Soft Comput (2017) 21:5805–5813
DOI 10.1007/s00500-016-2158-2

METHODOLOGIES AND APPLICATION

BAT-ANN based earthquake prediction for Pakistan region

Sehrish Saba[1] · **Faraz Ahsan**[2] · **Sajjad Mohsin**[1]

Published online: 10 May 2016

Abstract Earthquakes are natural disasters which may result in heavy losses. Accurate prediction of the time and intensity of future earthquakes can lead to minimizing losses due to earthquakes. A number of earthquake predictions have been proposed based on mathematical and statistical models. In this paper, we present an earthquake prediction technique using Bat Algorithm (BA) and Feed Forward Neural Network (FFNN). The BA is used to train the weights of the FFNN to predict future earthquakes on the basis of past input data. Experimental results show that our proposed approach is highly comparable and more stable than Back Propagation Neural Network (BPNN) with respect to accuracy.

Keywords Earthquake prediction · Bat Algorithm · Artificial neural network · Optimization technique

1 Introduction

Efficient prediction of earthquake is necessary to take precautionary measures and to avoid destruction, especially human causalities (Preethi and Santhi 2011). Though, earthquakes can hit any area, any time the mountain regions are more vulnerable to earthquakes and are also the most affected. Pakistan is located in such geographical region that three out of seven seismic plates lie underneath and seven of the top ten mountain peaks are situated in this region. Hence, the likelihood of an earthquake here and then increases. When observed closely, Pakistan has three areas where earthquakes of high magnitudes occur quite frequently. They are, Azad Kashmir, Hindukush range and Baluchistan. Unfortunately, this part of the world has not been inspected much for earthquake prediction as needed, in spite of the fact that such quakes resulted in mass destruction and loss of lives during the last 100 years or so.

An accurate prediction of a future earthquake is an extremely useful way of avoiding mass destruction. Earthquake prediction is an open area of research. A number of mathematical and statistical models have been proposed to predict upcoming earthquakes. Earthquake prediction techniques can be classified into three categories, i.e., mathematical/statistical model-based techniques, AI-based techniques and hybrid approaches. A reasonable number of earthquake prediction techniques featuring artificial intelligence (AI) also exist. One of these is the Artificial Neural Network (ANN) that has showed strong ability for finding out solution of problems in various domains. Different algorithmic variations have been proposed to improve the accuracy of ANN. The most commonly used algorithm is Back Propagation but has slow convergence rate and generally has the tendency to get stuck in local minima. However, Bat Algorithm when compared with other algorithms has shown better results on E-learning dataset (Khan and Sahai 2012). Genetic Algorithm, Levenberg Marquardt and Particle Swarm Optimization also have slow convergence and are found less efficient than Bat Algorithm. Bat Algorithm is an optimization algorithm and has been inspired by the way the natural bats hunt insects. In this study, we propose a hybrid approach for earthquake prediction by using Bat Algorithm to train ANN for Pakistani regions. To the best of our knowledge Bat Algorithm has not been tested for earthquake, which on the basis of literature is a potential candidate to minimize

Communicated by V. Loia.

✉ Sehrish Saba
sehrishsaba_1989@yahoo.com

1 Comsats Institute of Information Technology, Islamabad, Pakistan

2 HITEC University, Taxila, Pakistan

errors. Additionally, the Pakistani region which as stated above is quite at risk in terms of earthquakes, but has not been examined much for predicting seismic activity based on past events.

Rest of the paper is organized as follows: Sect. 2 presents literature review of earthquake prediction techniques applied and analyzes the previously proposed techniques, i.e., Neural Network, Swarm intelligence and Fuzzy logic. Section 3 describes the proposed methodology for earthquake prediction. Section 4 explains the results of proposed and comparison technique. Finally, we conclude and highlight future directions.

2 Related work

Different structures and techniques have been merged to improve earthquake prediction in recent past. In 2001 Bodri et al. considered four datasets of Carpathian-Pannonian, Hungary, Peloponnesos-Aegean Area and Greece and sampled two datasets. Aegean region provides spatial homogenous seismicity rates with shallow depth. Carpathian region offers heterogeneity thus only data with magnitude ≥ 4 has been used. Earthquake is predicted in Carpathian-Pannonian and Aegean region using three-layered Feed Forward Neural Network (FFNN). Mean seismicity rates were calculated using Least Square Fitting (Bodri 2001). Liu et al. contributed by introducing Adaptive Resonance Neural Network with impulsive force for rules extraction. Impulsive force was provided by Parallel Genetic Algorithm that gives more efficient results compared to simple Genetic Algorithm (Liu et al. 2004).

