect on DL model [16]. A fundamental method to solve this problem is the trial and error decision in which the hyperparameters are empirically selected. On the other hand, in DL architectures, there is a trade-off among the hyperparameters such that changing one may also affect the impact of the others. Given the above conditions, manually searching the optimal set of hyperparameters is very tedious [17].

Recently, some attempts have been done to improve the efficiency of DL models in optimizing their hyperparameters. For example, James and Yoshua [18] showed that randomly chosen trials are more efficient for hyper-parameter optimization than trials on a grid. Very recently, Bacanin et al. [19] introduced Enhanced Swarm Intelligence Metaheuristics to optimize a CNN's hyperparameters for handwritten digits clustering in computer vision. They showed that both proposed improved methods establish higher performance than the other existing techniques in terms of classification accuracy and the use of computational resources.

In this paper, we investigated seven well-known metaheuristic algorithms which can automatically optimize DL hyperparameters by searching the optimum value of each hyperparameter of a CNN on the HAR dataset in the UCI repository [20]. In addition, we compared the results with several other states of the art methods and experimental results to show higher efficiency of the proposed method. Therefore, the contribution of this paper includes:

1- Finding optimal hyperparameter values automatically without human intervention which may result in increasing the efficiency of the DL model. To the best of our knowledge, the proposed method for HAR has not yet been performed.
2- Using seven states of the art metaheuristic algorithms to find the optimal values of CNN hyperparameters and compare the experimental results with CNN, CNN-LSTM, LSTM and ConvLSTM models. Statistical results prove that one of the proposed methods was better than the other models.
3- This study provides a review of the state of the arts approaches for the researchers who are about using DL model on their dataset.

The rest of this paper is dedicated to related works and metaheuristic algorithms, methodology, results and discussion, and conclusion, respectively.

1.1. Related works for HAR classification

According to the benefits of using the build-in sensors of smartphones to collect HAR data mentioned earlier, numerous studies have been done by researchers in the HAR area. One can divide HAR classification approaches into three major areas: Machine learning, deep learning and hybrid models. We review some of them as follows. In the Machine learning studies for instance, Anguita et al. [21] used a multi-class support vector machine (SVM) to classify data collected from smartphone sensors (accelerometer and gyroscope) to recognize human activity and provided the online UCI HAR dataset (see also [22]). One of the most widely used model is Hidden Markov which has been modified in several studies to classify the HAR data. For example, Ronao et al. [23] used a two-stage continuous Hidden Markov to classify human activity using the same sensors.

As the second major area, there are a couple of deep learning models which are used for HAR classification by researchers. CNN, LSTM and Deep Q-Learning are the most significant approaches. For example in CNN model, Cho et al. [24] presented a new 1-D CNN-based method for HAR using a conquer-based classification. In another study, to identify the pattern of fine-grained movements, Zhou et al. [25] proposed a framework for HAR based on semi-supervised DL and auto-labeling model. They used six accelerometer sensors on different part of the body. Xia et al. [26] introduced a combined method of LSTM and convolution layers for activity classification. The idea was that the collected data from the sensors, firstly, goes to the LSTM and then feed the convolution layers. The dataset that they implemented their model on was collected by 5 types of sensors, namely accelerometers, gyroscopes, magnetometers, object sensors, and ambient sensors.

The third group of HAR classification approaches is dedicated to the hybrid schemes. There are several researches in which they combined different models such as CNN with LSTM or Hidden

Markov models to improve the performance of such classifiers. For example, San et al. [27] clustered six different human activities using data collected from accelerometer and smartwatch sensors. In this study, they manually extracted the features and used a CNN model. Then, they used Hidden Markov models, LSTM and machine learning techniques to modeling the time sequence of continuous physical activities. As another hybrid approach, Peng et al. [28] introduced a method called AROMA, which was used to classify simple and complex activities. This model consists of a CNN to detect simple activities and a LSTM for complex activities, in which the CNN output is given to the LSTM input. This model was evaluated using two public datasets which were collected using several sensors such as triaxial accelerometer, IMUs and inertial sensors. Zeng et al. [29] introduced a hybrid method based on CNN-encoder-decoder and CNN-ladder architecture for 3 HAR dataset. The datasets were collected using accelerometer, gyroscope, magnetometer, temperature, heart rate, ECG, etc. Mukherjee et al. [30] used three different EnsemConvNet classification models, i.e., CNN-Net, Encoded-Net and CNN-LSTM to classify human activity. This type of classification performs the clustering process based on majority voting, sum rule, product rule and so on. The datasets in their model were collected using accelerometer, gyroscope, and geomagnetic field sensors. Another example for hybrid approaches is a CNN-LSTM method suggested by Mutegeki et al. [31]. They evaluated the proposed method using two datasets UCI and ISPL. As the last but not least paper, we refer to a hybrid model based on CNN and LSTM to extract Spatio-temporal features of radar data introduced by Zhu et al. [32]. This model was evaluated on seven HAR datasets. This method uses 1-D CNN for extracting spatial features from the spectrograms and a LSTM to lean global time-dependent information.

