ISSN: 0067-2904

Evolutionary Feedforward Neural Network Algorithm and Its Application on The Example of Human Action Recognition

Ivan Stepanyan[1], Safa Hameed[2]*

[1] *Mechanical Engineering Research Institute of the Russian Academy of Sciences (IMASH RAN) 4, M. Kharitonyevskiy Pereulok, 101990, Moscow, Russian Federation*

[1,2] *Department of Mechanics and Control Processes, Academy of Engineering, Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University), 117198, Moscow, Russian Federation*

Received: 27/1/2025 Accepted: 27/5/2025 Published: 30/3/2026

Abstract

An optimization method is suggested in this paper to improve the learning process in the artificial neural network algorithm (ANN). There are three phases included in the optimisation process: the breaking ANN process, an evolutionary algorithm (EA), and the combining ANN process into a new powerful feedforward artificial neural network (FANN). ANN breaks into multiple ANNs to facilitate and accelerate the optimization process in the EA phase. EA enhanced the ANNs, starting with a selection step by sorting ANNs based on the obtained accuracy for each ANN; the crossover step exchanges the characteristics between the best and worst ANN, and the rest of the ANNs are enhanced by the mutation process. Distinguishing the type of human activity is an important topic and one of the issues that concerns society in several aspects, like health care, criminal cases, etc. The system is applied to human action datasets. A multi-dataset of human actions has been applied to the system. On the desired recognition task, the system demonstrated high performance and efficiency in a range higher than 90% for each different motion type.

Keywords: Artificial neural network, Breaking process, Evolutionary algorithm, Crossover, Mutation.

1. Introduction

The need to develop and construct an innovative computational approach to predict solutions has developed the ability to collect and analyse the data features, allowing it to spot patterns that might not be obvious at first. Consequently, this integration fosters a dynamic learning environment where continuous improvement is achievable, making it particularly valuable in finance, healthcare, and environmental modelling for the quick evolution of life [1].

An evolutionary algorithm is an extremely capable tool and the best choice to combine with the artificial neural network algorithm (ANN) [1][2]. By leveraging the strengths of both methodologies, it is possible to create a complex system capable of adapting and improving its predictions over time. This hybrid approach not only enhances accuracy but also allows

* Email : sa.programmer1@gmail.com

the model to learn from diverse datasets, ultimately leading to more robust and reliable solutions in a variety of applications [3-5].

Human action recognition (HAR) is a critical but challenging task through observation. Deep learning (DL)-based methods have successfully predicted a wide range of human activities [6][7]. In many fields, it is an important process to identify and resolve various problems [8][9]. ANN and linear discriminant analysis can be used to detect and recognize precise human actions [10]. Nevertheless, the original neural network approach has limited global search capabilities and cannot solve the problem of redundant data in behaviour recognition. For these reasons, the best recommendation is to use an evolutionary algorithm to select the ideal hyperparameters [11][12].

In this study, the structure of FANN was improved and empowered by optimizing its parameters in three crucial phases, starting with the breaking phase to accelerate the learning process. It was implemented by dividing the ANN into ANNs to evaluate each layer's weights. Then, in the EA phase, the EA model is chosen for hybridisation to accelerate and empower the ANN learning phase. The process of breaking the ANN that precedes the crossover and mutation operations of the evolutionary algorithm assists and speeds up the learning process by finding the parameters that are preferable to be used in the new generation and the parameters that need optimization, after that, ANNs combine to yield a new powerful FANN, which is the last phase.

Many contributions have worked with the combination of ANN and EA to improve the learning process. It's applied to accomplish a variety of tasks, including the recognition of human movement; a number of prior publications about the construction of hybrid systems of machine learning algorithms were examined. The technique that used a stochastic mutation function with a constant coefficient of variation was presented in [4] to address the complex combinatorial problem of structural optimization of ANN with a high dimension of the space of optimization parameters to forecast time series values. The lowest learning error in the meteorological data time series was 1.2%. Additionally, a model combining EA and ANN was presented in [10]; this model could predict and identify behaviour. The study introduces the concept of GA and advances the multidimensional coevolution technique in conjunction with ANN. The results showed that the behaviour identification based on a GA with an ANN method had 86.27 prediction accuracy and convergence, which is better. The GA yielded an accuracy of 59.22, while the standard ANN technique yielded an accuracy of 80.13.

HARNAS (Human Activity Recognition Based on Automatic Neural Architecture Searches), a technique for using neural architecture search (NAS) to identify models suitable for HAR problems, was given by the research authors in [12], the opportunity dataset was used for most of their experiments, and the UniMiB-SHAR dataset was used to evaluate the model's portability. The results showed that HARNAS can perform better than the best model with human adjustments when it is generated without human modifications. HARNAS obtained an F1 score of 92.16% with parameters of 0.32 MB on the Opportunity dataset. The research's [13] objective was to find defences against Japanese encephalitis using GA and ANN. The GA was used to construct and optimize the optimal string given the supplied data. The improvement rate was 96% while using these techniques. ANN-based surrogate models were proposed as a practical method of performing minimum weight optimization of composite laminates in [14]. Later, GA was applied to the trained ANN models to optimize the structural dimensions and stacking sequences and reduce the weight of the composite laminates. With R2 values of 0.996, 0.987, and 0.987 for the flat, blade-stiffened, and hat-stiffened laminates, respectively, the created ANN models demonstrate a high degree of

goodness-of-fit. With corresponding MAPE values of 0.023, 0.032, and 0.046, the generated ANN models can predict composite laminate buckling stresses with very small errors. R2 and MAPE comparisons between LR, DT, RF, and ANN reveal that ANN performed best in every scenario, proving its superiority in buckling load prediction. The hybrid benefits of deep learning and genetic algorithms were used in [15] to complete the activity recognition task. The genetic change detection method, which identified changes in succeeding frames' activities, was applied to the video frames taken by vision cameras. Consequently, the deep learning algorithm recognized the activity of the modified frame. The fog-assisted cloud framework served as the platform for this hybrid algorithm. GA optimisation criteria were used. With the use of three popular activity recognition data sets—Hollywood2, KTH, and UCF-ARG—the deep genetic model was able to accurately identify sounds. The accuracy of sound identification for the suggested model was 98.42%, 97.83%, and 81.40%, in that order.

Research from a number of sources indicates an effort to apply various methods to address various issues using hybrid systems. This paper aimed to demonstrate the effectiveness of the hybrid system by leveraging EA to accelerate and improve the accuracy of the ANN learning process, ultimately leading to a faster solution.

In this study, the structure of ANN is optimized by three crucial phases: breaking the original ANN into multiple ANNs, the EA process, and combining optimized ANNs to yield a new feedforward artificial neural network (FANN). The breaking process accelerates the optimization method by looking for the best and worst ANN layers to find the parameters that are preferable to be used in the new generation and the parameters that need improvement by EA; it speeds up the update process by EA with the crossover and mutation operations. The EA model was chosen for hybridisation to accelerate and empower the ANN learning phase. The model was applied to three different databases to distinguish the human action types; the system was trained to classify the different actions, and it was evaluated with a testing phase.

This paper is organized into the following sections: Section 1 introduces the background of the research and the related review, Section 2 explains the used datasets and the phases of the proposed model design, Section 3 presents the results obtained from the model implementation, and Section 4 provides the conclusion based on the findings.

2. The Methodology

Applying an evolutionary feedforward ANN with different databases is advised to improve the model's performance. Vicon Physical Action Data Set, HAR data Using Smartphones, and HARSense databases [16] – [17] were used in the suggested model. The Vicon Physical Action dataset represented aggressive and normal types. Each kind has ten actions corresponding to it; these values were obtained from the experiment's data [19]. The trial involved a number of participants (aged 25 to 30). An individual is observed participating in physical activities in a three-dimensional intelligent environment. As part of a perception-to-action unit, two external devices—a mobile robot (SCITOS G5) and a 3D tracker (Vicon system)—cooperate to give data and categorize actions taken according to mechanical features. Each type has ten activities. The activities for aggressive action are Elbowing, Front kicking, Hammering, Headering, Kneeing, Pulling, Punching, Pushing, Side kicking, Slapping. While the activities for normal action are Elbowing, Front kicking, Hammering, Headering, Kneeing, Pulling, Punching, Pushing, Side kicking, Slapping. Each activity has 1061 samples. Each sample has nine features, which are body parts (head, arm, and leg) that indicate the type of movement. Each body part has two markers (left and right, except for the head) that can indicate the type of movement. All the data in the samples, right and left arm signs represent the wrist and elbow, and right and left leg signs represent the ankle and knee. The measurements of these signs are obtained as x, y, and z coordinates, as shown in Figure 1.

