A Neuro Phenotypic Evolution Algorithm for Recognizing Human Motion Type

Ivan Stepanyan
Mechanical Engineering Research Institute of the Russian Academy of Sciences, Russian Federation
Department of Mechanics and Control Processes, Academy of Engineering, Peoples' Friendship University of Russia named after Patrice Lumumba, Russian Federation
neurocomp.pro@gmail.com

Safa Hameed
Department of Mechanics and Control Processes, Academy of Engineering, Peoples' Friendship University of Russia named after Patrice Lumumba, Russian Federation
sa.programmer1@gmail.com

Abstract: *Living in the modern world requires having a precise, intelligent system that may be suggested to do various activities. Due to the scalability of AI algorithms, we have proposed a phenotypic Evolutionary Algorithm (EA)-based system to assist the Artificial Neural Network algorithm (ANN) in the learning process. Combining the two strategies can result in a smart neuro-evolutionary model that is effective in accomplishing significant duties in various domains. The suggested multi-layer neural algorithm's design creates the conditions for learning via the EA's processes of crossover, mutation, and selection. To aid in the selection and crossover processes, the learning process phase breaks up the original ANN into multiple ANNs according to the number of hidden layers. ANNs are ranked from worst to best in the selection phase based on the soring function that is applied to the fitness list. The fitness list retains each ANN's accuracy even after breaking apart the original ANN. The crossover procedure is then applied between the two worst and best ANNs. Mutation provides a means of improvement for the less effective ANNs. Following completion of these processes, the ANN algorithms are combined to create a single ANN algorithm. The Vicon mobile robot (SCITOS G5) system's multi-dimensional data, which extracted both aggressive and typical human movement, as well as Human Activity Recognition (HAR) datasets extracted by smartphones, both have been applied using the suggested method. The system achieved a high performance and efficiency rate on the intended recognition problem.*

Keywords: *Breaking up processes, combining processes, multi-dimensional data, optimization, phenotypic neuro-evolutionary.*

Received January 5, 2024; accepted September 30, 2024
https://doi.org/10.34028/iajit/21/6/6

1. Introduction

With the rapid evolution of life, the necessity to propose and design an intelligent computational method to forecast solutions has become increasingly crucial. The Artificial Neural Network algorithm (ANN) plays a crucial role depending on the structure of the system and the flexibility of learning, as it can learn in a variety of ways [13, 18]. One way to use this ability is to hybridize the ANN algorithm with another algorithm that strengthens and supports its learning mechanism, like the traits exhibited by the Evolutionary Algorithm (EA). The EA is a widely used algorithm that is based on the biological evolution process [8, 11].

In the abundance of data available, Artificial Intelligence (AI) is proving that it can help in decision-making [17]. Combining AI systems such as EAs with ANNs results in a complex system [28, 29, 33, 34, 42] that can extract and analyze the necessary problems based on their structure and features [1, 4]. The EA provides the best possible input qualities to make up the supporting data [27], and multi-layered ANNs carry out tasks linked to pattern recognition [31]. The combined features and the resulting structural dimensions are employed together to optimize prediction and minimize the estimated errors [15]. The approach provides an example of how hybridised models and augmented forecasting accuracy can be improved [24].

One of the required and necessary tasks is to recognize human movement and the type of movement [12, 16, 20, 35, 39]. The academic community and other organizations find Human Action Recognition (HAR) to be an interesting topic [6, 22]. Identifying human motions through observation is a crucial yet difficult activity known as HAR. This topic includes machine learning algorithms' observations of changes in human movement and activity detection. Numerous human actions have been accurately predicted by Deep Learning (DL)-based techniques [2, 3].

From a surveillance standpoint, activity identification is one of the most important needs for smart home, smart healthcare, and smart city applications. It is common practice to detect unusual activity in security-prone regions to avert potential crimes [25, 26, 40].

Precise human action detection and recognition may be achieved by combining linear discriminant analysis with an ANN [19]. However, the global search capability of the original neural network method is limited, and it is unable to address the issue of redundant data in behaviour recognition. Experts must spend a great deal of time optimizing hyperparameters through trial and error. For these reasons, it is preferable to suggest a neural network and EA-based method for creating a behaviour recognition prediction model. The optimal hyperparameters are automatically determined using an EA [38, 43]. The EA can empower the ANN learning process and improve the validation of the outcome [32]. Since the efficacy of ANN in the recognition process [14, 21], it is crucial to hybridize its features with another algorithm such as EA.

Two AI techniques, ANN and EA, have been combined in a suggested model. The two algorithms' characteristics are combined in this work such that the benefits of the EA selected for hybridization speed up and improve the learning phase of the NN without requiring the backpropagation procedure.

2. Literature Review

In this work, a smart algorithm is proposed that combines the features of two AI algorithms, namely the EA and the ANN algorithm, the proposed mechanism was applied to human physical motion data to distinguish the type of motion, aggressive or normal. Ten different types of actions were used for each type of motion.

The EA and ANN algorithms, which were hybridized with other algorithms, are two of the many mechanisms and techniques that have been employed in the past to identify human action. Prior research was conducted in many settings to identify human behaviour for various objectives. A description of these applications and the various approaches taken is provided here. Research has recognized several human movements and behaviours, such as identifying upper limb activities, tracking and identifying human actions, or identifying sound and some types of movement through data retrieved from running, jogging, etc.

In contemporary applications, ANN have been enhanced by EAs, such as Genetic Algorithms (GAs) [15, 34], by investigating the parameters that enhance ANN structure and utilizing them for various tasks. Additionally, some studies have employed a hybrid approach and methods combining ANN and EA to measure human activity [19, 25, 26, 38, 40, 43]. The suggested approaches have been used on various data sets with varying degrees of success. However, there are certain drawbacks and shortcomings, including the use of complex data sets, multiple goals that need to be achieved using the suggested approach, and the application of the suggested model to various data sets with varying degrees of accuracy.

In the work [34], the ANN's structure has been improved by using a stochastic mutation function with a constant coefficient of variation, the work presented a technique permitted to tackle the intricate combinatorial issue of structural optimization of ANN with a high dimension of the space of optimization parameters. It was suggested that time series values be forecasted using this technique. The following was the ideal ANN architecture that GA discovered: Among the 104 neurons in the first, 80 in the second, 86 in the third, and 109 in the fourth hidden layers, the coefficient of the activation function α=10 and the norm of the learning speed η are 3.1. In the meteorological data time series, the least learning error was 1.2%.

The goal of the research [27] was to use ANN with GA to identify countermeasures against Japanese encephalitis. The best string for the input data was generated and optimized using the GA. By utilizing machine learning techniques on our real-time data, the work was attempted. When employing these algorithms, the improvement rate were 96%. ANN and OpenCV colour change detection are used to identify changes in brain cell colour and infection information. The work in [15], ANN-based surrogate models were suggested as an effective way to perform minimum weight optimization of composite laminates. The trained ANN models were then used in an evolutionary GA to optimize the structural dimensions and stacking sequences to minimize the weight of the composite laminates. The constructed ANN models demonstrate a high the fit factor, with R2 values for the flat, blade-stiffened, and hat-stiffened laminates, respectively, equal to 0.996, 0.987, and 0.987.

Raja Subramanian and Vasudevan [26] put up a paradigm for identifying human activity. Rather than using a standard machine learning or DL method, it operated according to a GA optimization criterion. They put out a brand-new, faster genetic method that can identify important variations in video frames. The activity of the altered frame is detected by means of the DL algorithm. Using three common activity recognition data sets Hollywood2, KTH, and UCF-ARG this deep genetic model demonstrated sound recognition accuracy. Using the Hollywood2 and KTH action data sets, the suggested model produced sound recognition accuracy of 98.42% and 97.83%, respectively. A further evaluation of the model was conducted using the UAV-acquired video data set, UCF-ARG, demonstrating an 81.40% recognition accuracy.

The research in [40] presented Human Activity Recognition Based on Automatic Neural Architecture Searchs (HARNAS), a method for leveraging Neural Architecture Search (NAS) to find models appropriate for HAR challenges. HARNAS employed the multi-objective search algorithm known as NSGA-II as its search strategy. A bi-objective issue is created when the F1 score and the quantity of Floating-Point Operations (FLOPs) are chosen to tradeoff between a model's performance and computation speed. Most of their trials were conducted using the opportunity dataset, and they also assessed the model's portability using the UniMiB-SHAR dataset. The outcomes demonstrated that HARNAS, when created without human modifications, can outperform the best model with human tweaks. On the Opportunity dataset, HARNAS achieved an F1 score of 92.16% and parameters of 0.32 MB.