In the reviewed papers, the used models have several hyperparameters which normally are adjusted empirically. In order to automate the hyperparameter selection, several methods can be used that we review seven among them as follows.

1.2. Metaheuristic algorithms

Unlike other optimization algorithms such as greedy searches, meta-heuristic algorithms randomly explore the search space. This randomness property leads to search more space in less time than other regular optimization algorithms [33]. Moreover, in the search space area that offers promising solutions, they search deeper [34]. Another advantage is that they use the search information which gets in each iteration for the next search. Furthermore, a chance of searching more areas in the search space stems from avoiding falling into the local optimization, which ultimately increases the likelihood of finding a global optimization [35]. Because of the advantages of metaheuristic algorithms in finding optimal solutions, we used seven among them to automatically find the optimal values of CNN hyperparameters. We give a brief description of these algorithms as follows.

1.2.1. Gray wolf optimizer (GWO)

Gray wolf algorithm was firstly proposed in 2014 by Mirjalali et al. [36]. GWO is inspired by the way that gray wolves hunt, eat and live-in groups. GWO is a population-based algorithm to overcome by the optimization problems. A hierarchical order exists within the group such that the alpha wolves at the head (commander-in-chief) of the group, and the other wolves follow them. Beta wolves are in the next hierarchy of the group that helps alpha wolves for decision, and in the absence of alpha wolves they are replaced. After alpha and beta, delta and omega wolves are in the next hierarchy that they have no role in group decision and are only obedient. The wolf hunting process takes place in several stages; tracking, approaching the prey, siege and then the hunt. The optimization process in GWO is led by alpha, beta and delta wolves. While alpha is considered as the best solution, the second and third best solutions are related to beta and delta. Therefore, GWO stores all three best current solutions and the other wolves look for a better solution.

1.2.2. Whale optimization algorithm (WOA)

WOA is another population-based algorithm introduced by Mirjalali and Lewis [37] in 2016, inspired by whale behavior and Bubble-net strategy to solve optimization problems. Since the whales prefer to catch krill or small fish near the surface of the water, they create prey by making circular bubbles and then hunt. Optimization based on this algorithm is performed in three phases; Siege of prey, bubble-net attack method and prey search.

1.2.3. Salp swarm algorithm (SSA)

SSA is a bio-inspired optimizer for engineering design problems released in 2017 [38]. Salp is a type of salpidae that has a clear, barrel-shaped body similar to jellyfishes. Swarming behavior of the salps is the main motivation considered in this algorithm. This behavior results in to create a swarm called a salp chain. The reason for this behavior is not clear, but they seem to do it to achieve better locomotion and foraging. Individuals in this algorithm are divided into two groups, leader and followers. The leader is located in front of the salp, which is responsible for leading the population, and the rest of them are followers. In this algorithm, a food source called F is considered as swarm's target, and the leader updates its position according to the food source. Three coefficients named c1, c2, and c3 are used in updating the leader position. The most important one i.e., c1 is used to balance the exploration and exploitation. c2 and c3 are as two random numbers uniformly distributed in the range [0-1] determine the step size and the next position in the jth dimension. The followers' positions are updated with an equation called Newton's law of motion.

1.2.4. Sine cosine algorithm (SCA)

As another population-based algorithm, SCA [39] searches for the optimal solutions with sine and cosine waves in the solution space. The waves always move towards the best solution and their oscillating behavior causes to examine the search space around the best solution. This algorithm has four main parameters: R1, R2, R3, R4 to control the movements. R1 is a parameter to determine the direction of the next move or the next search area. R2 is to describe the direction movement, how far moves to the destination or far from it. R3 is a weight for the movement step. R4 is a random variable in the range [0,1] for switching between sine and cosine motion.