HAR data Using Smartphones was used to experiment with the suggested system. The data was utilized to classify human behaviour. Thirty individuals participating in activities of daily living (ADLs) were recorded for the Human Activity Recognition database using information obtained by a smartphone fitted with an inertial sensor and worn around the waist. The dataset is multivariate and time-series in nature. Thirty people, ages 19 to 48, comprised the sample used in the investigations. Each subject was observed performing six distinct actions while the Samsung Galaxy SII smartphone was fastened around their waist: standing, sitting, lying, walking, walking downstairs, and walking upstairs. With the help of its integrated accelerometer and gyroscope, it captured 3-axial angular velocity and 3-axial linear acceleration at a constant 50 Hz. The trials were videotaped to manually identify the data. The gathered dataset was randomly split into two sets, with 30% of the participants selected to provide test data and 70% of the volunteers selected to offer training data.

The HARSense dataset was employed, comprising subject-specific daily life activity data obtained from the smartphones' built-in accelerometer and gyroscope sensors. The data is multivariate and time-series in nature. Users who weighed more than fifty kilograms and were older than twenty-three had their smartphones put on their waists and front pockets. Except running, which was done on a football pitch, every activity was done in a lab. Walking, standing, going upstairs, going downstairs, running, and sitting are among the many activities (ADLs) listed. The samples of the three different datasets are illustrated in Figure 2.

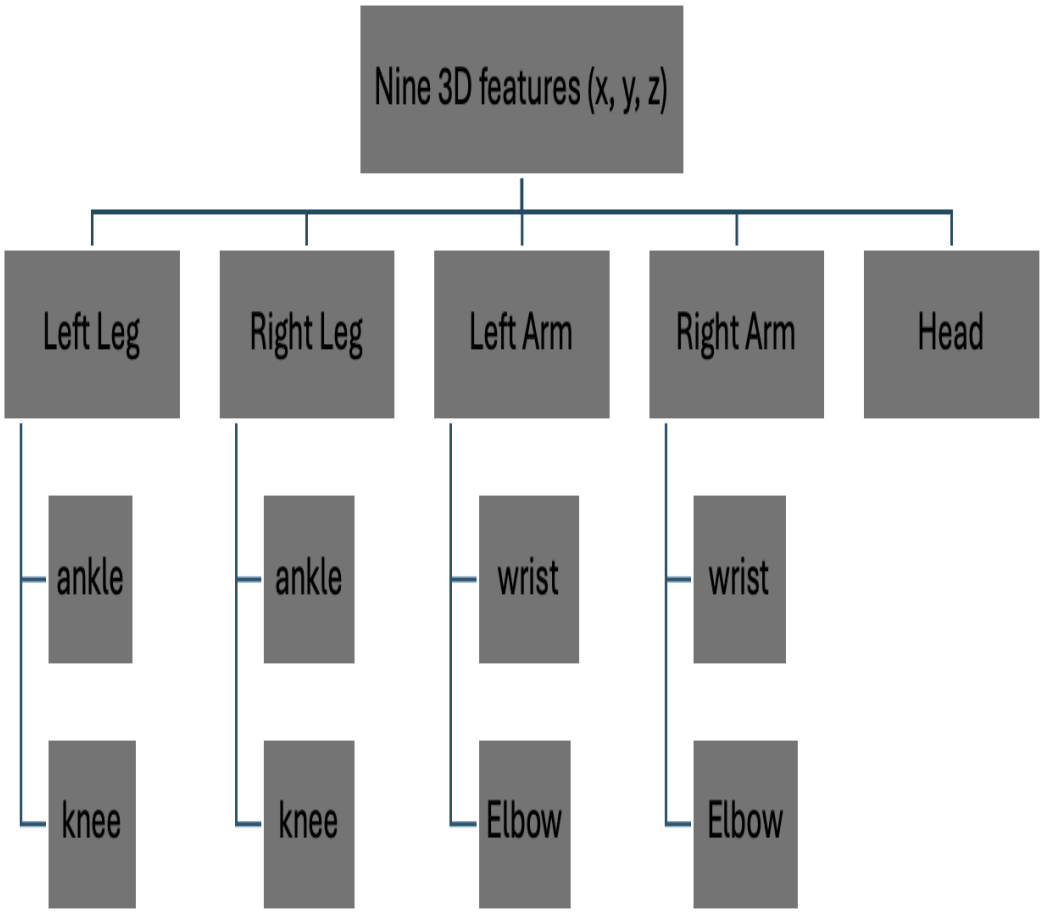

Figure 1: The 3d nine features of the first dataset

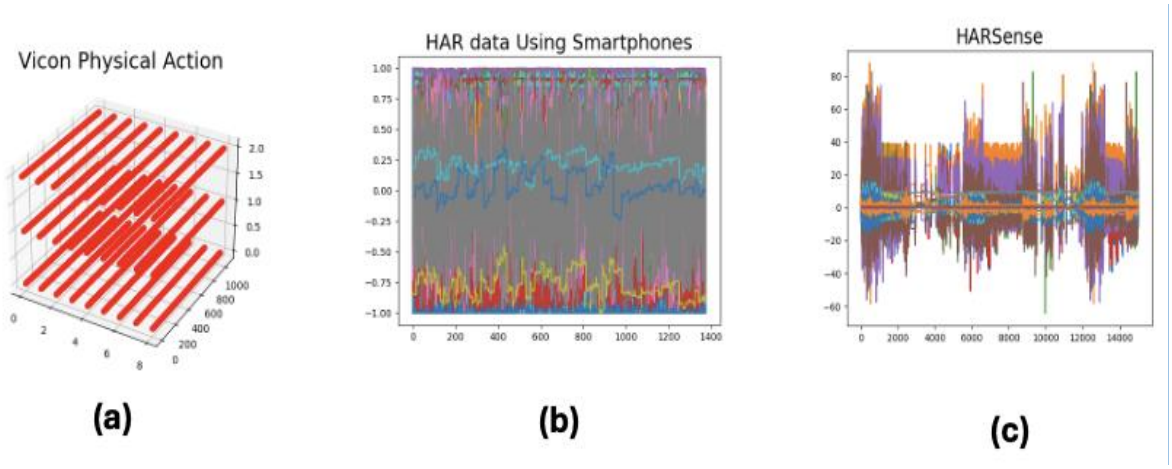

Figure 2: The three used datasets: (a) Vicon Physical Action (b) HAR data Using Smartphones (c) HARSense

To send the datasets to the model, it must undergo a normalization phase to prepare the data. To maximize the performance of the classifiers in classification, Quantile Transformer uses scikit-learn to implement the normalization function. This involves applying the scaling approach, also referred to as "Min-Max feature scaling," which consists in scaling data in the interval [0–1]. It involves rescaling the data and using the minimum and maximum values as boundaries. It enables scaling of a vector x in the interval [0, 1] [20]– [22] in mathematical terms, as in (1), where min and max represent the characteristic x's lowest and highest values, respectively [22]. To make the data easier to work with, it was normalized to fall between zero and one.

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

x_i: the normalized sample
x_{max}: the maximum value from the samples
x_{min}: the minimum value from the samples

After the normalization step, the data is ready to feed into the ANN. After assessing the result of the algorithm, the outcomes are compared with the desired output to complete the learning process. The learning process starts with a breaking phase, which operates by dividing the ANN into multiple ANNs to accelerate the optimisation process in the EA phase, which follows it. In the EA phase, implement the crossover process between the best and worst ANNs that are chosen by the selection step, and enhance the worst ANNs by the mutation process. The combining process is necessary to integrate the enhanced ANNs into the new optimized ANN.

2.1 The structure of FANN

FANN is composed of the input layer, eleven hidden layers, and an output layer; the size of the input data is used to form a matrix of weights of the first hidden layer, which is then multiplied by the input, added with the bias, Vicon Physical Action dataset, the used sample in the dataset consists of nine features with three dimensions for each, the data fed to the first hidden layer neuron as in Figure 3 (a). The nine three-dimensional features multiplied with the corresponding weight as in (2), while in the HAR data using Smartphones, and HARSense databases, the nature of the data is multivariate and time-series. The used sample in the dataset consists of one dimension for each multiplied by the corresponding weight as in (3), the data fed to the first hidden layer neuron as in Figure 3 (b). and the outputs are sent to the first hidden layer neurons for processing, and then proceed to the rest of the hidden layers.