With the help of triaxial signals obtained from accelerometer sensors, Quaid and Jalal [25] created a new, efficient framework for recognizing human behaviors. They also suggested a reweighted GA that offered chromosome adjustment variation and a combination of windowed signal patterns for the recognition of random human behaviors. Numerous tests conducted on three difficult accelerometer sensor datasets the Human Microbiome Project (HMP), Wireless Sensor Data Mining) WISDM, and self-annotated IM-Sporting Behavior (IMSB) datasets showed that the suggested approach dramatically outperformed state-of-the-art techniques in terms of accuracy, yielding results that were 85.43% higher. Combining linear discriminant analysis with an ANN for accurate human action detection and recognition was the main objective of the study [19], the suggested method identified intricate human movements in two cutting-edge datasets; that is, for each dataset, one-third of the picture sequences are retained for testing and validation. The suggested model received the remainder for training. Confusion matrices with real-class labels were created to demonstrate the results. Because running with the ball, dribbling, and standing were distinct movements that our multidimensional characteristics could easily categorize, the findings for these activities were 100% correct. There was also some misunderstanding when it came to passing, sprinting, shooting, and accepting passes. For this dataset, the average recognition accuracy was 87.57%. In [43], a neural network-based technique for Upper Limb Activity Recognition (UPLA) was introduced. The number of neurons in neural networks, the size of the sliding window, and the overlap of neighbouring sequences all affect how accurately activities are recognized. In contrast, there hasn't been as much effort put into autonomously optimizing neural networks' hyperparameters. Experts must spend a great deal of time optimizing hyperparameters through trial and error. The population selection method was enhanced, and the optimal hyperparameters were automatically found using the GA. Convolutional neural networks and seven conventional classification algorithms were compared with the improved approach; the new method produced an accuracy of 97.9%.

A neural network and EA model were proposed in [38]; this model was able to recognize and forecast behavior. The study advances the multi-dimensional coevolution approach in conjunction with neural network algorithms and presents the idea of GA. Concurrently, the behaviour recognition analysis model was built, and the neural network algorithm's threshold and weight were established. The findings demonstrated that, when compared to the prediction accuracy and convergence of behaviour recognition based on a GA combined with a neural network algorithm, it was 86.27. The accuracy of the GA and the classical neural network algorithm was 59.22 and 80.13, respectively.

Various studies show an attempt to use different techniques with the help of hybrid systems to solve different problems. In our paper, we tried to prove the efficiency of the hybrid system by using EA to carry out the learning process in ANN to reach the solution in a faster manner and with high accuracy. The proposed EA replaces the learning process by breaking the original ANN into multiple ANNs, and the selection process is done according to the highest accuracy obtained from each ANN after breaking. Crossover and mutation are taken for the purpose of improving each ANN; after that, the process of merging the ANNs into one ANN algorithm is done in order of the layers chosen for the new generation. The proposed optimization method of breaking the original ANN algorithm into multiple ANNs helped to easily and quickly find the parameters that need to be selected for the new generation and the parameters that need to be improved, while the EA helped to enhance the parameters that increase the efficiency of ANN through the process of crossover and improve the parameters with less efficiency through the mutation process.

3. Description of Data

Multidimensional data was suggested to apply the proposed methodology. The recognition of human actions that are aggressive or normal is the outcome of our implementation. Ten actions correspond to each type; the experiment's data was used to obtain these values [36].

Several people (aged 25 to 30) took part in the trial. During numerous trials, each participant was taught to perform ten normal and ten aggressive tasks. A human being is monitored engaging in physical activities within a three-dimensional intelligent setting. Two external devices, a 3D tracker Vicon system and a mobile robot SCITOS G5, work together as a perception-to-action unit to provide a categorization of the actions done and data based on mechanical properties [37].

The used dataset was represented by three databases in training and one in testing as in Figure 1. Each database included two kinds of actions, which are aggressive and normal; each action had 10 activities; there are 1061 samples for each activity; each sample consists of nine features represented by body parts that identify the sort of movement; each body component carries a pair of markers that are right and left (except for the head) that may be used to indicate the type of movement; and each of them is three-dimensional data.

The body component is represented by the following: right and left arm signs represent wrist and elbow, right and left leg signs represent ankle and knee, and their measurements are obtained as x, y, and z coordinates as in Figure 2-a), which forms a multi-dimension data for each feature as in Figure 1-b).

The proposed system was applied on other dataset. The data was used to do classification of human actions. The data gathered by an inertial sensor-equipped smartphone worn around the waist was used to record 30 people engaging in Activities of Daily Livings (ADLs) for the HAR database. Time-series and multivariate are the dataset's characteristics.

The studies were conducted on a sample of thirty individuals aged between 19 and 48. The Samsung Galaxy S II smartphone was worn around each person's waist as they engaged in six different activities: standing, sitting, laying, walking, walking downstairs, and walking upstairs.

It recorded 3-axial angular velocity and 3-axial linear acceleration at a steady rate of 50Hz using its integrated accelerometer and gyroscope. The trials were captured on camera so that the data could be manually labelled. The collected dataset was divided into two sets at random, with 70% of the volunteers chosen to provide training data and 30% to provide test data.

Noise filters were used to pre-process the accelerometer and gyroscope sensor signals before sampling them in fixed-width sliding windows with a 50% overlap and a duration of 2.56 seconds (128 readings/window). Using a Butterworth low-pass filter, the sensor acceleration signal which consists of both body motion and gravitational components was divided into two categories: body acceleration and gravity. Because the gravitational force is thought to be made up only of low frequency elements, a 0.3 Hz cutoff frequency filter was used.

A vector of features was derived from each window by computing variables in the frequency and time domains [9].

This concept validates the flexibility of the suggested structural model in implementation by Feed-Forward ANN (FFANN) and in learning by EA. This is due to the overlapping dimensions that correspond to the required categorization.

4. Methodology

The efficiency of the phenotypic neuro-EA has been improved by a methodology that can learn from multiple databases during the necessary training stage. The implementation of the model passed through stages: data preparation, feeding data to an artificial neural algorithm, evaluating the outcome of the algorithm, and the learning stage, which is composed of three phases: breaking up the ANN into multiple ANNs, the process of an EA, and combining ANNs into ANNs. The training phase passed through many iterations; in each iteration, the obtained result was examined; if it was satisfactory, it was approved, and if it was not, the learning process was repeated until the result was satisfactory, after training phase, the model is tested to do the validation.

Multidimensional numeric data was used, as previously described in the section of the data description. To prepare the data, it needs to work in stages:

4.1. Data Preparing

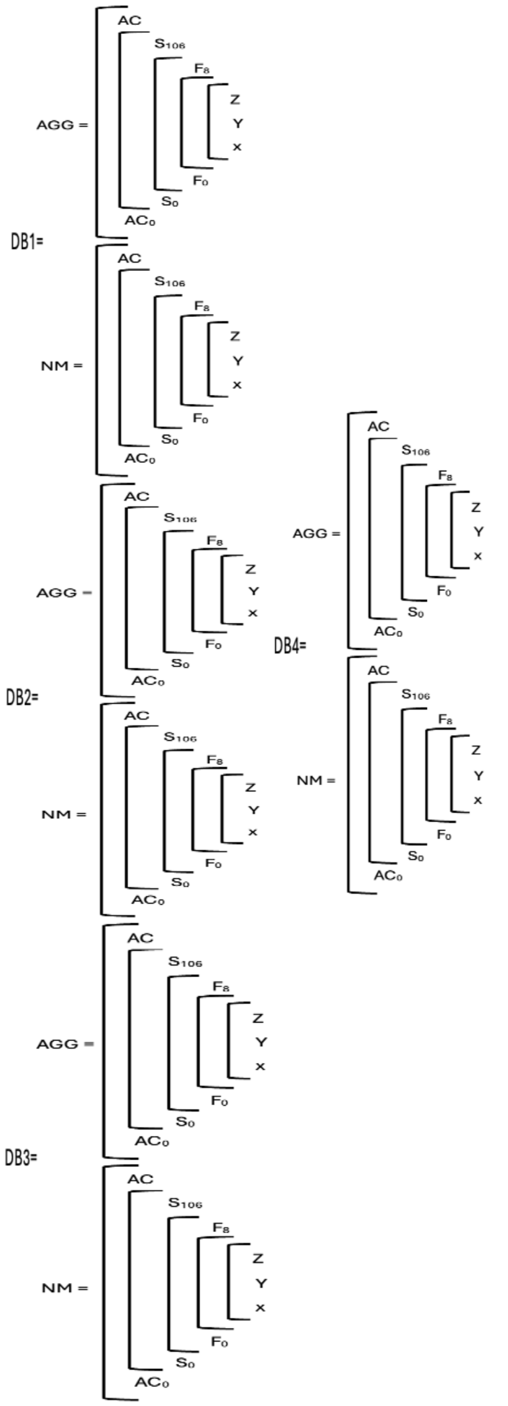
Our proposed model's data set has a connection to physical activities. First data set was gathered from each participant in 20 distinct experiments was given instructions to perform 10 aggressive and 10 normal tasks. The Vicon system, a 3D tracker, and the SCITOS G5, a mobile robot, work together to provide mechanical attribute-based data and effectively classify activities [36, 37], while second dataset was gathered by smart phones [9].