To avoid shortcomings of Back Propagation (BP) algorithm Zhang et al. added a globally convergent Genetic Algorithm. Neural Network weights and thresholds were arranged by BP algorithm. Initial population was generated using GA, whereas final training was carried out with BP; and was found better in efficiency and accuracy as compared to normal BP (Zhang and Wang 2008). Southwest Yunnan Province was considered by Ying et al. where the data were trained using Radial Basis Function Neural Network to predict the earthquake magnitude. Radial Basis Function is considered to provide faster prediction and better results than Back Propagation (Ying et al. 2009). Based on Tehran Geophysics Research Center, Suratgar et al. (2008) predicted earthquake magnitude. The parameters taken into account included geomagnetic field declination, horizontal component and hourly relative humidity, rain rate per day, and ground temperature. Non-linear autoregressive network with exogenous inputs (NARX) neural network is used. Consequently, they were able to predict a quake 2 days in advance.

Su et al. took varying factors that cause an earthquake to be disastrous. An ANN-based model was developed and later Tashkent City earthquake was projected by multilayer perceptron with two hidden layers (SU and ZHU 2009). Adeli et al. utilized Probabilistic Neural Network (PNN) based on the Bayesian Classifier and Parzen Window Classifier. Accuracies were calculated using statistical measures. Thus, authors claim that PNN model can predict a quake having magnitude less than 6.0, whereas recurrent neural network model greater than 6.0 (Adeli and Panakkat 2009). Dehbozorgi et al. introduced a combination of ANN and fuzzy logic. To remove noise high-pass filter has been applied on signals. For testing features, signals were converted into vectors and inserted into neuro-fuzzy classifier. Feature selection is performed by UTA algorithm, after training and testing is applied (Dehbozorgi and Farokhi 2010). Shah et al. utilized dataset of Southern California Earthquake Data Center (SCEDC). Artificial Bee Colony (ABC) algorithm has higher global optimization, so it has been used to overcome the local optimum convergence of BP. Neural Network structure where 3-3-1 was found best as per the dataset. The weights of MLP were initialized, evaluated, and fitted using ABC Algorithm. Normalized and typical mean square errors (MSE) were calculated and hence the authors concluded that MLP-ABC showed higher accuracy (Shah et al. 2011).

Alavi et al. used Pacific Earthquake Engineering Research Center (PEER) database for image formation. Artificial Neural Network with Simulated Annealing (SA) is introduced. The Principal Ground Motion parameters devised are as: Peak Ground Acceleration (PGA), Peak Ground Velocity (PGV) and Peak Ground Displacement (PGD). Artificial Neural Network and Multiple Linear Regression were compared with the proposed technique. Accurate Ground Motion parameters were acquired by ANN/SA attenuation models (Alavi and Gandomi 2011). Authors have discussed two case studies using ANN in the region of Greece. In first case study, time series values are used as input, whereas in second seismic electronic signals are taken as input. Magnitude being predicted is the output of the system (Moustra et al. 2011). Zamani designed ANN for structural hazard in Saudi Arabia (Zamani et al. 2012). Equation of motion was obtained through mass of single storey structure, stiffness of the structure, damping and displacement relative to the ground, etc. Ground acceleration was taken as input and response as output. In Alarifi (2012) considered data from Northern California Earthquake Data Center (NCEDC) for Red Sea Region. Two accuracy metrics used are MSE and mean absolute error (MAE). ANN is compared with other forecasting methods used such as Moving Average, Normal Distributed Random Predictor, and Uniformly Distributed Random Predictor. Thus ANN has shown a good trend in capturing non-linear relationship.

Different swarm intelligence learning algorithms are combined with ANN to enhance performance. Simple Particle Swarm Optimization (SPSO) is used to solve inverse problems (Deep et al. 2012). Quadratic PSO (QPSO) has also been introduced with factor CH = 30. Himalayan region data are used to compare both algorithms and QPSO shows more accuracy than SPSO (Deep et al. 2012). Deep et al. has implemented LXPSO algorithm to solve inverse problem which has Laplacian particle that is replaced by worst particle of swarm. Data of North West Himalayan region (2008) earthquake are taken. LXPSO has been compared with PSO and thus proved proposed algorithm has high accuracy (Deep et al. 2011). Prakash represented a modified ABC Algorithm to enhance the optimization performance for Hindu Kush and North Himalayan region. Greedy approach is introduced in out-looker bees. Thus Improved Artificial Bee Colony (IABC) algorithm gave more accuracy for hypocenter (Prakash 2012). Akhoondzadeh proposed the use of PSO for training ANN for Iran dataset for earthquake of 7.7 Richter scale (Akhoondzadeh 2014).