There are three neurons in each of the eleven hidden layers. As in (4) and (5), the output from every neuron is derived by using the sigmoid function. As seen in (6), the identity function transfers the output from the final hidden layer to the output layer, which is a combined output that classifies activities. The outputs are binary classifications of 0 and 1. Mean square error is used to compare the output with the real result, as in Eq. (7).

$$S = \left(\sum_{i=1}^{n}\sum_{j=1}^{c} x_{ij}.w_{ij}\right) + b \quad (2)$$

$$s = \sum_{i=1}^{n} W_i.X_i + b \quad (3)$$

$$\sigma = \frac{1}{1+e^{-s}} \quad (4)$$

$$neuron = \sigma(\hat{y}) = \sigma(\sum_{i=1}^{n} W_i.X_i + b \quad (5)$$

$$\hat{z} = \sum_{i=1}^{n} W_i.neuron_i + b \quad (6)$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(z_i - \hat{z}_i)^2 \quad (7)$$

x_{ij}, w_{ij}: The index of the feature and the weight for n features with c coordinates for the first dataset.
W_i, X_i : The index of the feature and the weight for n features for the second and third datasets.
b: The bias.
σ: The activation function (sigmoid),
neuron: the output from the neuron.
$\hat{z}$: The output from FANN for each sample of data.
z: The desired output.
MSE: The mean square error.
m: The total number of samples in the dataset.
m represents the number of features indicated by the index i.
n: the number of coordinates x, y, and z, indicated by the index j.

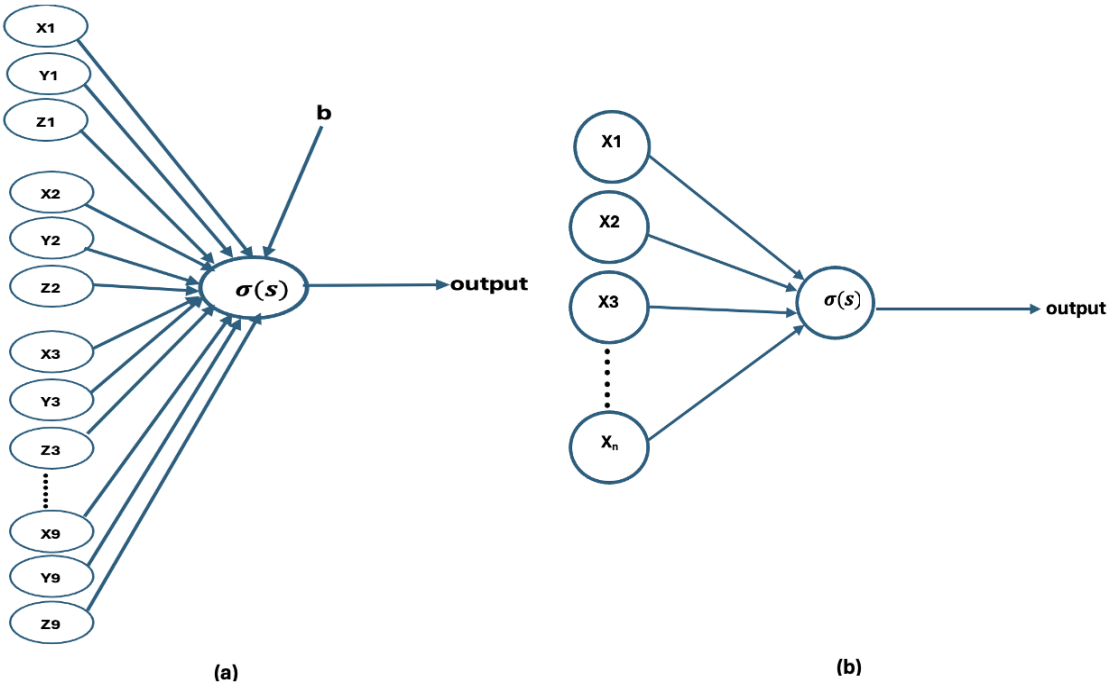

Figure 3:

Then, the computing output from each neuron is sent to the rest of the hidden layers to be processed, for Vicon Physical Action dataset, the output layer produces two classes of output, which are aggressive and normal. If unsatisfactory results are obtained, the learning process is required to optimize the parameters, as will be discussed later. The coefficient weights of neurons in each hidden layer changed the structure of FANN after the learning process. Some

of the coefficient weights of neurons in each hidden layer will be active for one class and nonactive for another, and vice versa. We assume that the active coefficient weights produce the intended class sent by the synapse to active neurons, and nonactive neurones receive the active coefficient weights for the other class, in Figure 4, an example of FANN trained from the Vicon Physical Action dataset with only two hidden layers with two neurons for each class, there are two classes of data fed to FANN; The neural network is trained using the complete set of data, and the system is trained using each sample from each activity that corresponds to the specified action. Each input is associated with a different weight, and each neuron receives the output from the preceding neuron multiplied by a different weight. The active neurons receive the data to produce the class to which it belongs to. The neuron in the first hidden layer receives nine 3D features of the sample. FANN receives the dataset, which consists of two classes; there are 10 activities in each class, and each activity consists of 1061 samples.

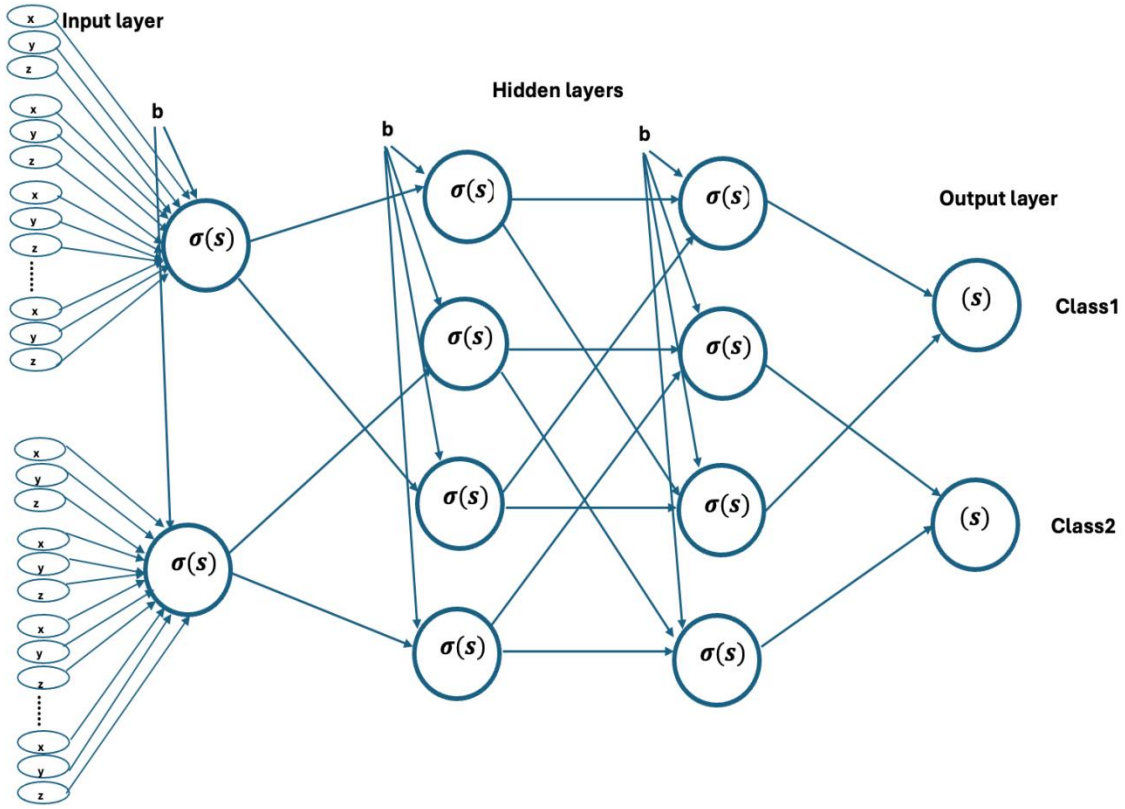

Figure 4: The feedforward phase of the first dataset for one sample of each class.

2.2. The learning process

The learning process consists of three phases, which are necessary if the obtained results are inadequate.