1.2.5. Multi verse optimizer (MVO)

This method is a metaheuristic algorithm that is able to solve optimization problems inspired by three physics concepts; black hole, white hole and wormhole [40]. This algorithm follows the Big Bang theory in astronomy in which it expands continuously, and creates new universe such that the worlds inside each universe are interconnected and collided. Every universe has an inflation rate that is an important factor in the formation of black holes and white holes, stars and planets and their suitability for life and habitation. Black holes and white holes are considered as exploration while wormholes indicate exploitation. In MVO, each variable in the solution is assumed as an object in that world. Each solution also has an inflation rate that is corresponding with its fitness function values. This algorithm follows a rule for optimization; the higher inflation rate, the higher probability of having a white hole. Therefore, the probability of a black hole is inversely proportional to the rate of inflation such that a higher inflation

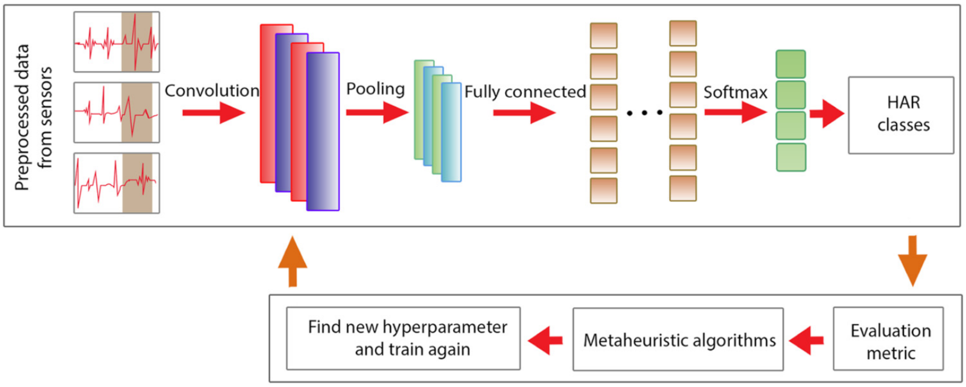

Fig. 2. Overall steps of the purposed model, i.e., a CNN with metaheuristic optimizer of the hyperparameters.

rate can reduce the likelihood of having a black hole. In addition, the worlds with high inflation rates can send the objects from the white hole to the black hole. Then, the worlds with low inflation tend to receive more objects from the black hole. Finally, the objects may go to the best world randomly, regardless of the rate of inflation, by the wormhole.

1.2.6. Particle swarm optimization (PSO)

This optimization algorithm is adapted from the swarm behavior of animals (fish, birds, etc.) [41]. PSO starts with a random solution called particle solution, and then updates it at each iteration. Since these particles are related to each other eventually converge to an optimal solution. The fundamental of this algorithm is based on this concept that each particle is moving at a speed in the search space and also memorizes the best value of its individual position, and shares its best position with other particles. Three values are used in updating each particle such that; the first value is the best value obtained by the particle itself, the second value is the best value obtained by the whole swarm, and the third value is the speed of movement of each particle in the search space. These values suggest the next position of the particles.

1.2.7. Moth flame optimization (MFO)

Moths have a navigation mechanism called transverse orientation to move in a straight line at night which is very efficient when flying long distances [42]. In this mechanism the moth maintains a constant angle with respect to the moon to move in a straight line. A disadvantage of this mechanism is that it works only when the light source is too far away. In addition, this mechanism loses its effectiveness when a moth is exposed to an artificial light source since the moth wants to maintain a constant angle with the light result in a spiral fly around the light source. This algorithm considers the search agents as the moth and the best position found by the moth as the fire. Each moth searches around a flag and updates it if it finds a better position. The moths update their positions using a logarithmic spiral equation. This spiral motion allows the moth to fly around the flames and not necessarily in the space between them. In this way, exploration and exploitation of the search space can be guaranteed.

2. Methodology

In this section, we introduce a new method for HAR classification based on the 1-D CNN model in which the hyperparameters of the model are automatically optimized by metaheuristic algorithms. Fig. 2 shows the overall steps of the proposed method. We describe the proposed model in details as follows.

2.1. CNN models

CNN is one of the most noteworthy DL algorithms which is used in various fields, e.g., computer vision, language recognition as well as classifying dataset. CNN consists of 3 main layers: 1-convolutional layer 2- pooling 3-fully connected layer. Usually, a convolutional layer is used to extract input features. In this layer, several kernels (or filters) are responsible to extract them from the input such that the kernels slide over the input. The greater number of kernels may result in more extracted features from the input. Indeed, the kernel values are multiplied by the input values to generate output which is also called feature map.