Various techniques have been developed for earthquake prediction. However, improvements can be made to get more accurate prediction results. Neural network demonstrates a high level of adaptability due to non-linear complex mapping between input and output, while Bat Algorithm (Yang 2010) is an optimization algorithm which provides diversity of solution and passes through evolution while attempting to gain a global best. In this study, we propose a hybrid approach for earthquake predicting using Bat Algorithm and Artificial Neural Network. The diversity of Bat Algorithm is combined with adaptability and efficient modeling of Neural Network to gain more accurate prediction results than many other proposed techniques (Dehbozorgi and Farokhi 2010; Alarifi et al. 2012). We have experimentally shown that proposed BAT-ANN is highly comparable with respect to BPNN with respect to prediction accuracy.

3 Proposed methodology

Evolutionary algorithms like Bat Algorithm provide variety of solutions that can help avoid overfitting in ANN. The Back Propagation in BPN works on the principle of gradient descent and may face overfitting as well as it can get stuck in a local optimum. Bat Algorithm on the other hand, does not guarantee the most optimum solution, yet it evolves to find optimum or near-optimum solution. By combining both the approaches, we can get diversity of Bat Algorithm and accuracy of ANN. In our proposed method, Feed Forward Neural Network (FFNN) is being trained by the Bat Algorithm. Flow diagram of purposed method is shown in Fig. 1.

The population of the algorithm consists of NP bat individuals. Each individual bat in the population (S) represents the set of weights of an entire FFNN. The proposed BAT-ANN algorithm consists of FFNN with an input layer, a hidden layer and an output layer. Number of input units in the FFNN is set to the *number of input records* times the number of attributes of the data set to be used for prediction. The number of output units is set to the number of attributes of the input dataset, because we predict the future record based on the data of the past input records. Number of hidden units is arbitrary. The FFNN works as follows:

Activation y_in at hidden unit j is calculated as in Eq. 1:

$$y\_in_j = b_j + \sum_{xi1}^{m} x_i \cdot v_{ij} \tag{1}$$

where b represents the bias weight, m represents the number of input units, x represents the input vector whose values are taken from the data set and v is the weight matrix between the input layer and the hidden layer.

The value y of the hidden unit j is calculated as in Eq. 2:

$$y_j = f(y\_in_j) \tag{2}$$

where $f(y\_in_j)$ is the value obtained by applying activation function at the unit j in the hidden layer.

Activation z_in at hidden unit j is calculated as in Eq. 3:

$$z\_in_j = b_k + \sum_{j=1}^{n} y_j.w_{jk} \tag{3}$$

where b represents the bias weight, n represents the number of hidden units, y represents the vector of hidden layer values and w is the weight matrix between the hidden layer and the output layer.

The value z of the output unit k is calculated as in Eq. 4:

$$z_k = f(z\_in_j) \tag{4}$$

where $f(z\_in_j)$ is the value obtained by applying activation function at the unit k in the output layer.

Squared error at the output unit k is calculated as in Eq. 5:

$$E_k = \frac{1}{2}(t_k - y_k)^2 \tag{5}$$

where t is the vector of the target values of the data (record) to predict.

Similarly, the absolute error is calculated as in Eq. 6:

$$E_k = \frac{1}{2}|t_k - y_k| \tag{6}$$

Mean squared error (MSE) or mean absolute error (MAE) of the FFNN is calculated as in Eq. 7:

$$\epsilon = \frac{1}{P}\left(\sum_{k=1}^{n} E_k\right) \tag{7}$$

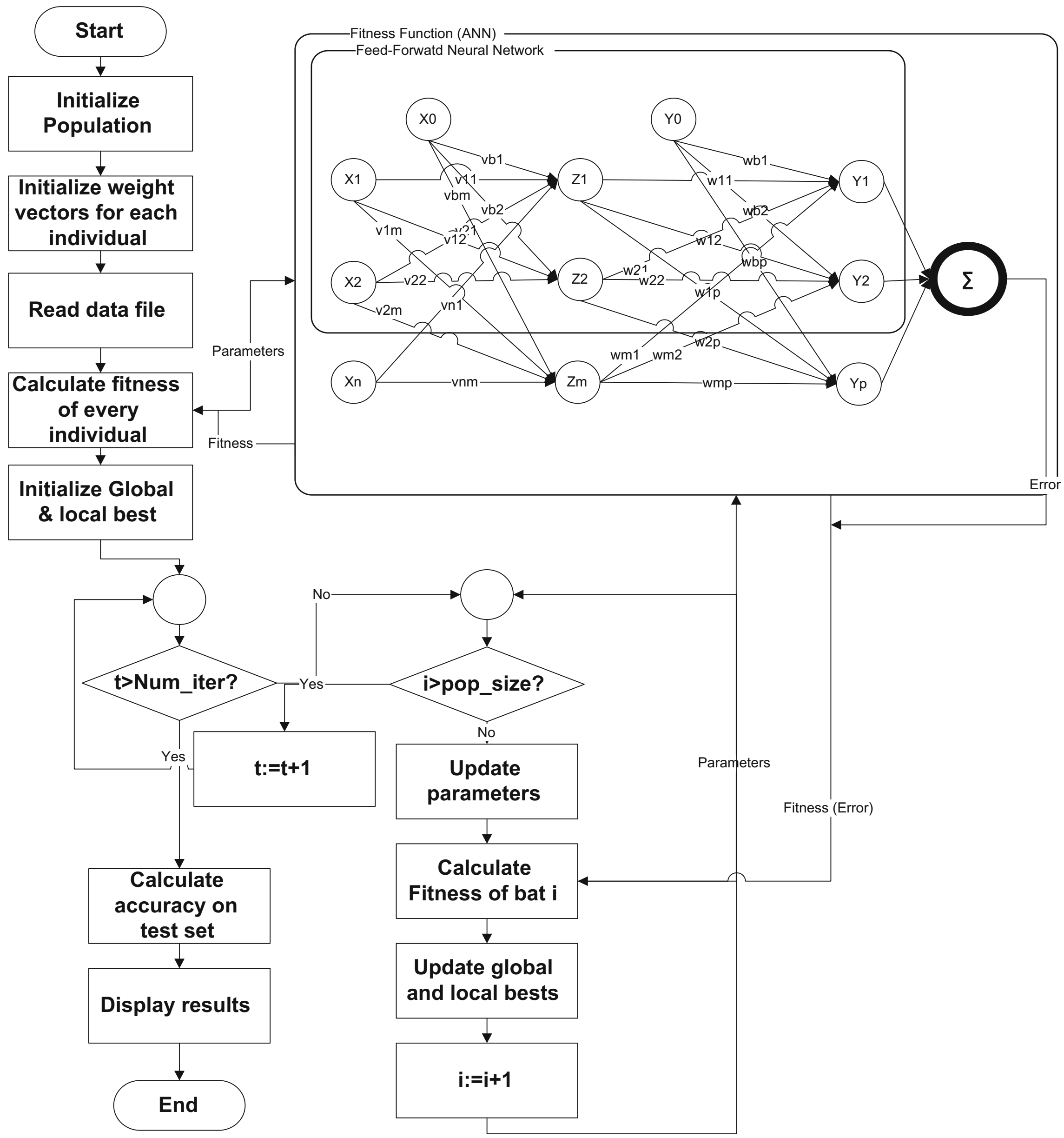

Fig. 1 Flow diagram of Bat-ANN

where P is the number of input data patterns (past records used for prediction of the future data), while ϵ is the MSE/MAE of the FFNN. This value is used as the fitness of each bat. The lower the error means the better is the fitness of the bat individual. FFNN is the function of the fitness of each bat individual as illustrated in Eq. 8:

$$\text{Fitness} = \text{ANN}(\text{WeightVector}, x, h, \text{data}), \qquad (8)$$

where *WeightVector* is the weight vector represented by each bat individual, x represents the number of input units, h represents number of hidden units while *data* contains the instances of the data set to be used for prediction (P input records and 1 output record).

The weight vectors (v and w) as well as the bias weights (b) are the dimensions of the bat. If there are a number of attributes in the data set and we want to use P number of input records (instances of data set) to predict temporarily next future record (instance) while there are h number of units in hidden layer and a number of output layer, then each bat individual would consist of the number of weights according to the following Eq. 9:

$$\text{No.of dimensions of bat individual} \\ = (P \times a + 1) \times h + (h + 1) \times a \qquad (9)$$

The above equation calculates the size of each bat individual stored in a vector of real values initialized with small random values. The expression $P \times a$ specifies the number of input layer units. The value of h is arbitrary.

If we want to predict 6th instance on the values of 5 instances where each instance has 3 attributes, the input units of the FFNN will be 15. If we set the number of hidden units (neurons) equal to number of units in the input layer then $h$will also be 15. As we want to predict all attribute values of one instance, the number of output layer units will be 3. There will be two bias weight vectors, i.e., a vector for hidden layer (consisting of 15 weight values) and the other for output layer (consisting of 3 weight values).

Hence according to the above equation, total number of weights required by the FFNN is 288. As each bat individual contains set of weights of entire neural network, hence a bat individual consists of 288 weights in this case.