2.2.1. The breaking up phase

To perform the optimization phase, the ANN algorithm was divided into multiple ANNs according to the number of hidden layers, by assuming an output layer after each hidden layer, as in Figure 5 (a). The output from each ANN is computed by the identity function, as in (8), and it then calculates the mean square error (MSE) after getting the output from each new ANN, as in (9). Every learning phase iteration's population of ANNs obtained through the breaking process is regarded as an individual ANN within the population. The errors

obtained from each ANN are stored in a list, as shown in (10), which is regarded as a fitness list for use in the necessary optimization stages [23]-[25].

$$ANN_i = \sum_{j=1}^{r} w_j^i . \text{neuron}_\text{j} + b \quad (8)$$

$$MSE_{ANN_i} = \frac{1}{m} \sum_{i=1}^{m} (z_i - ANN_i)^2 \quad (9)$$

$$Error_{ANN_s} = \left[MSE_{ANN_1}, MSE_{ANN_2}, .., MSE_{ANN_{11}}\right] \quad (10)$$

j: the index of feature for each sample of data.
r: the number of neurons.

2.2.2. The evolutionary algorithm role

EA has been presented to optimise parameters. There were phases to this process. The first step is to adopt the selection stage, the layer with the best weight is investigated. The process of dissecting the algorithm into multi-ANNs made it easier to look at the layers with the best weights, as in Figure 5 (b). The efficacy of ANNs is assessed using the list of saved errors, as in (10). Sorting ANNs are used to accomplish the ranking process, which ranks items from best to worst, as in (11).

$$ANN_i \leq ANN_{i-1} \leq ANN_{i-2} \leq \cdots \leq ANN_{i-n} \quad (11)$$

The best classes have the chance to endure and be utilized by the following generation during the crossover stage. The most effective layers with the lowest error were identified by the ranking procedure used in the fitness list, and they are listed first and then switched with the two levels with the greatest error values, as in (12) and (13). Figure 5 (c) depicts the crossover process.

$$\begin{aligned} F(ANN_{11}, ANN_1) &= F(ANN_{11}(W_{ijk}), ANN_1(G_{ijk})) \\ &= ANN_{11}(W_{ijk}) \Leftrightarrow ANN_1(G_{ijk}) \\ &= ANN_{11}(G_{ijk}), ANN_1(W_{ijk}) \end{aligned} \quad (12)$$

$$\begin{aligned} F(ANN_{10}, ANN_2) &= F(ANN_{10}(W_{ijk}), ANN_2(G_{ijk})) \\ &= ANN_{10}(W_{ijk}) \Leftrightarrow ANN_2(G_{ijk}) \\ &= ANN_{10}(G_{ijk}), ANN_2(W_{ijk}) \end{aligned} \quad (13)$$

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

By randomly altering the weights of the neurons and layers, as illustrated in Figure (5)(d), the mutation step enhances the residual ANNs with poor accuracy; however, the random alteration is constrained. These bounds were set in the opposite direction of the obtained error, which was derived from the preceding generation's error. We try to select values that are in opposition to the weights' values with the biggest error value.

4.2.3. The Combining Process

ANNs are integrated into a new FANN algorithm after the necessary enhancement processes are finished. The final stage of the evolutionary algorithm is the mutation process, which is followed by the combining process. As shown in (14), the combining process is carried out by combining the final hidden layer of each ANN algorithm to create a new feedforward artificial neural network algorithm FANN algorithm. Following the combining process, a new FANN with 11 hidden layers is created as in Figure 5 (e). The most effective hidden layers of the new FANN algorithm have been tested and shown to be capable of producing highly accurate classification. The flowchart of the neuro-evolutionary algorithm is represented in Figure 6.