After the convolutional layer, an activation layer can be used which is a non-linear layer. The purpose of this layer is to convert linear operations in the convolution layer to nonlinear operations that are applied to the feature map (output of the previous layer). This layer usually uses a RELU function, which is nonlinear and learns faster than older nonlinear functions such as tanh and sigmoid. RELU helps to reduce vanishing gradient.

The second layer is called pooling layer, which can be used after the RELU function. This layer is also known as subsampling. The pooling layer is applied to the entire output generated from the previous step. The purpose of applying this layer is to reduce the input dimensions, reduce network computations and control overfitting.

The third layer is fully connected, which receives input from the previous layers and produces an N dimension vector as output, which N is the number of classes that CNN is supposed to classify. This vector is given to Softmax to perform the classification. Finally, after performing all the above steps, the network can predict an activity when receives input sensor data.

The learning process in CNN is done with backpropagation. When fully connected predicts an output, this value is compared to the actual value (in Supervised learning the target values of the samples are already known) and its error is calculated by a loss function. Loss functions vary according to the application of deep networks, but for multiclassification, the loss function is usually categorical cross-entropy. There are several optimizers for the loss function such as gradient descent family, Stochastic optimizers, adaptive learning rate method, etc. We used Adam optimizer in our model [43].

After calculating the network error, update the weight values with the backpropagate of error in the network, the gradient descent algorithm updates them by calculating the value of gradient error relative to the network weights. Indeed, weight values are the same as kernel values. This operation is repeated until the amount of error reaches its minimum and more iteration does not reduce the error anymore.

2.2. Hyperparameters in CNN

As it was mentioned before, searching for the optimum hyperparameters in a deep neural network is a challenge. Moreover, finding hyperparameters automatically is crucial since it does not require expertise and experience, and finding the optimal hyperparameter values improves the performance of DL. In order to find the optimal values of hyperparameters, the models can be viewed from the perspective of optimization problems. There are several algorithms to solve this optimization problem. One class of optimizers for such problems is metaheuristic algorithms. This study is aimed to use metaheuristic algorithms to find the optimal values of hyperparameters in a CNN model.

2.3. Proposed approach

In this section, we introduce a new method for HAR classification. This method is based on a 1-D CNN model. The model has several important hyperparameters that have a great impact on its performance. We use the metaheuristic algorithms described in the previous section to optimize the hyperparameters of the model to achieve a high accurate prediction for the six different HAR activities in a dataset. The selected hyperparameters for the CNN model are Pooling size, Kernel size, Number of filters, Number of epochs, and Batch size. Therefore, metaheuristic algorithms must find the optimal values of these 5 hyperparameters. The metaheuristic algorithms start with a random solution of hyperparameters, which is an N-dimension vector the same size of the hyperparameters of the DL model (here is a 5-D vector). We perform a loop in which the metaheuristic algorithms try to optimize the initial random vector at each iteration using the cost function provided by the CNN model (train data, performs the classification for test data, and calculate the loss as cost function for optimization). Metaheuristic algorithms find the optimum solution in a continuous space [0,1] which should be translated as CNN hyperparameters. For example, using the following equation, we convert the continuous values of the GWO algorithm to discrete values as hyperparameters and then send it to the model for next training step of the CNN (see supplementary materials).

In order to know how the current solution is close to the optimal solutions in each iteration, a fitness function is used to show the performance of the metaheuristic algorithms. Since the aim of metaheuristic algorithms is to decrease the fitness function, we used 1-accuracy as the fitness function. As long as the algorithms reduce the fitness function, they increase the accuracy of the CNN model. Consequently, in each iteration in the loop, the model run using new hyperparameters, the calculated fitness function value for classifying the test data is given to the metaheuristic algorithms as the cost function to update the next solution of the hyperparameters accordingly.

2.4. Data collection

In this study, a public database called UCI HAR is used to evaluate the proposed method and compare the metaheuristic algorithms. This dataset is collected by a smartphone (Samsung Galaxy S II) 30 subjects in the age range of 19 to 48 years are asked to wear a smartphone to their waist and perform 6 predefined activities including WALKING, WALKING UPSTAIRS (WU), WALKING DOWNSTAIRS (WD), SITTING, STANDING, and LAYING. This data was collected by a tri-axial accelerometer sensor and a gyroscope sensor on a smartphone at a sampling rate of 50 Hz. After preprocessing, the raw data are manually labeled. The total number of data is 10299 vectors, each vector contains 561 attributes. The data is divided into two parts: train and test data include 7352 (∼71%) and 2947 (∼29%) vectors, respectively. Table 1 provides more detailed information about this dataset.

Table 1
HAR dataset details used in this study.