Pseudo-code for our proposed BAT-ANN algorithm is given below:

Algorithm 1. Proposed BAT ANN

1. Initialize Bat Population (S) of size (N)
2. Randomly initialize weights for each individual in *S*
3. Read data file
4. Calculate fitness of each individual in *S*
5. For *i*=1 to No_of_iterations do
   - 5.1. For j=1 to n
   - 5.2. Update velocities and other parameters for S[j]
   - 5.3. Calculate fitness of individual S[j]
   - 5.4. Update local best stochastically
   - 5.5. Update global best
   - 5.6. Update loudness and pulse rate parameters
6. Calculate fitness on test set
7. Display results

Bat population is initialized in step 1 with random small values. Weights are initialized in step 2. Data are loaded from input data set in step 3 and formatted as sliding dataset to give a ‘sliding’ sense where past data instances are gradually dropped from input as new inputs are inserted in the input.

After arranging the data to be used for prediction with sliding effect, data are split into training set and test set in step 4. Initial fitness of each individual is calculated in step 5. Evolution of the population is performed in step 6. Then calculate fitness of test set in step 7 and finally display results in step 8.

Bat Algorithm is an optimization algorithm and position of bat individuals depends on a number of parameters.

Frequency of each bat individual is calculated as in Eq. 10:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{10}$$

where f_i represents frequency of the bat individual I, $f_{\min}$ and $f_{\max}$ define the user-defined range of frequencies and $\beta \in [0, 1]$ is a random number. Hence frequency of each bat individual is randomly initialized.

Velocity v and initial position x are randomly initialized for each bat individual i.

Temporary velocity of bat individual I is updated as in Eq. 11:

$$\text{temp\_}v_i^{t+1} = v_i^t + (x_i^t - x_*^t) f_i \tag{11}$$

where t represents the iteration number of the algorithm, x_i^t represents the position and x_*^t represents the global best position attained till iteration t. Equation 12 updates the current position.

$$\text{temp\_}x_i^{t+1} = x_i^t + \text{temp\_}v_i^t \tag{12}$$

$$\text{temp2\_}x_i^{t+1} = \begin{cases} \text{temp\_}x_*^{t+1} + \mu \times \text{rand2}, & \text{rand1} > r_j \\ \text{temp}_{x_i^{t+1}} & \text{otherwise} \end{cases} \tag{13}$$

In Eq. 13 variables temp_x and temp2_x represent the temporary or intermediate positions of the individual bat i. Variable μ is a small number usually set to 0.001. It is multiplied by some random number (rand2) and added to the updated position to increment the updated position if some other random number (rand1) is greater than the pulse rate parameter r, which is user-defined parameter and has a small value (usually 0.01).

After the temporary position updates, fitness of the entire population for iteration $t$is calculated.

The final position of the individual bat $i$for iteration $t+1$ is calculated as in Eq. 14:

$$x_i^{t+1} = \begin{cases} \text{temp2\_}x_i^{t+1}, & \text{Fitness}_i^{t+1} > \text{Fitness}_*^t \wedge \text{rand} < A_i \\ \text{temp\_}x_i^{t+1}, & \text{otherwise} \end{cases} \tag{14}$$

where Fitness_i^{t+1} represents the fitness of the individual bat i in iteration $t+1$, Fitness_*^t is the best fitness of the previous iteration t and Arepresents the loudness value of the individual bat i.

The update condition used for position update in the Eq. 15 is also used to update the values of loudness factor A.

$$A_i^{t+1} = \begin{cases} \alpha \times A_i^t, & \text{Fitness}_i^{t+1} > \text{Fitness}_*^t \wedge \text{rand} < A_i \\ A_i^t, & \text{otherwise} \end{cases} \tag{15}$$

where $\alpha$is a constant parameter which plays the similar role as a cooling factor does in a cooling schedule in Simulated Annealing. The pulse rate parameter $r$is updated as in Eq. 16.

Table 1 A sample of earthquake data

Date	Time	Lat	Lon	Depth	Mag	Magt	Nst Gap	Clo RMS	SRC	Event ID
1/9/2007	01:55:01.53	34.7330	73.2660	61.00	3.70	Mb	23	0.56	NEI	200701094005
1/20/2007	19:10:09.59	33.2710	73.7070	60.80	3.60	Mb	14	0.84	NEI	200701204068
3/4/2007	17:44:19.37	34.7750	73.2280	30.30	3.90	Mb	31	1.31	NEI	200703044078
3/9/2007	21:46:36.60	34.7370	73.1960	10.00	4.00	Mb	20	1.26	NEI	200703094049
4/19/2007	17:22:06.76	34.0140	73.9220	46.40	3.80	Mb	15	1.13	NEI	200704194071
6/8/2007	15:29:27.59	34.2490	73.6840	15.00	4.60	Mb	118	0.89	NEI	200706084044
6/22/2007	19:12:43.31	34.3010	74.3750	48.90	3.50	Mb	30	1.12	NEI	200706224064
7/9/2007	22:30:22.92	34.8380	73.1420	66.80	3.50	Mb	24	0.88	NEI	200707094088
7/21/2007	04:53:22.80	34.8260	73.2190	57.00	4.40	Mb	23	1.00	NEI	200707214018
7/24/2007	21:02:48.65	34.6740	73.1840	72.50	3.40	Mb	20	0.77	NEI	200707244066
8/1/2007	04:53:38.70	34.0750	74.6120	47.60	3.40	Mb	14	0.93	NEI	200708014022