$$FANN_{new}=f(Na1(G_{ijk})).f(Na2(W_{ijk}))\ldots f(Na11(Z_{ijk})) \quad (14)$$

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

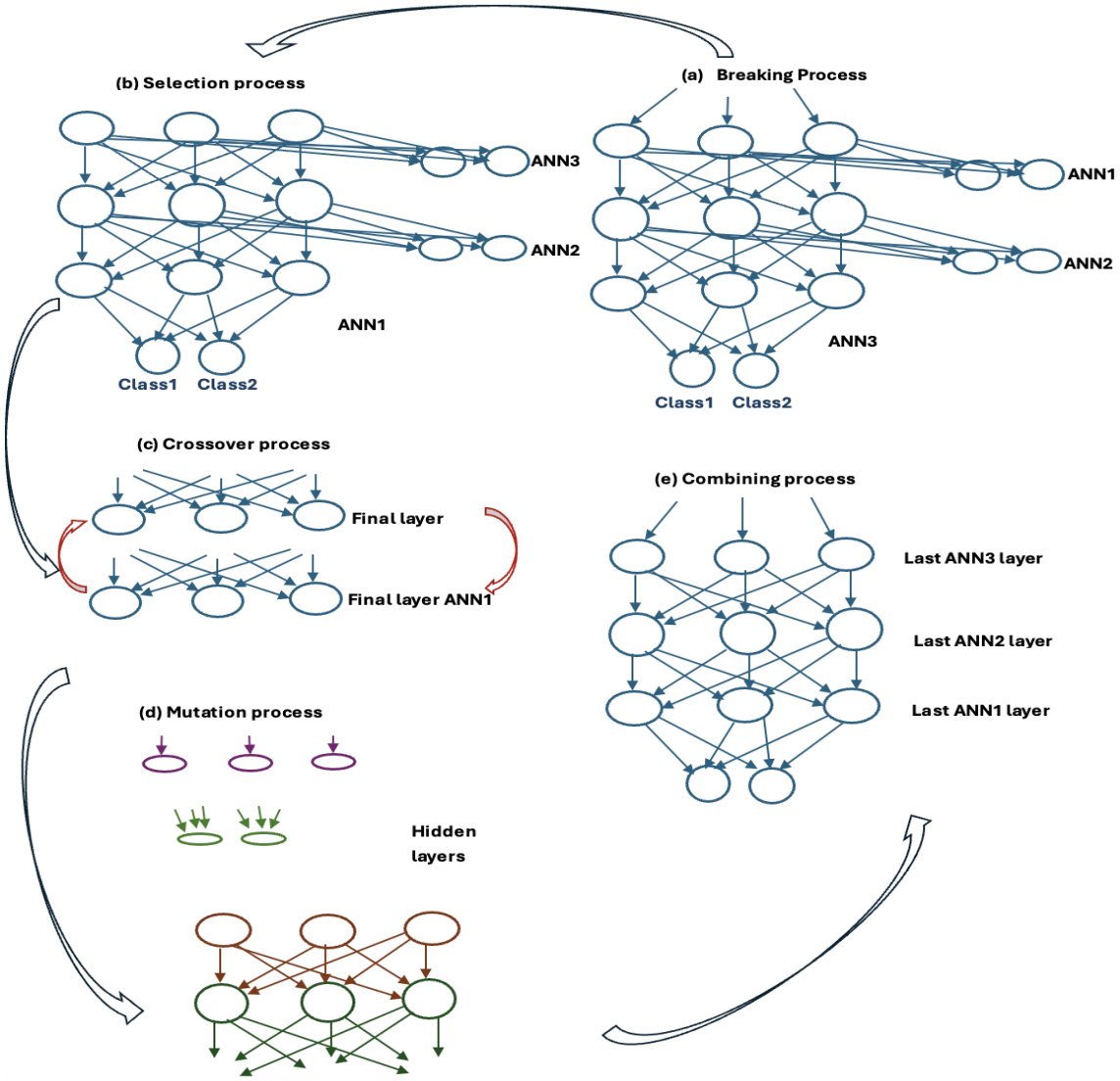

Figure 5: The learning phase of ANN with EA

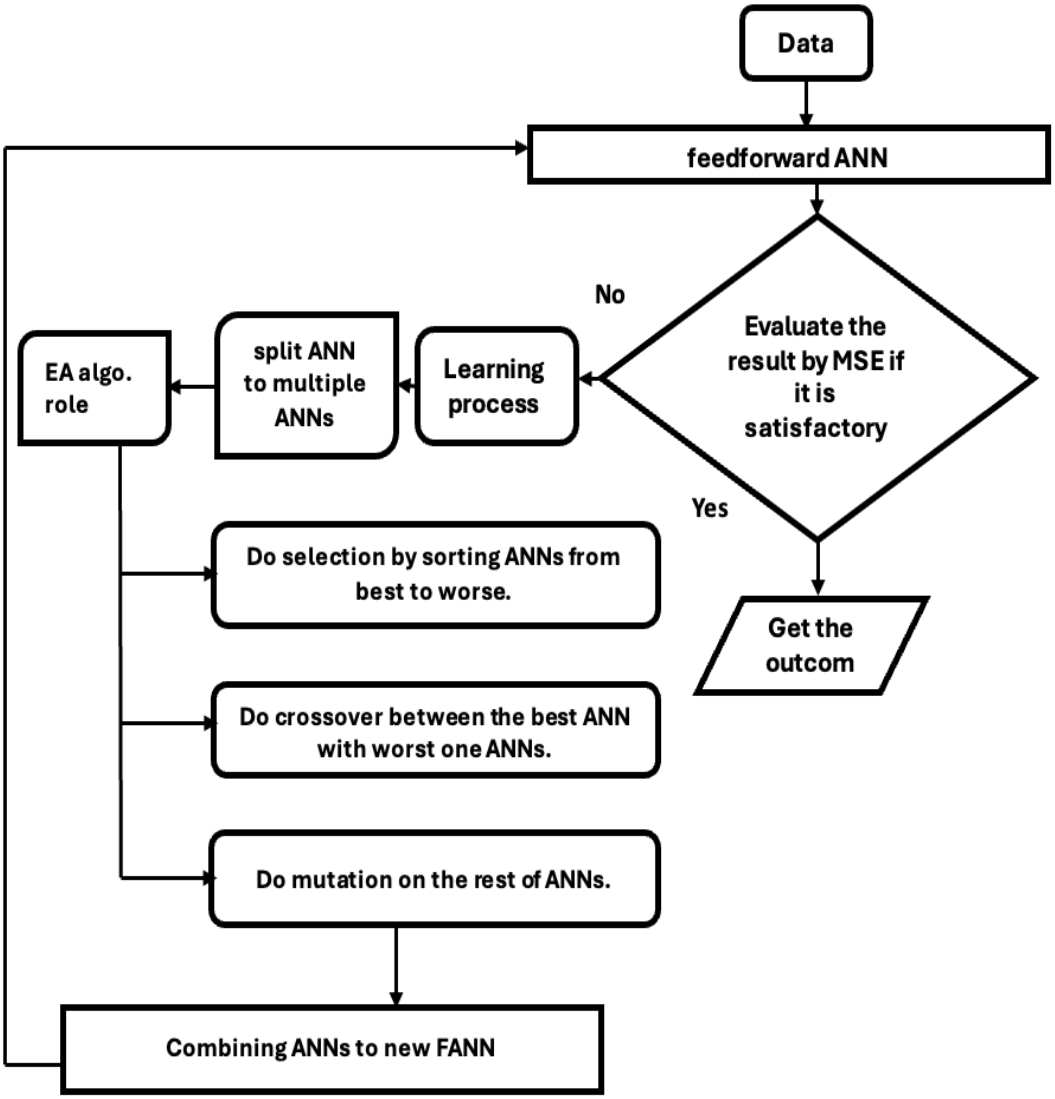

Figure 6: The flowchart of optimizing FANN by the EA model

5. The Results

The system was applied to three different data sets on human action recognition, each of which obtained satisfying results. Each implementation is divided into two stages: training and testing.

In the training phase, the model trained on 80% from each of the three different datasets which are Vicon Physical Action, HAR data Using Smartphones, and HARSense datasets [16]-[17], as mentioned in the methodology section. For the Vicon Physical Action Data Set, the data sampling by the Vicon system gives data and categorises actions taken according to mechanical features. Two types of actions, which are aggressive and normal, each have ten activities in 1061 samples for each. Four databases were gathered; three databases were used for training, and the fourth one was used for the testing phase. The HAR data was obtained using the Smartphones dataset, the data obtained by a smartphone fitted with an inertial sensor and worn around the waist. The dataset is multivariate and time-series in nature. The HARSense dataset was obtained from the smartphones' built-in accelerometer and gyroscope sensors; the data is multivariate and time-series in nature, with 80% used for training and 20% used for testing for both of the last two datasets using the sklearn.model_selection tool.

5.1 Training phase

The system was trained on the Vicon Physical Action dataset. Each one includes two types of aggressive and normal actions, with ten activities for each, which were used to train the system. Multidimensional data that is stored in three databases and passed as multidimensional data represents both typical and aggressive types of activities. To verify the outcome, the artificial neural front-end algorithm received data from the three databases. When unsatisfactory results are obtained, the optimization process starts with a breaking process to do the selection operation by sorting ANNs' accuracy from high to low, as explained in Table 1. It is for the last generation for the optimization phase. Figures. Figure 7 shows the accuracy of each ANN in the breaking phase for both aggressive and normal action, respectively.

Table 2 explains the accuracy and MSE for the three databases from the first dataset for both aggressive and normal actions, respectively. Figure. 8. Illustrates the obtained accuracy of each activity in each action.

The system was trained on the HAR data using the Smartphones dataset with 80% of the data. Table 3 explains the accuracy of ANNs in the breaking phase. Table 4 explains the accuracy and MSE for each action from the second dataset. Figure. 9. Illustrates the obtained accuracy of each ANN in the breaking phase and for each action.

Table 1: The accuracy for each ANN in breaking and selection phases for first dataset

Breaking phase				**Selection phase**			
Aggressive action		**Normal action**		**Aggressive action**		**Normal action**	
ANNs	**Accuracy**	**ANNs**	**Accuracy**	**ANNs**	**Accuracy**	**ANNs**	**Accuracy**
1	80.5	1	99.1	9	93.6	1	99.1
2	93.5	2	98.7	8	93.6	2	98.7
3	93.6	3	98.55	3	93.6	3	98.55
4	83.95	4	98.5	7	93.6	4	98.5
5	69.7	5	98.5	2	93.4	5	98.5
6	93.2	6	98.5	6	93.2	6	98.5
7	93.6	7	98.5	4	83.95	7	98.5
8	93.6	8	98.5	10	83.9	8	98.5
9	93.6	9	98.5	11	83.85	9	98.5
10	83.9	10	98.5	1	80.5	10	98.5
11	83.85	11	98.5	5	69.7	11	98.5

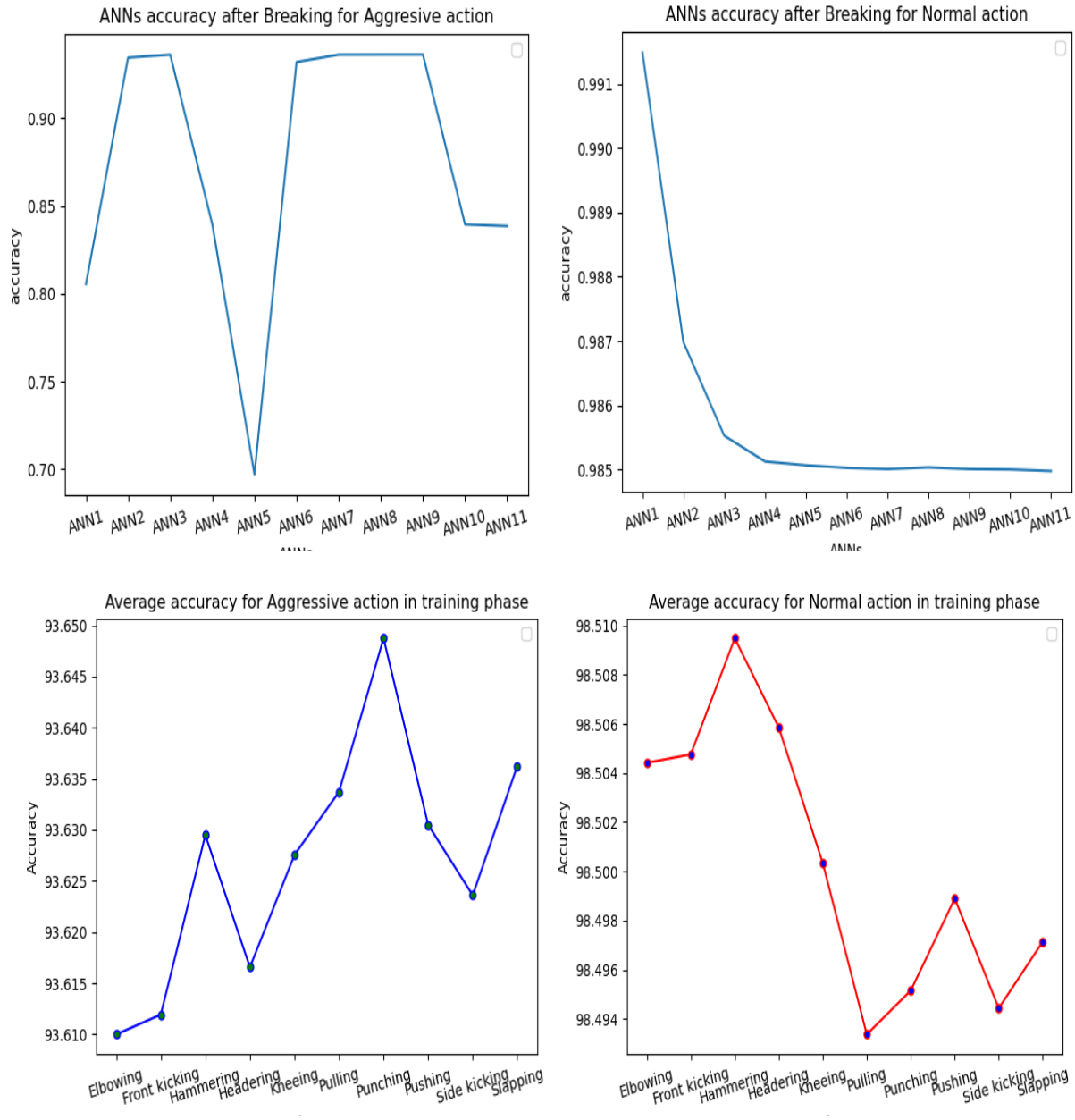

Figure 8: The aggressive and normal action average accuracy in the training phase for the first dataset