The normalisation function is implemented by Quantile Transformer using scikit-learn to prepare the first dataset for classification before introducing it to the classifiers, to maximise the classification performance of the classifiers, and by applying the scaling approach, commonly known as "Min-Max feature scaling," which entails scaling data in the interval [0-1]. It entails rescaling data and employing minimal and maximum data as bounds. In mathematical terms, as in Equation (1), it allows a vector x to be scaled in the interval [0, 1] [5, 7, 30].

$$x_i = (x_{max} - x_{min}) * \frac{x_i - x_{min}}{x_{max} - x_{min}} + x_{min} \qquad (1)$$

Where min and max are the minimum and maximum values of the feature x [5]. The data was normalized to a range between zero and one to be more convenient for the work.

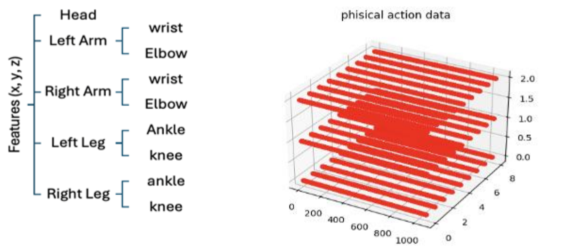
The data was arranged based on the factors required to obtain the classification. The factors are, in order, the size of the dataset, the markers or features, and the coordinates for each marker. The data are arranged with suitable format for each activity for both aggressive and normal behaviour. According to the required phases, there are two parts of data that are provided for the suggested method. During the training phase, three databases are used. Each database is separated into two categories: aggressive and normal behavior. Each category has ten actions, and each one has the data ordered based on the required factors, resulting in 6D. During the testing phase, a single database is utilized, which, as previously stated in this section, results in 5D. Data dimensions are specified as in Figure 1. In the training phase, we used six-dimensional data, which is represented by three databases; for each database, there are two datasets represented by the two actions (aggressive and normal); each action consists of ten activities; each activity includes 1061 samples; and each sample contains nine features. With each feature, there are three coordinates(x, y, z) to represent the dimension (3, 2, 10, 1061, 9, 3), whereas in Figure 1-b) the testing phase used a 5-dimension database, represented by (2, 10, 1061, 9, 3), whereas DB represents the database, AGG represents the aggressive action, NM represents the normal action, ACn represents the 10 activities, Sn represents the samples of data, Fn represents the nine features, and x, y, and z represent the coordinates of data. Each database included two kinds of actions, which are aggressive and normal; each action had 10 activities; there are 1061 samples for each activity; each sample consists of nine features represented by body parts that identify the sort of movement; each body component carries a pair of markers that are right and left (except for the head) that may be used to indicate the type of movement; and each of them is three-dimensional data.



a) DB1, DB2, DB3 for training phase. b) DB4 for testing phase.

Figure 1. The data dimension.

The body component is represented by the following: right and left arm signs represent wrist and elbow, right and left leg signs represent ankle and knee, and their measurements are obtained as x, y, and z coordinates as in Figure 2-a), the nine features for each sample, Figure 2-b) The activity dimension is represented by the nine features, with each feature having three coordinates (x, y, and z) that specify each 3D marker’s origin in space.



a) The nine features for each sample. b) (x, y, z) coordinates for each 3D market.

Figure 2. The representation of samples.

The proposed system was applied on other dataset. The data was used to do classification of human actions. The data gathered by an inertial sensor-equipped smartphone worn around the waist was used to record 30 people engaging in ADLs for the HAR database. Time-series and multivariate are the dataset’s characteristics.

The studies were conducted on a sample of thirty individuals aged between 19 and 48. The Samsung Galaxy S II smartphone was worn around each person's waist as they engaged in six different activities: standing, sitting, laying, walking, walking downstairs, and walking upstairs. It recorded 3-axial angular velocity and 3-axial linear acceleration at a steady rate of 50Hz using its integrated accelerometer and gyroscope. The trials were captured on camera so that the data could be manually labelled. The collected dataset was divided into two sets at random, with 70% of the volunteers chosen to provide training data and 30% to provide test data.

Noise filters were used to pre-process the accelerometer and gyroscope sensor signals before sampling them in fixed-width sliding windows with a 50% overlap and a duration of 2.56 seconds (128 readings/window). Using a Butterworth low-pass filter, the sensor acceleration signal-which consists of both body motion and gravitational components-was divided into two categories: body acceleration and gravity. Because the gravitational force is thought to be made up only of low frequency elements, a 0.3 Hz cutoff frequency filter was used. A vector of features was derived from each window by computing variables in the frequency and time domains [9].

This concept validates the flexibility of the suggested structural model in implementation by FFANN and in learning by EA. This is due to the overlapping dimensions that correspond to the required categorization.

4.2. Feed-Forward ANN Architecture

FFANN consists of an input layer with two neurons, each neuron represents one type of action, which is aggressive or normal. A matrix of weights is created with the size of the input data and multiplied with the input, adding the bias to it as in Equation (2), and sending results to the hidden layers to be processed. There are 11 hidden layers, each with three neurons. The output is obtained from each neuron using the sigmoid function, as in Equations (3) and (4). The output is passed from the last hidden layer to the output layer by the identity function, as in Equation (5). The output layer consists of two neurons to give the result, each classifying aggressive or normal activity. The result is compared with the real result by mean square error, as in Equation (6) [10, 23, 41].

$$s = \sum_{i=1}^{n} W_i . X_i \; + b \tag{2}$$

$$\sigma = \frac{1}{1 + e^{-s}} \tag{3}$$

$$neuron = \sigma(\hat{y}) = \sigma(\sum_{i=1}^{n} W_i . X_i \; + b \tag{4}$$

$$\hat{z} = \sum_{i=1}^{n} W_i . neuron_i \; + b \tag{5}$$

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (z_i - \hat{z}_i)^2 \tag{6}$$

i: the index of feature for each sample of data.
n: the number of features.
W: the weight.
X: the features.
b: the bias.
σ: the activation function(sigmoid).
neuron: the output from the neuron.
$\hat{z}$: the output from FFANN for each sample of data.
z: the desired output.
MSE : the mean square error.
m: the total number of samples in dataset.

4.3. Optimization Process

If unsatisfactory results are obtained, the optimization process is required in FFANN, and it is implemented through three phases.

4.3.1. Breaking up Phase

The optimization phase requires breaking up the ANN algorithm into multiple ANNs algorithms based on the number of hidden layers by assuming an output layer after each hidden layer as shown in Figure 3.

After each hidden layer, the result of activation function sent to the assuming output layer to be the result of new breaking ANNs, the output from each ANN is computed by identity function, as in Equation (7), and it needs to calculate the mean square error after getting the output from each new ANNs, , which is represented by Equation (8).

The obtained ANNs from the breaking process will be considered a population from each iteration of the learning phase, and each ANN is considered an individual in this population.