$$r_i^{t+1} = \begin{cases} r_i^0 \times [1 - \exp(-\gamma \times r_i^t)], & \text{Fitness}_i^{t+1} > \text{Fitness}_*^t \wedge \text{rand} < A_i \\ r_i^t, & \text{otherwise} \end{cases} \tag{16}$$

where γ constant parameter is used to control the rate of change in pulse rate r.

Aftershocks and pre-shocks are not main earthquake. Its main identification is through its magnitude and specific time of occurrence after and before larger earthquake, respectively. Thus dataset is preprocessed to cater the specific earthquake data. Bat Algorithm provides best weight to neural network as it is termed best for NN training, as discussed earlier.

After the completion of the training on training set, the weight vector of the global best bat individual are used on the test set to calculate accuracy.

4 Experiments and results

In order to study the performance of proposed algorithm, some standard data have been chosen from United States Geological Survey(USGS) US geological survey (2013). The algorithm has been applied on the earthquake data for three areas of Pakistan:

- Azad Kashmir
- Baluchistan
- Hindukush

The time duration for data collection has been from Jan 2002 to Dec 2012. Data are divided into training and testing of 75:25, respectively. Here is a sample of dataset shown in Table 1.

Pre-processing is necessary to make data more feasible. Only three attributes have been chosen from the selected data.

Table 2 Training and testing division

	Azad Kashmir	Baluchistan	Hindukush
Actual no. of instances	336	276	413
No. of instances	166	136	204
Training set size	125	102	153
Test set size	41	34	51

- Time difference between any two occurrences between earthquakes (Delta T) in terms of days.
- Depth of the source of earthquake.
- Magnitude (3.5–7.5).

The readings that belong to different aftershocks have been eliminated and are out of the scope of this study. The proposed algorithm can be used to predict earthquakes of magnitude between 3.5 and 7.5. This is because the earthquakes less than 3.5 are usually unnoticed by the human beings and happen more frequently, hence they can create noise in the data. On the other hand, earthquakes of magnitude greater than 7.5 are relatively few and far between, hence they are harder to predict. Overlapping datasets were generated to provide moving effect in the predictions. For every instance around half instances were that of previous input and remaining were the newer occurrences. The data were divided into training and testing set having a ratio of 75:25. Azad Kashmir has 336 numbers of instances, whereas Baluchistan has 276 and Hindu Kush has 413, each. Table 2 shows division of training and testing dataset.

In view of experimental specification, Bat-ANN is compared with Back Propagation Neural Network. The computing language MATLAB was used to simulate the results. Parameter values of Bat-ANN that are commonly being used in literature are considered and listed in Table 3 (Yang 2010).

Table 3 BAT-ANN parameters

Loudness rate update factor α	0.9
Initial loudness (A)	1
Initial pulse rate (r)	0.01
Pulse rate update factor ($\bar{Y}$)	0.9
Population size	40
No. of iterations	100

The parameters used for feed forward neural network approach are: the transfer function used in BPNN is log-sigmoid keeping learning rate at 0.3 while the convergence rate is 0.001 and 100 iterations were run. For proposed algorithm parameters taken into account are: population size is 40, initial loudness is taken as 1, pulse rate (r) is initialized at 0.01, loudness rate update factor (α) is 0.9, whereas pulse rate update factor ($\bar{Y}$) is 0.9. Numbers of iterations are 100 keeping step-size of 2. Input units are 15 and so are hidden neurons while the output units are taken as 3. For the resulted data, standard deviation is calculated to find out correct numbers of magnitude and delta (λ). Accuracy is measured through Eq. 17:

$$\text{Accuracy } (\%) = \frac{DM + D + M}{DM + D + M + I} \times 100 \tag{17}$$

where *DM* represents the count of predictions which have been correct both with respect to time and magnitude. *D* represents the count of predictions which have been correct with respect to time but incorrect with respect to magnitude. *M* represents the count of predictions which have been incorrect with respect to time but is correct with respect to magnitude, only. *I* represents the count of predictions which have been incorrect with respect to both the magnitude and time. The "correctness" of a test instance is defined as:

- A test instance is considered to be correctly predicted with respect to both the date and magnitude if the difference between its predicted values (date and magnitude) and its actual values (date and magnitude) is less than the standard deviations of the predicted values (date and magnitude) of all the test instances.
- A test instance is considered to be correctly predicted with respect to date if the difference between its predicted date and its actual date is less than the standard deviations of the predicted date of all the test instances. But magnitude prediction violates this condition.
- A test instance is considered to be correctly predicted with respect to magnitude if the difference between its predicted magnitude and its actual magnitude is less than the standard deviations of the predicted magnitude of all the test instances. But date prediction violates such condition.
- All other test instances are incorrect.