Table 2: The Mean Squared Error and accuracy values for aggressive and normal actions for the three databases of the first dataset of the training phase

Aggressive action				Normal action			
Action type	**Mean Squared Error**			**Action type**	**Mean Squared Error**		
	1st DB	**2nd DB**	**1st DB**		**1st DB**	**2nd DB**	**1st DB**
Elbowing	0.0639	0.064	0.0639	Elbowing	0.014956	0.0149558	0.014956
Front kicking	0.0638	0.064	0.06388	Front kicking	0.014952	0.01495	0.01495
Hammering	0.0637	0.0637	0.0637	Hammering	0.0149	0.0149	0.0149
Headering	0.0638	0.0638	0.0638	Headering	0.01494	0.01494	0.01494
Kneeing	0.0637	0.0637	0.0637	Kneeing	0.014995	0.014997	0.0149967
Pulling	0.0636	0.0636	0.06367	Pulling	0.0150663	0.015067	0.015067
Punching	0.0635	0.0635	0.0635	Punching	0.015049	0.015048	0.0150486
Pushing	0.0637	0.0637	0.0637	Pushing	0.015	0.015	0.015
Side kicking	0.0638	0.06376	0.06376	Side kicking	0.015	0.015	0.015
Slapping	0.0636	0.0636	0.06364	Slapping	0.015	0.015	0.015
Action type	**Accuracy %**			**Action type**	**Accuracy %**		
	1st DB	**2nd DB**	**1st DB**		**1st DB**	**2nd DB**	**1st DB**
Elbowing	93.60	93.6	93.6	Elbowing	98.50	98.504	98.5
Front kicking	93.6	93.61	93.61	Front kicking	98.505	98.505	98.505
Hammering	93.63	93.63	93.63	Hammering	98.51	98.51	98.51
Headering	93.62	93.62	93.62	Headering	98.506	98.506	98.506
Kneeing	93.63	93.628	93.63	Kneeing	98.5	98.5	98.5
Pulling	93.6	93.63	93.6	Pulling	98.493	98.493	98.493
Punching	93.65	93.65	93.65	Punching	98.495	98.495	98.495
Pushing	93.63	93.63	93.63	Pushing	98.499	98.499	98.4989
Side kicking	93.6	93.62	93.62	Side kicking	98.494	98.49	98.49
Slapping	93.6	93.64	93.64	Slapping	98.497	98.497	98.497
Accuracy	93.63	93.63	93.63	Elbowing	98.5	98.5	98.5

Table 3: The accuracy for each ANN in the breaking and selection phases for second dataset

Breaking phase		Selection phase	
ANN	**Accuracy**	**ANN**	**Accuracy**
1	98.397	1	98.4
2	97.179	7	98.03
3	97.89	3	97.9
4	97.8	9	97.8
5	97.8	5	97.8
6	95.74	4	97.8
7	98.03	10	95.8
8	97.8	8	97.8
9	97.83	2	97.2
10	95.7	11	95.4
11	95.4	10	95.8

The system was trained on the HARSense dataset with 80% of the data. Table 5 explains the accuracy of ANNs in the breaking phase. Table 6 explains the accuracy and MSE for each action from the third dataset. Figure. Figure 10 illustrates the obtained accuracy of each ANN in the breaking phase and for each action.

Table 4: The Mean Squared Error and accuracy values of the second dataset

Action type	Mean Squared Error	Accuracy
Standing	0.043	95.66296
Sitting	0.0458	95.41818
Laying	0.0447	95.528
Walking	0.04478	95.52177
Walking downstairs	0.04478	95.52177
Walking downstairs	0.045478	95.45217
Accuracy	95.543369	

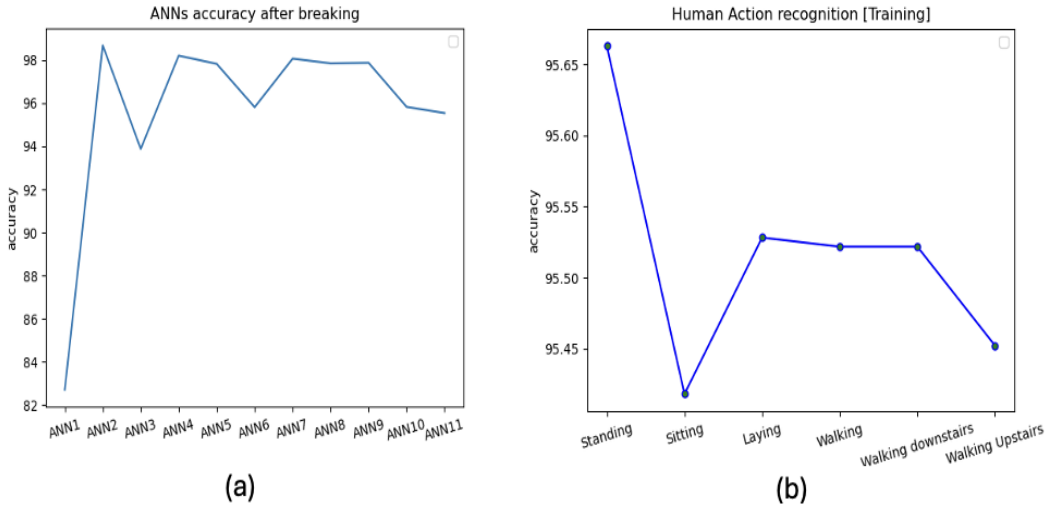

Figure 9: The accuracy of (a) ANNs breaking phase, (b) for each action in the second dataset

Table 5: The accuracy for each ANN in the breaking and selection phases for the third dataset

Breaking phase		Selection phase	
ANN	**Accuracy**	**ANN**	**Accuracy**
1	98.397	1	98.397
2	97.179	7	98.03
3	97.89	3	97.893
4	97.8	9	97.83
5	97.83	5	97.83
6	95.74	4	97.8
7	98.03	10	95.83
8	97.796	8	97.796
9	97.828	2	97.179
10	95.745	11	95.4448
11	95.445	10	95.745

Table 6: The Mean Squared Error and accuracy values of the third dataset

Action type	Mean Squared Error	Accuracy
Running	0.04337	95.45119
Standing	0.0458	95.48376
Sitting	0.0447	95.418556
Walking	0.04478	95.3925
Going up stairs	0.04478	95.45909
Going Downstairs	0.045478	95.463425
Accuracy	95.444756	