Type of activity	Activities	Train	Test	Total
Static	Sitting	1286	491	1777
	Standing	1374	532	1906
	Laying	1407	537	1944
Dynamic	Walking	1226	496	1722
	WU	1037	471	1508
	WD	1022	420	1442
	Total	7352	2947	10299

WU=Walking Upstairs, WD= Walking Downstairs

Table 2
Control parameter used for the experiments in metaheuristic algorithms; PSO, MVO, GWO, MFO, WOA and SCA.

Algorithm	Parameter	Value
PSO	Acceleration constants	[2.1, 2.1]
	Inertia weights	[0.9, 0.6]
MVO	Wormhole existence probability	[0.2, 1]
GWO	Random and adaptive vector with linearly decrease	[0-2]
MFO	Convergence constant with linearly decrease	[-1, -2]
WOA	Constant to define spiral	1
SCA	A constant in changed adaptively equation	1
SSA	No parameters	-

Table 3
List of hyperparameters' intervals to perform in the loops for method evaluation.

Hyperparameters	Value
Batch size	[10-100]
Number of epochs	[1-200]
Number of filters	[1-400]
Kernel size	[1-20]
Pooling size	[1-20]

2.5. Experimental settings

In order to define and run the proposed method and the algorithms, we employed Python and Keras library with Tensor Flow backend and a cluster including a system with the hardware configuration of RAM 11 GB and GTX-1080ti GPU architecture Turing (chip TU102) with 4352 cores. Each metaheuristic algorithm has several parameters that must be set before the run. Table 2 shows the parameters of each algorithm with the set value for them.

The maximum number of iterations and search agents of each algorithm are set at 20 and 25, respectively. These values are selected by trial and error in such a way that an acceptable answer can be reached in a short time. In order to achieve reliable and fair results, each experiment (algorithm) has been performed 10 times independently, and the results of the experiments have been reported as the average obtained by different models.

We specified the lower and upper bound in the search space for each hyperparameter in Table 3, restricting the search agents to find the optimal value for each hyperparameter according to previous literature [44].

3. Results and discussion

In this section, the experimental results of the proposed method using seven metaheuristic algorithms including PSO, GWO,

Table 4
The optimal set hyperparameters for CNN based on metaheuristic algorithm for HAR dataset.

Algorithms	filters	Kernel	epochs	Batch	Pooling
GWO	310	4	50	16	20
MFO	180	6	200	51	20
MVO	258	6	176	43	15
PSO	218	5	122	38	19
SCA	128	5	58	19	19
SSA	200	13	132	34	10
WOA	200	20	200	64	20

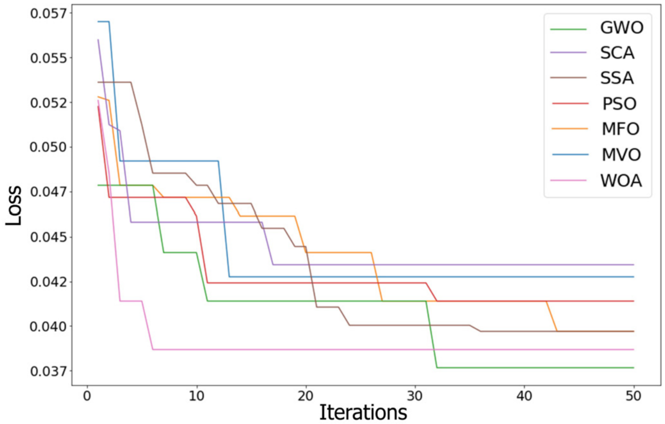

Fig. 3. Convergence profiles of the proposed model on HAR dataset classification for 50 iterations using 7 metaheuristic optimizers. Loss= (1-accuracy) or fitness.

MFO, MVO, SSA, SCA and WOA and four well-known DL models including LSTM, CNN_LSTM, CLSTM and CNN on UCI HAR database were compared. We evaluated them using several classification metrics such as Accuracy, Precision, Recall, and F1-score.