Table 4 Sample results

		Azad Kashmir	Baluchistan	Hindukush
Training set size		125	102	153
Test set size		41	34	51
DM count	BAT-ANN	9	17	9
	MLP (BPN)	8	12	9
D count	BAT-ANN	25	5	18
	MLP (BPN)	20	10	16
M count	BAT-ANN	1	3	10
	MLP (BPN)	3	0	11
Incorrect count	BAT-ANN	6	9	14
	MLP (BPN)	10	12	15

Definition of correctness of an instance is given below:

- If the difference between predicted magnitude (m) of an earthquake and actual magnitude of the next earthquake is less than the standard deviation of magnitude of the entire test set, the prediction of the next earthquake is said to be correct with respect to magnitude.
- If the difference between predicted number of upcoming days (d) within which the next earthquake would occur and the actual number of days is less than the standard deviation of delta-T (in days) of the entire test set, the prediction of the next earthquake with respect to time (days) is said to be correct.

Table 4 shows the results of a sample run. *DM* count shows correct prediction of time and magnitude that Bat-ANN and BPN have predicted. It shows highest value in case of Baluchistan. *D* count shows correct prediction of time which is observed good for Azad Kashmir region. *M* count shows correct prediction of magnitude which is found better for Hindukush. As obvious from the results that Bat-ANN has shown significantly higher accuracy on Kashmir data set throughout than the other two datasets. This phenomenon occurs because Kashmir dataset contains relatively uniform distribution of events with respect to time difference and magnitude of earthquakes.

Table 5 shows the calculated mean square error (MSE) and mean absolute error (MAE). ANN when trained with Bat Algorithm comprehensively surpasses BPN in MSE minimization for Azad Kashmir dataset. Similar is the trend in minimizing MSE for Hindukush dataset. Other MSE and

Table 5 MSE and MAE

Dataset	MSE		MAE	
	BAT-ANN	MLP (BPN)	BAT-ANN	MLP (BPN)
Azad Kashmir	0.0091	0.0108	0.0488	0.0560
Baluchistan	0.0155	0.0150	0.2097	0.2136
Hindukush	0.0274	0.0320	0.1974	0.2069

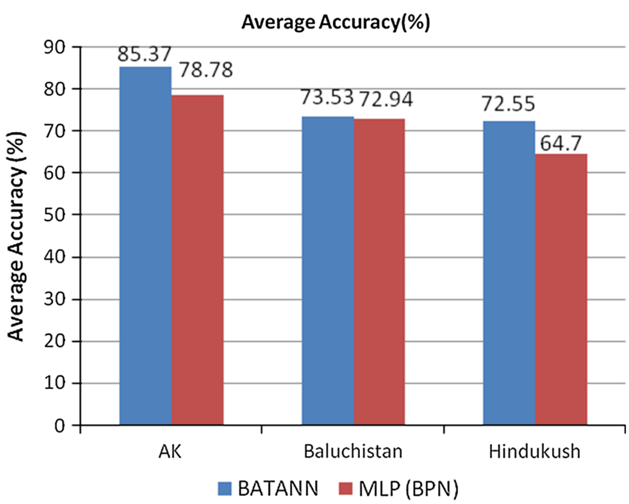

Fig. 2 Average accuracy (%)

MAE readings were observed pretty much close for both algorithms in terms of MSE and MAE minimization.

Results show that Bat-ANN has outclassed BPN algorithm with respect to prediction accuracy in Hindukush and Azad Kashmir data sets. Azad Kashmir data set had a lot of abrupt changes in instances following previous ones.

For Azad Kashmir dataset, Bat-ANN showed an average accuracy of 85 % with a standard deviation of just 3.5 %, whereas BPN showed an average accuracy of 78 % with a standard deviation of 5.05 %. This gives an evidence of more stability of Bat-ANN along with better prediction accuracy than BPN. Bat-ANN also beats BPN both in stability and prediction accuracy on Hindukush Dataset, where Bat-ANN showed an average accuracy of 72.55 % with a standard deviation of 2.96 %. BPN on the other hand showed an average prediction accuracy of 64.7 % with standard deviation of 5.18 %. The average accuracy (%) of the considered dataset with proposed methodology when compared with existing solution is shown in Fig. 2.