The obtaining errors from each ANN are kept in a list, as in Equation (9), and it is considered a fitness list to be used in the required optimization steps.

$$ANN_i = \sum_{j=1}^{r} w_j^i . neuron_j \; + b \tag{7}$$

$$MSE_{ANN_i} = \frac{1}{m} \sum_{i=1}^{m} (z_i - ANN_i)^2 \tag{8}$$

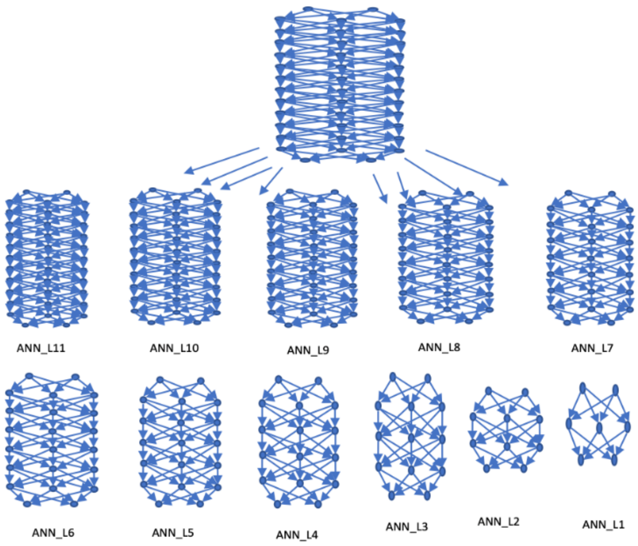


Figure 3. The breaking up the original ANN into multiple ANNs.

i: the index of hidden layer.
j: the index of feature for each sample of data.
r: the number of neurons.
neuronj: the output from neuron as in Equation (4).

$$Error_{ANN_s} = [MSE_{ANN_1}, MSE_{ANN_2}, .., MSE_{ANN_{11}}] \tag{9}$$

4.3.2. Evolutionary Algorithm Role

For the purpose of the optimize parameters, the EA has been proposed and has replaced the backpropagation process. This process operated through stages.

The selection stage is considered important and is adopted by the subsequent stages. At this stage, it is examined which layer is better in terms of weight. The process of breaking the algorithm into multi-ANNs facilitated the process of examining the layers with the best weights, as explained in the section on breaking the algorithm.

At this point, each ANN is evaluated individually following the process of breaking up into several ANNs, which represents the breaking up process. In this procedure, we feed the data into each ANN, compute the result, and compare it with the actual result using the mean squared error.

The list of saved errors, as in Equation (9), is used to evaluate the effectiveness of ANNs algorithms or layers in the original algorithm.

The evaluation process is carried out through the process of ranking from best to worst; it is implemented by a sorting algorithm as explained in Equation (10).

$$ANN_i \leq ANN_{i-1} \leq ANN_{i-2} \leq \cdots \leq ANN_{i-n} \qquad (10)$$

The stage of crossover provides the opportunity for the best classes to survive and be used by the next generation. By using the selection method, it will be possible to determine which ANN is the most effective and which one requires improvement, as well as the accuracy of each ANN. The qualities that show how well an ANN is implemented are swapped out during the crossover phase, which is represented by weights.

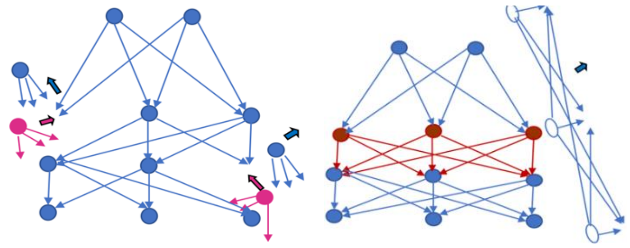
From the ranking process that was carried out in the fitness list, the most efficient layers with the least error were known, and they are in the first positions in the fitness list. After ranking step, the two layers with the lowest error value are switched with the two layers with the highest error value, as in Equations (11) and (12) that formed from our suggestion. From the ranking process that was carried out in the fitness list, the most efficient layers with the least error were known, and they are in the first positions in the fitness list. After ranking step, the two layers with the lowest error value are switched with the two layers with the highest error value, as in Equations (11) and (12) that formed from our suggestion.

$$\begin{aligned} F(ANN_{11}, ANN_1) &= F(ANN_{11}(W_{ijk}), ANN_1(G_{ijk})) \\ &= ANN_{11}(W_{ijk}) \Leftrightarrow ANN_1(G_{ijk}) \\ &= ANN_{11}(G_{ijk}), ANN_1(W_{ijk}) \end{aligned} \qquad (11)$$

$$\begin{aligned} F(ANN_{10}, ANN_2) &= F(ANN_{10}(W_{ijk}), ANN_2(G_{ijk})) \\ &= ANN_{10}(W_{ijk}) \Leftrightarrow ANN_2(G_{ijk}) \\ &= ANN_{10}(G_{ijk}), ANN_2(W_{ijk}) \end{aligned} \qquad (12)$$

W, *G*: the coefficient weights for *ANN*.
i, *j*: the rows and columns for *W* and *G*.
k: the last layer for *ANN*.

The step of mutation improves the remaining ANNs with low accuracy by changing the weights randomly on neurons and layers. ANNs that are less effective and unsuitable for survival in the following generation are currently given the chance. The weighted properties of these ANNs need to be altered in order for them to survive. By altering the weights' values, the mutation process is used to carry out this process of change, as shown in Figure 4-a) and (b), respectively, but the random change is restricted under certain limits, these limits are determined by the error obtained from the previous generation; the limits were in the opposite direction of the error. We try to choose values opposite the direction of the values of the weights with the highest error value. We set the standard boundaries in range ($v1$=0.001, $v2$=1).



a) Neurons. b) Layers.

Figure 4. The process of mutation on neurons and layers.

If the value of the overall accuracy system in the upcoming generation is low, it must update the range with direction to close the zero-error value, as in Equations (13) and (14), which formed from our suggestion. The minimum limit is updated in increase, and the random choice will be among the values that are close to 1, which represents aggressive action, and vice versa with Equation (13), the higher limit is updated in decrease to be the random choice among the values that are close to 0, which represents normal action. In this process, the mutation is restricted to the suggested limit of the range of values to generate weights randomly.

$$v1 = v1 + 0.1 \qquad (13)$$

$$v2 = v2 * 0.1 \qquad (14)$$

4.3.3. Combining Process

After completing the required steps for improvement, ANNs algorithms are combined into one ANN algorithm. The combining process is done after the mutation process, which is the last phase of the EA.

The combining process is implemented by merging the last hidden layer from each ANNs algorithm to form a new FFANN algorithm, as in Equation (15). Accordingly, in the learning process, we adopted a mechanism to make the last layer have the highest accuracy in each ANN through crossover and mutation processes when executing an EA.

The resulting combining process produces a new FFANN, which consists of 11 hidden layers. The new FFANN algorithm contains the most efficient hidden layers, which have been examined and are able to provide classification with high accuracy.

$$\mathrm{FFANN_{NEW}} = \mathrm{f}\left(\mathrm{Na_1}(G_{ijk})\right).\mathrm{f}\left(\mathrm{Na_2}(W_{ijk})\right)\ldots\mathrm{f}\left(\mathrm{Na_{11}}(Z_{ijk})\right) \qquad (15)$$

Na1 …. Na11: the ANNs.
W: the coefficient weights for *Na*.
G: the coefficient weights for *Nb*.
i, *j*: the rows and columns for *W* and *G*.
k: the last layer of *ANN*.

The phenotypic neuro-EA pseudocode and flowchart in Figure 5. Explain the optimization process, and the algorithm as follows:

Algorithm 1: Phenotypic Neuro-Evolutionary Algorithm.

Input: Train: DB1, DB2, DB3
Test: DB4
Output: Actions classification

```
data-trn = [DB1, DB2, DB3]
w1 = np.random.rand(num_layer, num_neuron, len(data-trn))
data-tst = DB4
w2 = np.random.rand(num_layer, num_neuron, len(data-tst))
def normaliz (data):
    a1  = np.max(datai)
    a2 = np.min(data)
    a = (a1-a2)*(( datai – a2)/(a1-a2))+a2
     return(a)
data_train= normaliz (data-trn)
def SumP(data, weight, Bias):
     return(np.dot(data, weight) + Bias)
def sigmoid(x):
     return 1 / (1 + np.exp(-x))
def feedforwardANN(data_train, w1):
     For i in range (num_layer):
          For j in range(num_neuron):
               For k1 in range(len(data_train)):
                    Z1 = SumP (data_train, w1, Bias)
                    Z2 = sigmoid (Z1)
                    L1.append(Z2)
                L2.append(L1)
          L3.append(L2)
     For i in range (num_layer):
          For j in range(num_neuron):
      mse =1/(len(data_train) * sqr(desired_output – L3))
               er1.append(mse)
          er2.append(er1)
     error = er2(num_layer-1)
     return er2, error
def select(er2, w1):
     b = np.sort(er2)
     for i in range(len(er2)):
         for j in range(len(er2)):
         if er2[i] == b[j]:
             d.append(i)
     For i in range (num_layer):
          h = d[i]
          selc.append(w1[h])
     return(selc)
 def crossover(w):
    x = w[0]
    w[0] = w[11]
    w[11] = x
    x = w[10]
    w[10] = w[1]
    w[10] = x
    return(w)
v1, v2 = 0, 1
def mutat(w, data, v1, v2):
    a = len(data)
     For i in range(num_layer -2):
          x = random.randint(0,8)
          x1.append(x)
          For j in range(num_neuron):
               s = np.random.uniform(v1, v2, size=(a))
               s1.append(s)
          w[x].append(s1)
    return(w)
r1, r2 = feedforwardANN(data_train, w1)
while r2>0.1:
   // set v1, v2 with negative direction of error
   selc = select(r1, w1)
   w = crossover(selc)
   mu = mutat(w, data_train, v1, v2)
   r1, r2 = feedforwardANN(data_train, mu)
   if r2 < 0.1:
       feedforwardANN(data_tst, w2)
       break
```

5. Results

The proposed system applied on the suggested method has been evaluated using multiple datasets [9, 36] in order to perform the task of recognizing human actions. Four databases were implemented to identify the aggressive and normal action types, which comprise ten different types of actions each. More varied data was used to apply the system; data larger than 7000 samples and 561 features for each [9], it was used to identify human activity; roughly 80% of the data was used during training and 20% during testing, a dataset was used to identify the following actions: standing, sitting, laying, walking, walking downstairs, and walking upstairs.