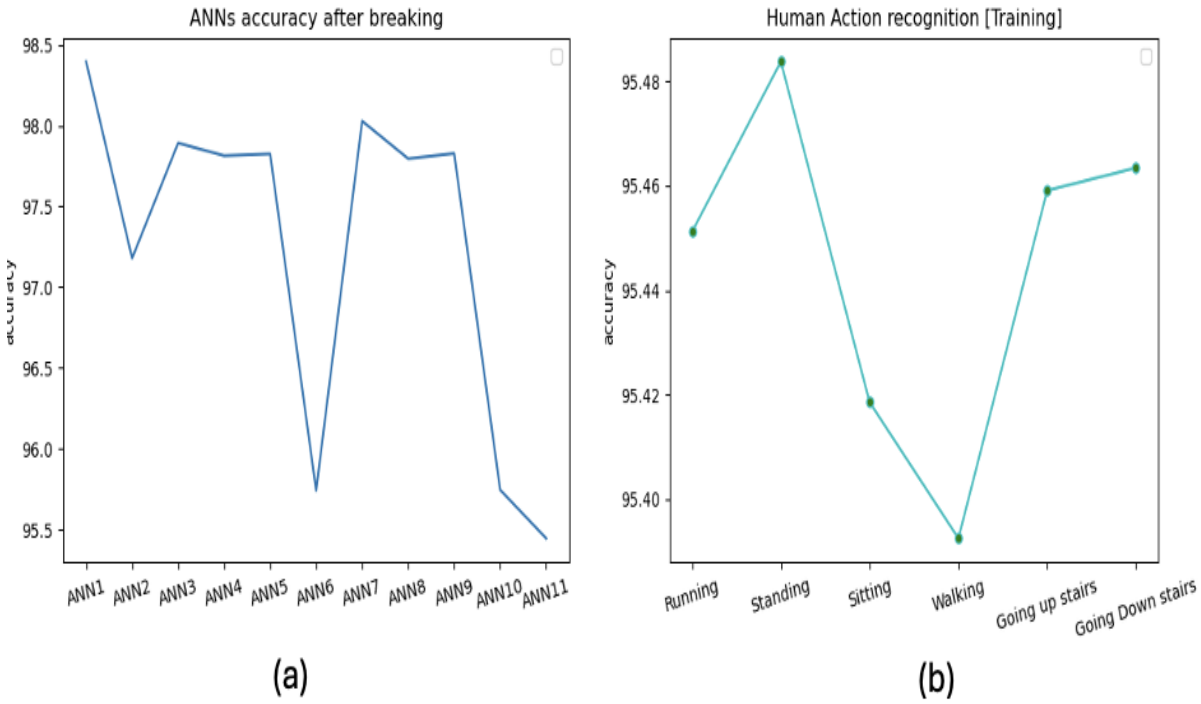

Figure 10: The accuracy of (a) ANNs inbreaking phase, (b) for each action in the third dataset

5.2 Testing phase

After obtaining satisfying results in the training phase, the system was tested with other data from Vicon Physical Action, HAR data Using Smartphones, and HARSense datasets. The system was tested on the fourth database from the Vicon Physical Action dataset. Each one includes two types of aggressive and normal actions, with ten activities for each, which were used to train the system. In Table 7. Explains the accuracy and MSE for the fourth database from the first dataset for both aggressive actions and normal actions, respectively.

The system was tested on the HAR data using the Smartphones dataset, with 20 % of the data. Table 8 explains the accuracy and MSE for each action from the second dataset.

The system also tested 20 % of the data from the HARSense dataset. Table 9 explains the accuracy and MSE for each action from the third dataset. Figure 11. Illustrates the obtained accuracy of (a) each activity in aggressive action, (b) each activity in normal action, (c) each activity in the second dataset, and (d) each activity in the third dataset.

Table 7: The Mean Squared Error and accuracy values for aggressive and normal actions of the fourth databases of the first dataset of testing phase

Aggressive action			Normal action		
Action type	**Mean Squared Error**	**Accuracy**	**Action type**	**Mean Squared Error**	**Accuracy**
Elbowing	0.0639	93.61	Elbowing	0.014956	98.504
Front kicking	0.06389	93.61	Front kicking	0.01495	98.5048
Hammering	0.0637	93.63	Hammering	0.014905	98.5095
Headering	0.0638	93.6166	Headering	0.0149	98.506
Kneeing	0.0637	93.6276	Kneeing	0.014997	98.5003
Pulling	0.06366	93.634	Pulling	0.015066	98.493
Punching	0.0635	93.6487	Punching	0.015048	98.495
Pushing	0.0637	93.6305	Pushing	0.015011	98.4989
Side kicking	0.06376	93.6237	Side kicking	0.015056	98.494
Slapping	0.0636	93.6363	Slapping	0.015029	98.497
Accuracy	93.62684		Accuracy	98.50039	

Table 8: The Mean Squared Error and accuracy values for second dataset of testing phase

Action type	Mean Squared Error	Accuracy
Standing	0.04262	95.7378677
Sitting	0.045956	95.404401
Laying	0.046058	95.3941569
Walking	0.04589	95.4117168
Walking downstairs	0.042355	95.764464
Walking downstairs	0.04718857	95.2811434
Accuracy	95.498958312	

Table 9: The Mean Squared Error and accuracy values for third dataset of testing phase

Action type	Mean Squared Error	Accuracy
Running	0.04458	95.54154
Standing	0.04367	95.633
Sitting	0.04614	95.3859
Walking	0.046966	95.3034
Going up stairs	0.04652	95.3482
Going Downstairs	0.04596	95.4042
Accuracy	95.436101	

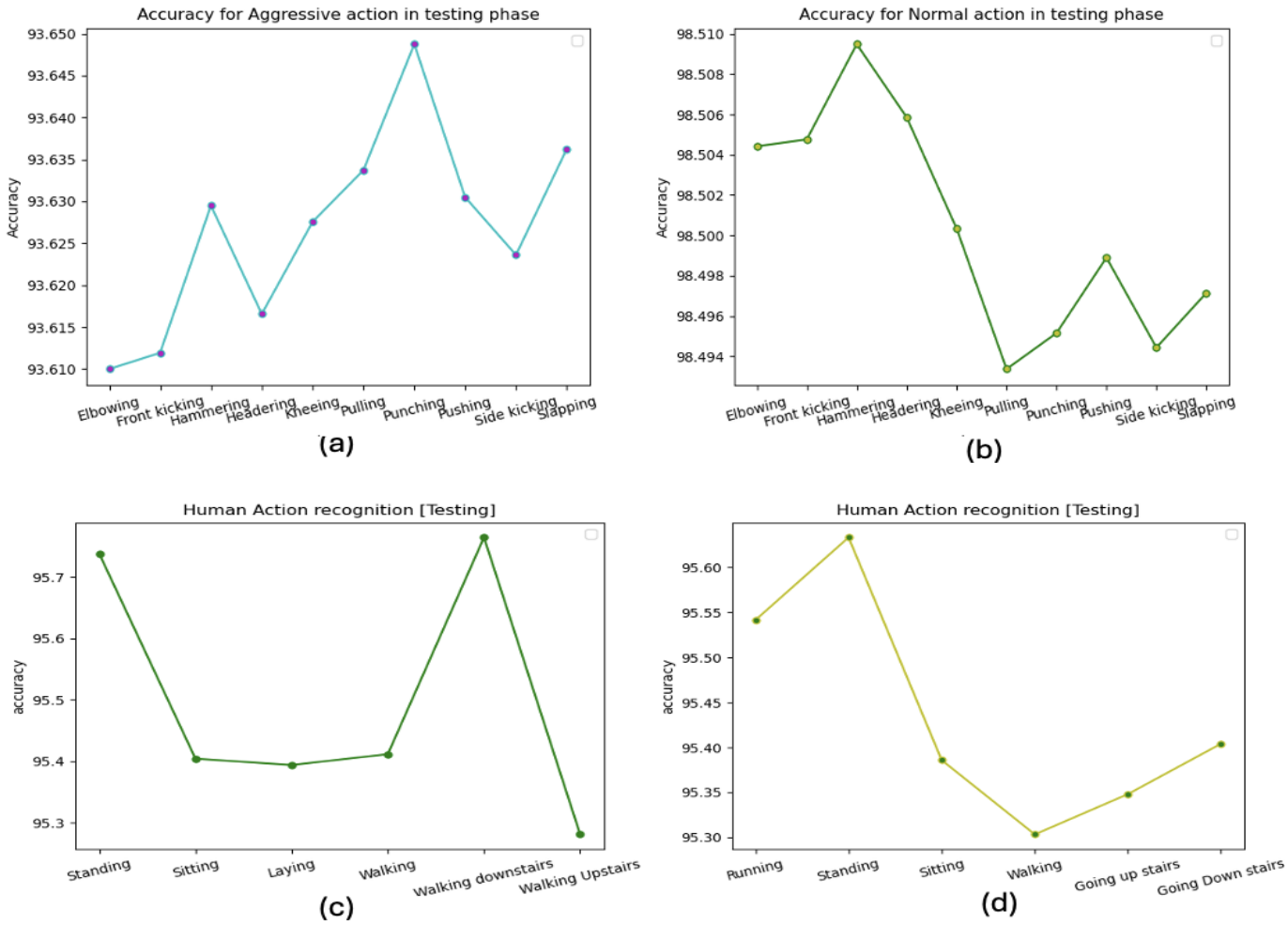

Figure 11: The testing phase accuracy of (a) the first dataset for aggressive action, (b) first dataset for normal action (c) the second dataset, (d) the third dataset

The implementation was evaluated for each iteration in the training phase by computing the mean square error (MSE) and measuring the accuracy for each sample in the different datasets used. The MSE and accuracy are obtained as explained in each table of 2, 4, and 6.
In the testing phase, the evaluation was compared with the last iteration in the training phase by computing the MSE and the accuracy for each dataset.
The suggested work performance was contrasted with that of another earlier research, as shown in Table 10.