Table SI in supplementary materials shows the average of accuracy, precision, recall and f1-score for all models on the training dataset. The statistical results of this table prove that the models that use metaheuristic algorithms to optimize hyperparameters are more efficient than the models that their hyperparameters are manually selected based on the previous literature. Furthermore, in this table, the accuracy of Laying activities is higher than other activities. Standing and sitting also have the lowest results. Moreover, the last column of the table shows the average of each metric. According to this table, the GWO_CNN, although it has higher AVG accuracy than all the other models, the MVO_CNN has better results in other AVG metrics. Table SII in supplementary materials also shows the confusion matrix of the proposed method for each of the metaheuristic algorithms by each activity on UCI HAR dataset based on the highest accuracy. The main diagonal of each matrix is the number of activities that the model has classified them correctly and the rest are wrong classified activities. In the last column of this table, accuracy values are reported separately for each activity. According to Table 3, we determined the lower and upper boundary of search space for metaheuristic algorithms. Thus, the optimum results obtained after 50 iterations for each model are presented in Table 4 are within the boundary of specified search space. Table 3 shows the optimal values of hyperparameters obtained for the CNN model using the proposed method, based on the implemented metaheuristic algorithms on HAR dataset. Since the fitness function (Loss) in this paper was considered as 1-accuracy, metaheuristic algorithms update the set of hyperparameters in each iteration to reduce the value of the fitness function.

Fig. 3 shows convergence curves of seven metaheuristic algorithms over 50 iterations based on a defined fitness function.

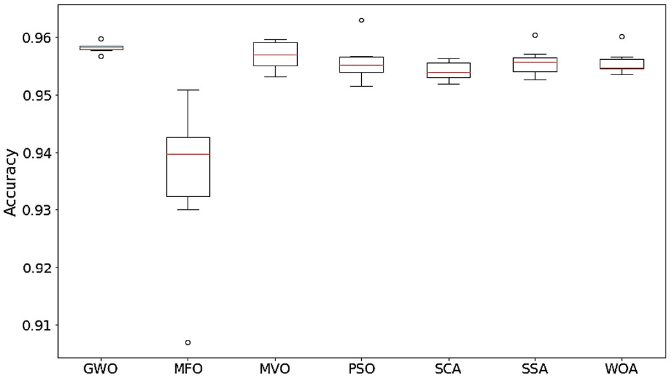

Fig. 4. Boxplot of purposed model based on accuracy metric.

In this figure, the convergence trend of the algorithms shows that each algorithm did not fall in local optima results in reducing the classification error. That is because of getting better during 50 iterations (except for WOA which is not getting better after the 7th iteration). Fig. 4 shows the accuracy boxplot diagram for 10 independents runs of all metaheuristic algorithms. Each boxplot shows the first and third quarters, the middle, and the maximum and minimum values. The length of the rectangle in the boxplot shows the range of variation between the first and third quarters. The outlier values are shown in this figure with a hollow black circle. The smaller rectangle, the shorter whiskers, and the lack of outlier value for each algorithm indicate less scatter of the results and higher reliability of the models.

Fig. SI in Supplementary Materials shows the violin diagrams for each hyperparameter, with each subplot showing the values obtained by each metaheuristic algorithm in 10 independent iterations for each hyperparameter. The thicker points of the diagrams indicate that the optimal values of each hyperparameter are around those points. In other words, this hyperparameter may result in the CNN model to perform more accurate classification for those values. For example, the GWO has obtained the optimal value for the Number of epochs 50.

In summary, the experimental results showed that the proposed method can reduce the classification error by finding optimum values of hyperparameters without the need for prior knowledge and experience of the problem and increase significantly the efficiency of the model. Moreover, metaheuristic algorithms can explore the search space of several hyperparameters together and use the results of previous searches in subsequent searches, so they can establish a trade-off between the values of the hyperparameters which allows to increasing the computational speed and decrease calculations.

The metaheuristic algorithms introduced in this paper search the optimum hyperparameters without checking all combinations possible like grid search. Thus, in training step of DL models, they considerably save time and energy. Also, such algorithms guarantee the convergence unlike random selection of the hyper parameters. In addition, metaheuristic algorithms have memory and search smartly to decrease the loss function.

Since the model can predict the human activities (with an acceptable accuracy), it is possible to implement it on cell phone to process data from either smartwatch or phones, monitoring each activity as future step. Also, we would like to investigate the possibility of detect wrong positioning in sitting, walking and laying which leads several injuries of knee, hip and vertebral column.

4. Conclusions

Human activity recognition (HAR) has become one of the important research fields due to its numerous applications, especially in healthcare systems. Researchers use a wide variety of methods to classify such activities. DL methods due to their robustness and flexibility for almost all classification approaches have attracted a huge attention. DL models have several hyperparameters that affect the network performance. Since there is no specific method for setting them, the values of hyperparameters are mainly adjusted based on trial and error. In this paper, we propose a class of methods for automatically adjust the hyperparameters of 1-D CNN for HAR classification, optimally. The proposed method using metaheuristic algorithms with random search in the search space of each hyperparameter can finally find the optimal set of hyperparameters that increase the efficiency of CNN. The results showed that the proposed method has a higher average than other methods in classification metrics. Since the random values of weights and biases at the beginning of the learning process may fall in local optima and results in low convergence speed in DL or machine learning algorithms, as future step we examine the method to avoid this drawback.