To approve learning and get satisfactory outcomes, the implementation is separated into two phases: training and testing.

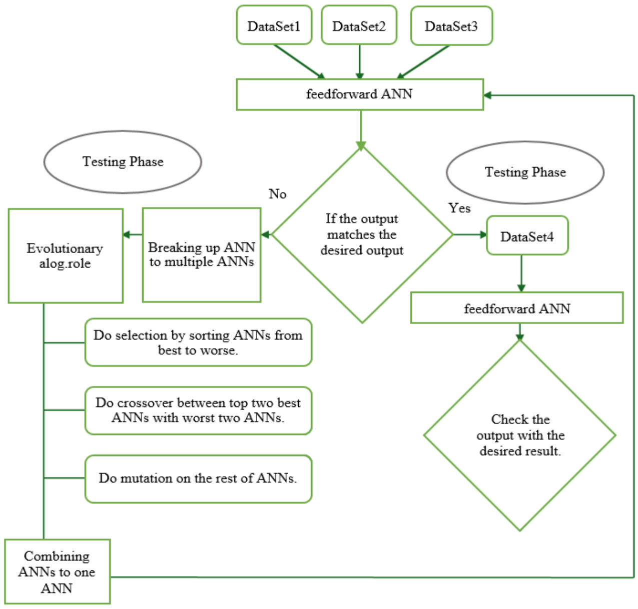


Figure 5. Phenotypic neuro-evolutionary flowchart.

5.1. Training Phase

The system applied on variety datasets, to recognize the type of action which are aggressive or normal the system was trained using three databases, each with two types of aggressive and normal activity and 10 categories for each activity. The system was trained on each of the 10 types. The data from the three databases was passed to the artificial neural front-end algorithm to check the result. If the result is not satisfactory, the algorithm will be improved using an EA, which requires breaking the artificial neural algorithm into multiple ANNs as shown in the Figure 3 and maintaining the error value resulting from each post- break algorithm to be used in the EA phase. The role of the EA begins with the selection phase based on breaking the algorithm, which uses the error values resulting from each ANN and rearranges the algorithms from best to worst, as shown in Tables 1 and 2 for aggressive and normal actions respectively.

Table 1. The breaking and selection phases for aggressive action.

Aggressive action			
Breaking phase		The selection phase	
ANNs	Accuracy	ANNs	Accuracy
0	95.06158360027369	0	95.06158360027369
1	0.0	5	87.91257054235956
2	70.48007827510644	10	75.37471114729755
3	0.0	2	70.48007827510644
4	0.0	9	69.33591670018728
5	87.91257054235956	6	25.53584519667622
6	25.53584519667622	1	0.0
7	0.0	3	0.0
8	0.0	4	0.0
9	69.33591670018728	7	0.0
10	75.37471114729755	8	0.0

Table 2. The breaking and selection phases for normal action.

Normal action			
Breaking phase		The selection phase	
ANNs	Accuracy	ANNs	Accuracy
0	81.35306284696951	8	95.6849235023387
1	94.34914244614108	7	95.58084144150105
2	87.26224371406316	4	95.57670307204171
3	94.79460998505589	6	94.8s3192204412498
4	95.57670307204171	3	94.79460998505589
5	87.75386133967102	1	94.34914244614108
6	94.83192204412498	9	87.8096880562277
7	95.58084144150105	5	87.75386133967102
8	95.68492350233878	2	87.26224371406316
9	87.8096880562277	10	85.27270655573171
10	85.27270655573171	0	81.35306284696951

Table 3. The first trained dataset.

Aggressive action		
Action type	**MSE**	**Accuracy**
Elbowing	0.09897487120762768	90.10251287923722
Front kicking	0.09800611425948148	90.19938857405185
Hammering	0.09929454029578523	90.07054597042148
Headering	0.09834289985632305	90.16571001436769
Kneeing	0.09877155432842066	90.12284456715793
Pulling	0.09850367614771283	90.14963238522871
Punching	0.09837148994202279	90.16285100579772
Pushing	0.09814084253214674	90.18591574678533
Side kicking	0.09779541407864159	90.22045859213584
Slapping	0.09739016775488114	90.26098322451188
Accuracy	**90.16408429596956**	
Normal action		
Action type	**MSE**	**Accuracy**
Bowing	0.04190016579326249	95.80998342067375
Clapping	0.04152967269503824	95.84703273049617
Handshaking	0.04181133964618717	95.81886603538128
Hugging	0.04190459693808727	95.80954030619128
Jumping	0.04163499408209211	95.83650059179078
Running	0.041872208529894483	95.81277914701054
Seating	0.041871959693298225	95.81280403067018
Standing	0.04186325852684629	95.81367414731538
Walking	0.04171678131303149	95.82832186869685
Waving	0.04157420545086991	95.84257945491301
Accuracy	**95.82320817331393**	

Following the steps of breaking, selection, crossover and mutation are carried out, and finally, the ANNs algorithms are combined into a single ANN algorithm to generate the new FFANN algorithm, which is used to send data to the next generation. Ten actions representing both aggressive and typical sorts of activities are represented by multi-dimensional data that is stored in three databases and is passed as multi-dimensional data that is stored in two databases. The data stored in the three databases is passed for training, as shown in the Tables 3, 4, and 5. Tables 3, 4, and 5 show the results of mean square error and accuracy for each activity related to each type (aggressive and normal) in each iteration of the training phase.

Table 4. The second trained dataset.

Aggressive action		
Action type	**MSE**	**Accuracy**
Elbowing	0.09767150091341069	90.23284990865893
Front kicking	0.09907209444128845	90.09279055587116
Hammering	0.09943420649619074	90.05657935038093
Headering	0.09798022330296213	90.2019776697038
Kneeing	0.09677557861330183	90.32244213866983
Pulling	0.09787102697573434	90.21289730242657
Punching	0.09821260161291197	90.1787398387088
Pushing	0.09808966260534242	90.19103373946575
Side kicking	0.09760931803604808	90.2390681963952
Slapping	0.09804417377244205	90.1955826227558
Accuracy	**90.19239613230368**	
Normal action		
Action type	**MSE**	**Accuracy**
Bowing	0.041728030728533524	95.82719692714664
Clapping	0.04167096448315706	95.8329035516843
Handshaking	0.041743773786799776	95.82562262132002
Hugging	0.041838898054688066	95.81611019453119
Jumping	0.04188486333066445	95.81151366693355
Running	0.04181148980954863	95.81885101904514
Seating	0.04194933841371019	95.80506615862899
Standing	0.041662435279492434	95.83375647205075
Walking	0.04193733720242114	95.80626627975789
Waving	0.0419489634245092	95.80510365754908
Accuracy	**95.81823905486475**	

Table 5. The third trained dataset.