Table 10: The comparison of the proposed work with other previous studies.

The ref, year	The method	The dataset	The accuracy
4, 2020	ANN AND GA	Time series data	learning error 1.2%
6, 2021	raw depth maps	SU 3DHOI dataset	92.16%
7, 2022	GA	MuHAVi-Uncut, iXMAS, and IAVID-1	81.25%
9, 2020	GA	HMP, WISDM and self-annotated IMSB datasets	85.43%
10, 2022	GA with ANN	sports behavior dataset	86.27%
12, 2020	Opportunity dataset, and UniMiB-SHAR dataset.	HARNAS (Human Activity Recognition Based on Automatic Neural Architecture Searchs)	92.16% 92.16%
15, 2021	A deep GA	Hollywood2, KTH, and UCF-ARG	98.42%, 97.83%, and 81.40%
The proposed method	ANN and EA	Vicon Physical Action Data Set, HAR data Using Smartphones, and HARSense	1st dataset 93%, 98%, 2nd and 3rd 95%

6. Conclusion

This combined model integrates two methods of artificial intelligence: the artificial neural network algorithm and the evolutionary algorithm. The effectiveness of the system was demonstrated by applying the proposed method to three databases. A feedforward artificial neural network with multiple layers receives the data. Verification and evaluation of the result are done using the mean square error approach. The initial ANN is split up into several ANNs to begin the optimization process, depending on how many hidden layers it has. To carry out the selection phase, the evolutionary algorithm classifies the errors obtained from each ANN to identify which two are the best and worst. Through the process of mutation, the remaining ANNs take advantage of the chance to enhance it. After these steps, ANNs merge to form a single ANN, which is further fed data until the outcome is confirmed. These processes are carried out in every generation. The system was implemented in two stages: training and testing, which produced a high accuracy rate on a range of datasets; the model was compared with other previous studies.

7. Statements and Declarations

The work was carried out within the framework of the State assignment, the code of the scientific topic FFGU-2024-0019.

Conflict of Interest: The authors declare that they have no conflicts of interest.

References

[1] Gen, M. and Lin, L. "*Genetic Algorithms and Their Applications* in *Engineering Statistics"*. Springer. 2023. pp. 635–674. https://doi.org/10.1007/978-1-4471-7503-2_33.

[2] K. Stefan, D. Davide, J. Moritz, and H. Leon, "*Deep Learning in Computational Mechanics* in *Computational Intelligence",* Springer, 2021. https://doi.org/10.1007/978-3-030-76587-3.

[3] C. Damien, S. Y. Nadia, C. Didier, and P. Marie-Cécile, "*Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering"*, VPPC. IEEE., 2021, DOI: 10.1109/VPPC53923.2021.9699234.

[4] V.S. Tormozov, A.L. Zolkin, and K.A. Vasilenko, "*Optimization of neural network parameters based on a genetic algorithm for prediction of time series"*, FarEastCon. IEEE. 2020, DOI: 10.1109/FarEastCon50210.2020.9271536.

[5] S. V. Ivan, "*Evolutionary operations of interneuron synaptic structure for feed-forward multilayer networks"*, J Mach Manuf Reliab, (2020), 49(10):874–7. https://doi.org/10.3103/S105261882010009X.

[6] K. A. Muhammad, A. Majed, A. Ammar, A. Fayadh, T. Usman, N. Yunyoung, and A. Tallha, "*Video Analytics Framework for Human Action Recognition"*, Comput Mater Contin, 68(2021), pp. 3841-3859, DOI:10.32604/cmc.2021.016864.

[7] N. Nudrat, Y. H. Muhammad, I. Aun, and V. A. Sergio, "*Video augmentation technique for human action recognition using genetic algorithm"*, ETRI 44(2022), pp. 327-338, https://doi.org/10.4218/etrij.2019-051.

[8] S. V. Ivan and H. A. Safa, "*An improved neurogenetic model for recognition of 3D kinetic data of human extracted from the Vicon Robot system"*, Baghdad Sci.J. 20.6 (Suppl.): 2608-2608, (2023), https://doi.org/10.21123/bsj.2023.9087.

[9] Q. K. A. Majid and J. Ahmad, "*Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm"*, Multimed Tools Appl, 79(2020), pp. : 6061-6083, https://doi.org/10.1007/s11042-019-08463-7.

[10] W. Qifu and L. Shuzhi, "*A prediction model analysis of behavior recognition based on genetic algorithm and neural network"*, Comput Intell Neurosci, (2022), https://doi.org/10.1155/2022/3552908.

[11] W. Xiaojuan, H. Mingshu, W. Hui, and Z. Hui," *Human Activity Recognition Based on an Efficient Neural Architecture Search Framework Using Evolutionary Multi-Objective Surrogate-Assisted Algorithms"*, Electronics 12(2022), https://doi.org/10.3390/electronics12010050.

[12] W. Xiaojuan, W. Xinlei, L. Tianqi, J. Lei, and H. Mingshu, "*Harnas: human activity recognition based on automatic neural architecture search using evolutionary algorithms"*, Sensors, 21(2021), https://doi.org/10.3390/s21206927.

[13] M. Rishabh, P. Kaustubh, K. Kottilingam, and A. Saranya, "*An initiative to prevent Japanese encephalitis using genetic algorithm and artificial neural network"*, ICCI. IEEE, 2020, DOI: 10.1109/ICCI51257.2020.9247744.

[14] L. Xiaoyang, Q. Jian, Z. Kai, F. A. Carol, K. David, J. Yucai, and Y.Guotao, "*Design optimization of laminated composite structures using artificial neural network and genetic algorithm"*, Compos Struct, (305) 2023, https://doi.org/10.1016/j.compstruct.2022.116500.

[15] R. S. Raja and V. Vasudevan, "*A deep genetic algorithm for human activity recognition leveraging fog computing framework"s*, J Vis Commun Image Represent. 77 (2021), https://doi.org/10.1016/j.jvcir.2021.103132.

[16] T. Theodoridis, *Vicon Physical Action Data Set.* UCI Machine Learning Repository. 2007; Available at: https://doi.org/10.24432/C5MG7Z.

[17] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, *Human Activity Recognition Using Smartphones,* UCI Machine Learning Repository, 2013; dataset available at https://doi.org/10.24432/C54S4K.

[18] N. Amin Choudhury, S. Moulik, and D. Sinha Roy, *HARSense: Statistical Human Activity Recognition Dataset*, IEEE Dataport, 2021; dataset available at https://dx.doi.org/10.21227/9pt3-2m34.

[19] G. Keller, Vicon Motion Systems. 2024. Available at: https://www.vicon.com/.

[20] S. Dalwinder and S. Birmohan, "*Investigating the impact of data normalization on classification performance"*, Appl Soft Comput 97 (2020), https://doi.org/10.1016/j.asoc.2019.105524.

[21] S. Dalwinder, S. Birmohan, "*Feature wise normalization: An effective way of normalizing data*," Pattern Recognit., 122(2022). https://doi.org/10.1016/j.patcog.2021.108307.

[22] C. Damien, S. Y. Nadia, C. Didier, and P. Marie-Cécile, "*Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering"*, VPPC. IEEE., 2021, DOI: 10.1109/VPPC53923.2021.9699234.

[23] Y. C. Jong, "*Artificial Neural Networks and Backpropagation* in *Geometry of Deep Learning, Mathematics in Industry"*, Springer, 2022, https://doi.org/10.1007/978-981-16-6046-7_6.

[24] P. Harshini and V. K. Nagaraja, "*Artificial neural network and math behind it* in *Smart Trends in Computing and Communications"*, SmartCom, Springer, 2022, Singapore, pp. 205-221, https://doi.org/10.1007/978-981-16-9967-2_21.

[25] J. V. Ameet, "*Perceptron and Neural Network"s* in Machine Learning and Artificial Intelligence. Springer, 2022, pp. 57–72, https://doi.org/10.1007/978-3-031-12282-8_6.