Human and animal rights

The authors declare that the work described has been carried out in accordance with the Declaration of Helsinki of the World Medical Association revised in 2013 for experiments involving humans as well as in accordance with the EU Directive 2010/63/EU for animal experiments.

Funding

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

Author contributions

All authors attest that they meet the current International Committee of Medical Journal Editors (ICMJE) criteria for Authorship.

Declaration of competing interest

The authors declare that they have no known competing financial or personal relationships that could be viewed as influencing the work reported in this paper.

Acknowledgements

The authors would like to thank Jean-Francois Clement, technical manager in IBRBS Institut Brestois de Recherche en Bio-Sante, Faculty of Medicine and Health Sciences, University of Western Brittany, Brest, France.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neuri.2022.100078.

References

[1] F. Khan, S.K. Leem, S.H. Cho, Human–computer interaction using radio sensor for people with severe disability, Sens. Actuators A, Phys. 282 (2018) 39–54, https://doi.org/10.1016/j.sna.2018.08.051.

[2] A. Subasi, K. Khateeb, T. Brahimi, A. Sarirete, Human activity recognition using machine learning methods in a smart healthcare environment, in: M.D. Lytras, A. Sarirete (Eds.), Innovation in Health Informatics, Academic Press, 2020, pp. 123–144, Chapter 5.

[3] Y. Wang, S. Cang, H. Yu, A survey on wearable sensor modality centred human activity recognition in health care, Expert Syst. Appl. 137 (2019) 167–190, https://doi.org/10.1016/j.eswa.2019.04.057.

[4] L. Onofri, P. Soda, M. Pechenizkiy, G. Iannello, A survey on using domain and contextual knowledge for human activity recognition in video streams, Expert Syst. Appl. 63 (2016) 97–111, https://doi.org/10.1016/j.eswa.2016.06.011.

[5] M. Mario, Human activity recognition based on single sensor square HV acceleration images and convolutional neural networks, IEEE Sens. J. 19 (2019) 1487–1498, https://doi.org/10.1109/JSEN.2018.2882943.

[6] M. Uddin, S. Syed-Abdul, Data analytics and applications of the wearable sensors in healthcare: an overview, Sensors 20 (2020) 1379, https://doi.org/10.3390/s20051379.

[7] H. Zawar, Q.Z. Sheng, W.E. Zhang, A review and categorization of techniques on device-free human activity recognition, J. Netw. Comput. Appl. 167 (2020) 102738, https://doi.org/10.1016/j.jnca.2020.102738.

[8] G. Yuan, Z. Wang, F. Meng, Q. Yan, S. Xia, An overview of human activity recognition based on smartphone, Sens. Rev. 39 (2019) 288–306, https://doi.org/10.1108/SR-11-2017-0245.

[9] A. Rasekh, C.-A. Chen, Y. Lu, Human activity recognition using smartphone, preprint, arXiv:14018212, 2014.

[10] J.B. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 3995–4001.

[11] Z. Wang, J. Chen, S.C.H. Hoi, Deep learning for image super-resolution: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 3365–3387, https://doi.org/10.1109/TPAMI.2020.2982166.

[12] A. Chattopadhyay, M. Maitra, MRI-based brain tumour image detection using CNN based deep learning method, Neuroscience Informatics 2 (2022) 100060, https://doi.org/10.1016/j.neuri.2022.100060.

[13] G. Kaur, R. Sinha, P.K. Tiwari, S.K. Yadav, P. Pandey, R. Raj, et al., Face mask recognition system using CNN model, Neuro Informatics 2 (2022) 100035, https://doi.org/10.1016/j.neuri.2021.100035.

[14] S.D. Pande, P.P. Jadhav, R. Joshi, A.D. Sawant, V. Muddebihalkar, S. Rathod, et al., Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support, Neuro Informatics 2 (2022) 100016, https://doi.org/10.1016/j.neuri.2021.100016.

[15] K. Shankar, Y. Zhang, Y. Liu, L. Wu, C.H. Chen, Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification, IEEE Access 8 (2020) 118164–118173, https://doi.org/10.1109/ACCESS.2020.3005152.

[16] A. Zela, A. Klein, S. Falkner, F. Hutter, Towards automated deep learning: efficient joint neural architecture and hyperparameter search, preprint, arXiv:180706906, 2018.