Aggressive action		
Action type	**MSE**	**Accuracy**
Elbowing	0.09910828139077435	90.08917186092256
Front kicking	0.0984306689957578	90.15693310042423
Hammering	0.10036535339502255	89.96346466049775
Headering	0.09761926113743659	90.23807388625634
Kneeing	0.09986006514254286	90.01399348574571
Pulling	0.0982508114049132	90.17491885950868
Punching	0.09937187687054405	90.0628123129456
Pushing	0.0992029236481672	90.07970763518328
Side kicking	0.09894738083013926	90.10526191698607
Slapping	0.09766722776966477	90.23327722303353
Accuracy	**90.11176149415039**	
Normal action		
Action type	**MSE**	**Accuracy**
Bowing	0.0419121650011947	95.80878349988053
Clapping	0.04121699348755734	95.87830065124426
Handshaking	0.041698056506847686	95.83019434931524
Hugging	0.04157930332086695	95.84206966791331
Jumping	0.041624438247608773	95.83755617523913
Running	0.041796577372944156	95.82034226270558
Seating	0.04183029861267171	95.81697013873283
Standing	0.04161819055245526	95.83818094475447
Walking	0.041576037516821776	95.84239624831781
Waving	0.04160277133952405	95.8397228660476
Accuracy	**95.83545168041508**	

For three databases, the overall training phase accuracy of the last generation is 90.15608064080789 for aggressive action and 95.82563296953126 for normal action, Figures 6 and 7 explain the accuracy for training phase for aggressive and normal data.

The system achieved high accuracy when trained on the three databases with less error by the using of mean square error as explained in the Tables 3, 4, and 5.

Figure 6. The accuracy of training phase for aggressive action for the three DBs.

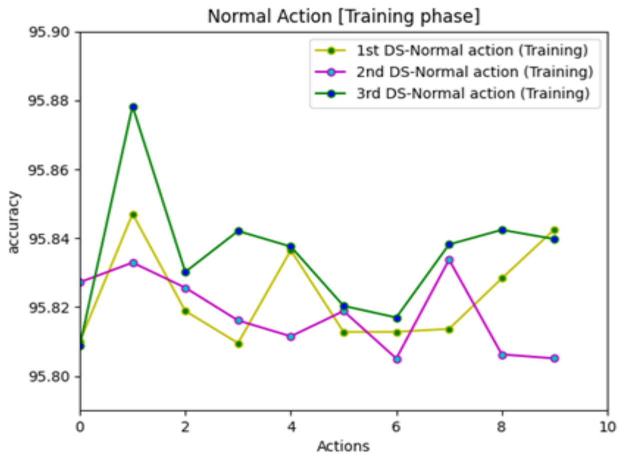


Figure 7. The accuracy of training phase for normal action for the three DBs.

Table 6. The training on classifying type of human action.

Training phase		
Type of Action	**MSE**	**Accuracy**
Standing	0.0194445942	98.05554057708704
Sitting	0.0111889345	98.88110655044301
Laying	0.04353273224	95.64672677596047
Walking	0.04359575838	95.64042416179619
Walking downstairs	0.0352878852	96.4712114801326
Walking upstairs	0.04615227861	95.38477213898932

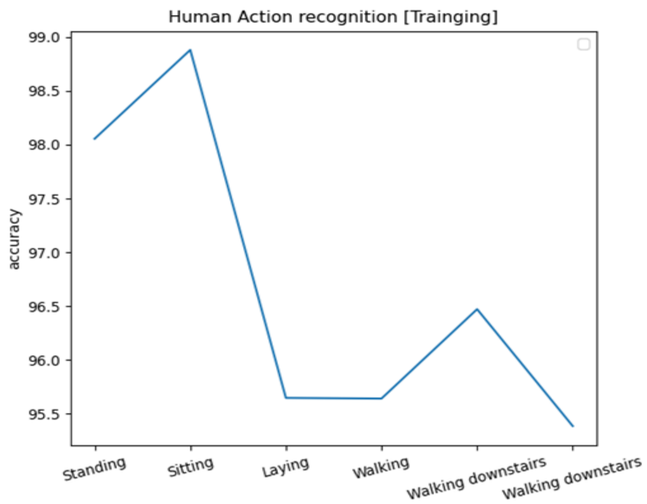


Figure 8. The training phase accuracy of classifying human action.

The system trained on other dataset to do action recognition. 80% from 7000 samples and 561 features for each [9]. The implementation used to recognize the sixth actions as explained in Table 6. Figure 8 explains the accuracy for training phase for classifying human action.

5.2. Testing Phase

After obtaining satisfactory results from the last generation in the training phase, additional data was passed, represented by aggressive and normal activities, and stored in a single database for the purpose of testing the validity of the results. This was done in order to verify the validity of the proposed system and the optimization process. During testing, satisfactory results were obtained, as shown in Table 7. The accuracy obtained from the testing phase is 90.16408429596957 for aggressive and 95.82320817331393 for normal, as depicted in Figures 9 and 10 respectively.

Table 7. Testing phase on the fourth dataset.

Aggressive action		
Action type	**MSE**	**Accuracy**
Elbowing	0.098974871207627	90.1025128792373
Front kicking	0.09800611425948146	90.19938857405185
Hammering	0.09929454029578465	90.07054597042153
Headering	0.09834289985632313	90.16571001436769
Kneeing	0.0987715543284206	90.12284456715794
Pulling	0.09850367614771266	90.14963238522873
Punching	0.09837148994202319	90.16285100579768
Pushing	0.098140842532147	90.1859157467853
Side kicking	0.09779541407864148	90.22045859213586
Slapping	0.0973901677548813	90.26098322451188
Accuracy	**90.16408429596957**	
Normal action		
Action type	**MSE**	**Accuracy**
Bowing	0.04190016579326288	95.80998342067372
Clapping	0.0415296726950383	95.84703273049617
Handshaking	0.04181133964618717	95.81886603538128
Hugging	0.04190459693808716	95.80954030619128
Jumping	0.04163499408209218	95.83650059179078
Running	0.041872208529894546	95.81277914701054
Seating	0.041871959693298134	95.81280403067018
Standing	0.04186325852684642	95.81367414731535
Walking	0.04171678131303115	95.82832186869689
Waving	0.041574205450869976	95.84257945491301
Accuracy	**95.82320817331393**	

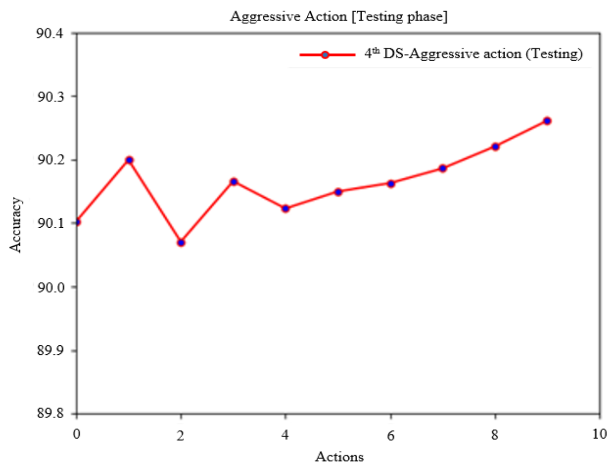


Figure 9. The accuracy of testing phase for aggressive action for the fourth DB.

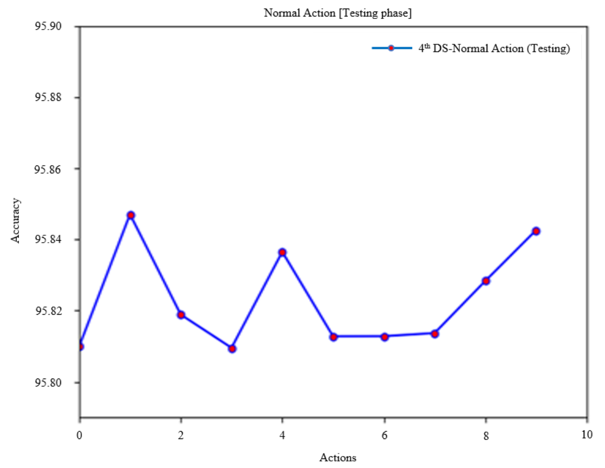


Figure 10. The accuracy of testing phase for normal action for the fourth DB.

Table 8. The testing on classifying type of human action.

Training phase		
Type of action	**MSE**	**Accuracy**
Standing	0.01901035943	98.09896405695486
Sitting	0.01117056942	98.88294305827982
Laying	0.04364527616	95.63547238410109
Walking	0.04349302024	95.65069797595265
Walking downstairs	0.03723951763	96.27604823715042
Walking upstairs	0.04728140522	95.27185947798723

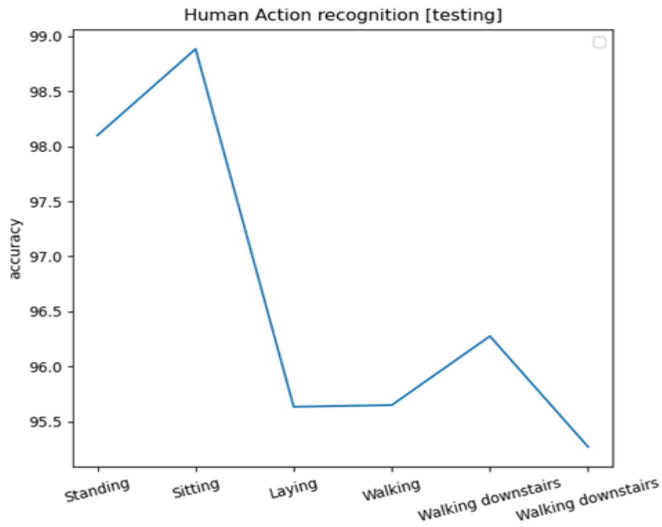


Figure 11. The training phase accuracy of classifying human actions.

The system was tested using a dataset to classify human actions, as shown in Table 8 and Figure 11, which provides an explanation of the testing accuracy.

The system implementation performance was compared with other state-of-the-art HAR techniques as in Table 9.

Table 9. The comparison implementation techniques of HAR.

Accuracy	**Dataset**	**Method**	**Reference/Year**
93.68%, 59.06%	Fusion techniques of virtual reality and SOM neural network	UT-Kinect action, MSRdailyactivity3D	[16], 2020
81.25%	GA	MuHAVi-Uncut, iXMAS, and IAVID-1	[20], 2022
92.16%	raw depth maps	SU 3DHOI dataset	[35], 2021
95.79%	Feature Selection and Dense Neural Network	UCI HAR data set	[3], 2020
For the Convolution Memory Fusion Algorithm (CMFA), the percentage was 96.76%, while for the Convolution Gated Fusion Algorithm (CGFA), it was 84.35%, with 96.91% for both smartwatches and smartphones	Time-series data produced by wearable sensors and smartphones	Hybrid Learning Algorithms (HLA)	[2], 2022
98.42%, 97.83%, and 81.40%	Hollywood2, KTH, UCF-ARG	A deep GA	[26], 2021
92.16%	HARNAS	Opportunity dataset, and UniMiB-SHAR dataset	[40], 2020
%85.43	HMP, WISDM and self-annotated IMSB datasets	GA	[25], 2020
%87.57	KTH-dataset and Weizmann human action.	ANN	[19], 2020
97.9%	Raw data was gathered by an LP-RESEARCH sensor.	ANN with GA	[43], 2020
91-96%, 95-98%	HAR [36] by Vicon mobile robot SCITOS G5 system, HAR by Smartphones [9]	A neuro phenotypic evolution algorithm	The proposed method

The computational complexity of the proposed algorithm is represented by the combination of mathematical features such as entering the equation to obtain the result from each layer to facilitate the learning process represented by breaking the original algorithm, thus reducing the training time due to the fact that the process of the mutation facilitated the learning process and through the processes of breaking and selection, which accelerated the training process, thus leading to the ease and speed of the learning process of the artificial neural algorithm.

Depending on the adaptability and speed of the learning process, the suggested algorithm can be used for increasingly difficult tasks. The fuzzy logic algorithm, which can speed up the process of selecting the best neuron and avoiding potential limitations, is one way to enhance and add more features to the learning process.

6. Conclusions

A combination model comprising two AI algorithms: the EA and the ANN algorithm utilizing multi-dimensional data, the suggested approach was applied to prove the system's efficacy. The data is fed to a multi-layer FFANN, and the results are obtained from the two neurons in the output layer. The mean square error method is used to verify and assess the outcome. Depending on how many hidden layers it has, the original ANN is divided into multiple ANNs to start the optimization process. In order to determine which two

ANNs are the best and worst, the EA sorts the errors acquired from each ANN in order to perform the selection phase. The rest of the ANNs seize the opportunity to improve it through the process of mutation. Following these procedures, ANNs combine to produce one ANN, which is then fed data until the result is affirmed. Every generation undergoes these procedures. Training and testing were the two stages of the system's implementation on a variety of datasets. During the training phase, the first datasets applied were the three databases with two different types of actions for recognition employed. Each kind is assigned ten activities, each with 1061 samples. Each sample has nine features; each one is represented by three-dimensional coordinates in space. Another dataset used the HAR data extracted from smart phones to classify different human actions with 80%. Training yielded a high accuracy for aggressive and normal actions. Two different types of actions from another database were used to test the system for the first datasets suggested to be used, and the system tested on 20% of the second suggested datasets. The results were close to the training stage and satisfactory, with a high level of accuracy of the two actions.

Statements and Declarations

Funding Declaration: This research was funded by the RUDN University Scientific Projects Grant System, project No. 202235-2-000.The work was carried out within the framework of the State assignment, scientific topic code FFGU-2024-0019.

Competing Interest declaration: No conflict of interests, and there is no interest between the co-authors.

References

[1] Al-Nima R., Fawaz S., and Nathim A., "Design a Technology Based on the Fusion of Genetic Algorithm, Neural Network and Fuzzy Logic," *arXiv Preprint*, vol. arXiv:2102.08035, pp. 1-11, 2021. https://arxiv.org/abs/2102.08035

[2] Athota R. and Sumathi D., "Human Activity Recognition Based on Hybrid Learning Algorithm for Wearable Sensor Data," *Measurement: Sensors*, vol. 24, pp. 100512, 2022. https://doi.org/10.1016/j.measen.2022.100512

[3] Bashar S., Al Fahim A., and Chon K., "Smartphone Based Human Activity Recognition with Feature Selection and Dense Neural Network," *in Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Montreal, pp. 5888-5891, 2020. https://ieeexplore.ieee.org/document/9176239

[4] Bashiri A. and Hosseinkhani R., "Infant Crying Classification by Using Genetic Algorithm and Artificial Neural Network," *Acta Medica Iranica*, vol. 58, no. 10, pp. 531-539, 2020. https://doi.org/10.18502/acta.v58i10.4916

[5] Chanal D., Steiner N., Chamagne D., and Pera M., "Impact of Standardization Applied to the Diagnosis of LT-PEMFC by Fuzzy C-Means Clustering," *in Proceedings of the IEEE Vehicle Power and Propulsion Conference*, Gijon, pp 1-6, 2021. DOI:10.1109/VPPC53923.2021.9699234

[6] Colpas P., Vicario E., De-La-Hoz-Franco E., Pineres-Melo M., Oviedo-Carrascal A., and Patara F., "Unsupervised Human Activity Recognition Using the Clustering Approach: A Review," *Sensors*, vol. 20, no. 9, pp. 1-27, 2020. https://www.mdpi.com/1424-8220/20/9/2702

[7] Dalwinder S. and Birmohan S., "Feature Wise Normalization: An Effective Way of Normalizing Data," *Pattern Recognition*, vol. 122, pp. 108307, 2022. https://doi.org/10.1016/j.asoc.2019.105524

[8] Gen M. and Lin L., *Springer Handbook of Engineering Statistics*, Springer, 2023. https://doi.org/10.1007/978-1-4471-7503-2_33

[9] Human Activity Recognition with Smartphones, https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones/data, Last Visited, 2024.

[10] Joshi A., *Machine Learning and Artificial Intelligence*, Springer, 2022. https://link.springer.com/chapter/10.1007/978-3-031-12282-8_6

[11] Katoch S., Chauhan S., and Kumar V., "A Review on Genetic Algorithm: Past, Present, and Future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091-8126, 2021. https://doi.org/10.1007/s11042-020-10139-6

[12] Khan M., Alhaisoni M., Armghan A., Alenezi F., Tariq U., Nam Y., and Akram T., "Video Analytics Framework for Human Action Recognition," *Computers, Materials and Continua*, vol. 68, no. 3, pp. 3841-3859, 2021. https://www.techscience.com/cmc/v68n3/42487

[13] Kollmannsberger S., D'Angella D., Jokeit M., and Herrmann L., *Deep Learning in Computational Mechanics*, Springer, 2021. https://doi.org/10.1007/978-3-030-76587-3

[14] Li G. and Wang F., "Image Object and Scene Recognition Based on Improved Convolutional Neural Network," *The International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 925-937, 2024. https://doi.org/10.34028/iajit/21/5/13

[15] Liu X., Qin J., Zhao K., Featherston C., Kennedy D., Jing Y., and Yang G., "Design Optimization of Laminated Composite Structures Using Artificial Neural Network and Genetic Algorithm," *Composite Structures*, vol. 305, pp. 116500, 2023. https://doi.org/10.1016/j.compstruct.2022.116500