[17] M. Feurer, F. Hutter, Hyperparameter optimization, in: F. Hutter, L. Kotthoff, J. Vanschoren (Eds.), Automated Machine Learning: Methods, Systems, Challenges, Springer International Publishing, Cham, 2019, pp. 3–33.

[18] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012).

[19] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba, Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics, Algorithms 13 (2020) 67.

[20] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, Esann 3 (2013) 3.

[21] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in: International Workshop on Ambient Assisted Living, Springer, 2012, pp. 216–223.

[22] A. Jain, V. Kanhangad, Human activity classification in smartphones using accelerometer and gyroscope sensors, IEEE Sens. J. 18 (2018) 1169–1177, https://doi.org/10.1109/JSEN.2017.2782492.

[23] C.A. Ronao, S.-B. Cho, Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models, in: 2014 10th International Conference on Natural Computation (ICNC), IEEE, 2014, pp. 681–686.

[24] H. Cho, S.M. Yoon, Divide and conquer-based 1D CNN human activity recognition using test data sharpening, Sensors 18 (2018), https://doi.org/10.3390/s18041055.

[25] X. Zhou, W. Liang, K.I.K. Wang, H. Wang, L.T. Yang, Q. Jin, Deep-learning-enhanced human activity recognition for Internet of healthcare things, IEEE Int. Things J. 7 (2020) 6429–6438, https://doi.org/10.1109/JIOT.2020.2985082.

[26] K. Xia, J. Huang, H. Wang, LSTM-CNN architecture for human activity recognition, IEEE Access 8 (2020) 56855–56866, https://doi.org/10.1109/ACCESS.2020.2982225.

[27] R. San-Segundo, H. Blunck, J. Moreno-Pimentel, A. Stisen, M. Gil-Martín, Robust human activity recognition using smartwatches and smartphones, Eng. Appl. Artif. Intell. 72 (2018) 190–202, https://doi.org/10.1016/j.engappai.2018.04.002.

[28] L. Peng, L. Chen, Z. Ye, Y. Zhang, AROMA: a deep multi-task learning based simple and complex human activity recognition method using wearable sensors, Proc. ACM Interact Mob. Wearable Ubiquitous Technol. 2 (2018) 74, https://doi.org/10.1145/3214277.

[29] M. Zeng, T. Yu, X. Wang, L.T. Nguyen, O.J. Mengshoel, I. Lane, Semi-supervised convolutional neural networks for human activity recognition, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 522–529.

[30] D. Mukherjee, R. Mondal, P.K. Singh, R. Sarkar, D. Bhattacharjee, EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications, Multimed. Tools Appl. 79 (2020) 31663–31690, https://doi.org/10.1007/s11042-020-09537-7.

[31] R. Mutegeki, D.S. Han, A CNN-LSTM approach to human activity recognition, in: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, 2020, pp. 362–366.

[32] J. Zhu, H. Chen, W. Ye, A hybrid CNN–LSTM network for the classification of human activities based on micro-Doppler radar, IEEE Access 8 (2020) 24713–24720, https://doi.org/10.1109/ACCESS.2020.2971064.

[33] N. Manavizadeh, H. Farrokhi-Asl, P. Beiraghdar, Using a metaheuristic algorithm for solving a home health care routing and scheduling problem, Proj. Manag. J. 5 (2020) 27–40.

[34] M. Abdel-Basset, L. Abdel-Fatah, A.K. Sangaiah, Metaheuristic algorithms: a comprehensive review, in: A.K. Sangaiah, M. Sheng, Z. Zhang (Eds.), Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, Academic Press, 2018, pp. 185–231, Chapter 10.

[35] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (2019) 2191–2233, https://doi.org/10.1007/s10462-017-9605-z.

[36] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Adv. Eng. Softw. 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.

[37] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.

[38] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191, https://doi.org/10.1016/j.advengsoft.2017.07.002.

[39] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133, https://doi.org/10.1016/j.knosys.2015.12.022.

[40] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2016) 495–513, https://doi.org/10.1007/s00521-015-1870-7.

[41] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.

[42] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249, https://doi.org/10.1016/j.knosys.2015.07.006.

[43] D.P. Kingma, J. Ba, Adam: a Method for Stochastic Optimization, 2015.

[44] W. Zhu, W. Yeh, J. Chen, D. Chen, A. Li, Y. Lin, Evolutionary convolutional neural networks using ABC, in: Proceedings of the 2019 11th International Conference on Machine Learning and Computing, 2019, pp. 156–162.