[16] Ma Z., "Human Action Recognition in Smart Cultural Tourism Based on Fusion Techniques of Virtual Reality and SOM Neural Network,"

Computational Intelligence and Neuroscience, vol. 2021, pp. 1-12, 2021. https://doi.org/10.1155/2021/3495203

[17] Meena V., Choudhary J., and Singh D., "Heart and Cardiac Arrest Analysis by Prediction of Hybrid Model," *in Proceedings of the 6th International Conference on Intelligent Computing and Control System*, Madurai, pp. 1600-1606, 2022. DOI:10.1109/ICICCS53718.2022.9788212

[18] Montesinos Lopez O., Montesinos Lopez A., and Crossa J., *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer, 2022. https://doi.org/10.1007/978-3-030-89010-0_10

[19] Nadeem A., Jalal A., and Kim K., "Human Actions Tracking and Recognition Based on Body Parts Detection via Artificial Neural Network," *in Proceedings of the 3rd International Conference on Advancements in Computational Sciences*, Lahore, pp. 1-6, 2020. https://ieeexplore.ieee.org/document/9055951

[20] Nida N., Yousaf M., Irtaza A., and Velastin S., "Video Augmentation Technique for Human Action Recognition Using Genetic Algorithm," *ETRI Journal*, vol. 44, no. 2, pp. 327-338, 2022. https://onlinelibrary.wiley.com/doi/epdf/10.4218/etrij.2019-0510

[21] Oguntimilehin A. and Balogun K., "Real-Time Sign Language Fingerspelling Recognition Using Convolutional Neural Network," *The International Arab Journal of Information Technology*, vol. 21, no. 1, pp. 158-165, 2024. https://doi.org/10.34028/iajit/21/1/14

[22] Ozcan T. and Basturk A., "Human Action Recognition with Deep Learning and Structural Optimization Using a Hybrid Heuristic Algorithm," *Cluster Computing*, vol. 23, no. 4, pp. 2847-2860, 2020. https://doi.org/10.1007/s10586-020-03050-0

[23] Pothina H. and Nagaraja K., *Smart Trends in Computing and Communications*, Springer, 2022. https://link.springer.com/chapter/10.1007/978-981-16-9967-2_21

[24] Prado F., Minutolo M., and Kristjanpoller W., "Forecasting Based on an Ensemble Autoregressive Moving Average-Adaptive Neuro-Fuzzy Inference System-Neural Network-Genetic Algorithm Framework," *Energy*, vol. 197, pp. 117159, 2020. https://doi.org/10.1016/j.energy.2020.117159

[25] Quaid M. and Jalal A., "Wearable Sensors Based Human Behavioral Pattern Recognition Using Statistical Features and Reweighted Genetic Algorithm," *Multimedia Tools and Applications*, vol. 79, pp. 6061-6083, 2020. https://link.springer.com/article/10.1007/s11042-019-08463-7

[26] Raja Subramanian R. and Vasudevan V., "A Deep Genetic Algorithm for Human Activity Recognition Leveraging Fog Computing Frameworks," *Journal of Visual Communication and Image Representation*, vol. 77, pp. 103132, 2021. https://doi.org/10.1016/j.jvcir.2021.103132

[27] Rishabh M., Pachpor K., Kottilingam K., and Saranya A., "An Initiative to Prevent Japanese Encephalitis Using Genetic Algorithm and Artificial Neural Network," *in Proceedings of the International Conference on Computational Intelligence*, Bandar Seri Iskandar, pp. 142-148, 2020. DOI:10.1109/ICCI51257.2020.9247744

[28] Savarapu P., Shankar M., Itnal S., Madupu R., Basha C., and Sreedevi E., "Advanced Prediction of Heart Diseases Using Artificial Neural Network and Genetic Algorithm," *in Proceedings of the 5th International Conference on Computing Methodologies and Communication*, Erode, pp. 875-878, 2021. DOI:10.1109/ICCMC51019.2021.9418459

[29] Shen X., Zheng Y., and Zhang R., "A Hybrid Forecasting Model for the Velocity of Hybrid Robotic Fish Based on Back-Propagation Neural Network with Genetic Algorithm Optimization," *IEEE Access*, vol. 8, pp. 111731-111741, 2020. DOI:10.1109/ACCESS.2020.3002928

[30] Singh D. and Singh B., "Investigating the Impact of Data Normalization on Classification Performance," *Applied Soft Computing*, vol. 97, pp. 105524, 2020. https://doi.org/10.1016/j.asoc.2019.105524

[31] Singh S., Sarma A., Jao N., Pattnaik A., Lu S., Yang K., Sengupta A., Narayanan V., and Das C., "NEBULA: A Neuromorphic Spin-Based Ultra-Low Power Architecture for SNNs and ANNs," *in Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, Valencia, pp. 363-376, 2020. DOI:10.1109/ISCA45697.2020.00039

[32] Stepanyan I. and Hameed S., "An Improved Neurogenetic Model for Recognition of 3D Kinetic Data of Human Extracted from the Vicon Robot System," *Baghdad Science Journal*, vol. 20, no. 6, pp. 2608-2608, 2023. https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/9087/4606

[33] Stepanyan I., "Evolutionary Operations of Interneuron Synaptic Structure for Feed-Forward Multilayer Networks," *Journal of Machinery Manufacture and Reliability*, vol. 49, no. 10, pp. 874-877, 2020. https://link.springer.com/article/10.3103/S105261882010009X

[34] Tormozov V., Zolkin A., and Vasilenko K., "Optimization of Neural Network Parameters Based on a Genetic Algorithm for Prediction of Time Series," *in Proceedings of the International Multi-Conference on Industrial Engineering and*

Modern Technologies, Vladivostok, pp. 1-4, 2020. DOI:10.1109/FarEastCon50210.2020.9271536

[35] Trelinski J. and Kwolek B., "Human Action Recognition on Raw Depth Maps," *in Proceedings of the International Conference on Visual Communications and Image Processing*, pp. 1-4, Munich, 2021. DOI:10.1109/VCIP53242.2021.9675349

[36] UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml, Last Visited, 2024.

[37] Vicon, https://www.vicon.com/, Last Visited, 2024.

[38] Wang Q. and Liu S., "Aprediction Model Analysis of Behavior Recognition Based on Genetic Algorithm and Neural Network," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-11, 2022. DOI: 10.1155/2022/3552908

[39] Wang X., He M., Yang L., Wang H., and Zhong Y., "Human Activity Recognition Based on an Efficient Neural Architecture Search Framework Using Evolutionary Multi-Objective Surrogate-Assisted Algorithms," *Electronics*, vol. 12, no. 1, pp 1-20, 2022. https://www.mdpi.com/2079-9292/12/1/50

[40] Wang X., Wang X., Lv T., Jin L., and He M., "HARNAS: Human Activity Recognition Based on Automatic Neural Architecture Search Using Evolutionary Algorithms," *Sensors*, vol. 21, no. 20, pp 1-24, 2021. https://pubmed.ncbi.nlm.nih.gov/34696140/

[41] Ye J., *Geometry of Deep Learning*, Springer, 2022. https://link.springer.com/chapter/10.1007/978-981-16-6046-7_6

[42] Yilmaz A., Guzel M., Bostanci E., and Askerzade I., "A Novel Action Recognition Framework Based on Deep-Learning and Genetic Algorithms," *IEEE Access*, vol. 8, pp. 100631-100644, 2020. DOI:10.1109/ACCESS.2020.2997962

[43] Zhang J., Sun G., Sun Y., Dou H., and Bilal A., "Hyper-Parameter Optimization by Using the Genetic Algorithm for Upper Limb Activities Recognition Based on Neural Networks," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1877-1884, 2021. DOI:10.1109/JSEN.2020.3018629

Ivan Stepanyan was born on May 1, 1980, in Moscow. In 2003, he graduated with a Master's degree in Computer Science and Engineering. In 2006 he defended his Ph.D. Thesis in Technical Sciences, and in 2012 he defended his doctoral thesis in biology. A leading researcher at the A.A. Blagonravov Institute of Mechanical Engineering of the Russian Academy of Sciences. Professor of the Engineering Academy of the Peoples' Friendship University of Russia.

Safa Hameed is Ph.D. student in Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University), Moscow, Russian Federation in the field of Artificial Intelligence, she got Bachelor's in Computer Science in 2013 and Master's degree in Computer Science in the field of Artificial Intelligence from Anbar University/Ramadi/Iraq in 2017, she has research in the field of Artificial Intelligence and she conducted her research in the field of Bioinformatics.