Bat-ANN showed a highly comparable average accuracy to that of BPN on Baluchistan dataset, which was found out to be 73.53 % with standard deviation of 2.08 %, while BPN showed an average accuracy of 72.94 % with standard deviation of 4.93 %. In the larger picture, results of BPN deviated above 5 % while that of Bat-ANN was steadier being less than 3 %. On average, Bat-ANN provided 5 % more better accuracy which extended to approximately 13 % for best case, with respect to the considered datasets.

5 Conclusion and future directions

Prediction of future earthquakes on the basis of historical information is a challenging task. Though, literature suggests that quite many schemes have been proposed but are either location or data dependent. Additionally, the generalized algorithms lack performance issues and hence there exist room for optimization. In this paper, we have used Bat-ANN as a tool for predicting future earthquakes. Instead of using back propagation of error in ANN, we have used Bat Algorithm to update neural network weights. The results show that Bat-ANN showed more accuracy and stable predictability of earthquake than ANN with respect to time and magnitude of the next earthquake. Moreover, since Bat-ANN uses the guided random (pseudo-random) to global best, thus provides more diversity than BPN which uses rigid approximation rules to adjust optimal weights that have higher probability to end getting in a sub-optimal solution. Due to its diversity and stochastic approach, Bat-ANN has more chances to find global optima. The proposed methodology can be applied in various fields where prediction is of utmost importance such as landslides prediction. Bat Algorithm can be optimized by tuning other parameters thus it can be beneficial to improve results. In future, other seismic data can be tested and compared with this study to develop a generalized functioning framework, initially for mountain region that can later be extended to other areas, too.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Statement of human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

Adeli H, Panakkat A (2009) A probabilistic neural network for earthquake magnitude prediction. J Neural Netw 22(7)

Akhoondzadeh M (2014) Thermal and TEC anomalies detection using an intelligent hybrid system around the time of the Saravan, Iran, (Mw = 7.7) earthquake of 16 April 2013. Adv Space Res 53:647–655

Alarifi ASN, Alarifi NSN, Al-Humidan S (2012) Earthquakes magnitude predication using artificial neural network in northern Red Sea area. J King Saud Univ Sci 24:301–313

Alavi AH, Gandomi AH (2011) Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing. Comp Struct 89:2176–2194

Bodri B (2001) A neural-network model for earthquake occurrence. J Geodyn 32:289–310

Deep K, Yadav A, Kumar S (2012) Improving local and regional earthquake locations using an advance inversion technique: particle swarm optimization. World J Model Simul 8(2):135–141

Deep K, Yadav A, Kumar S (2011) Determining earthquake locations in NW Himalayan region: an application of particle swarm optimization. Int J Comput Sci Math 3(2):173–181 (ISSN 0974–3189)

Dehbozorgi L, Farokhi F (2010) Effective feature selection for short-term earthquake prediction using neuro-fuzzy classifier. In: 2010 Second IITA International Conference on Geoscience and Remote Sensing

Khan K, Sahai A (2012) A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. IJISA 4(7):23–29

Liu Y, Liu H, Zhang B, Wu G (2004) Extraction of if-then rules from trained neural network and its application to earthquake prediction. In: Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04)

Moustra M, Avraamides M, Christodoulou C (2011) Artificial neural networks for earthquake prediction using time series magnitude data or Seismic Electric Signals. Exp Syst Appl 38:15032–15039

Prakash D (2012) Bespoke artificial Bee Colony Algorithm to determine the earthquake locations. Adv Mech Eng its Appl (AMEA) 2(3):207 (ISSN 2167–6380)

Preethi G, Santhi B (2011) Study on techniques of earthquake prediction. Int J Comp Appl 29(4) (0975–8887)

Shah H, Ghazali R, Nawi NM (2011) Using artificial bee colony algorithm for MLP training on earthquake time series data prediction. arXiv:1112.4628

SU YP, ZHU QJ (2009) Application of ANN to prediction of earthquake influence. In: Second International Conference on Information and Computing Science

Suratgar AA, Setoudeh F, Salemi AH (2008) Magnitude of earthquake prediction using neural network. In: Fourth International Conference on Natural Computation IEEE. doi:10.1109/ICNC.2008.781

US geological survey hazards program website. earthquake.usgs.gov/Retrieved 2013-6-10

Yang XS (2010) A new meta-heuristic bat-inspired algorithm. In: Gonzalez JR et al. (eds) Nature Inspired Co-operative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, Springer, Berlin, vol 284, pp 65–74

Ying W, Yi C, Jinkui Z (2009) The application of RBF neural network in earthquake prediction. In: 2009 Third International Conference on Genetic and Evolutionary Computing. doi:10.1109/WGEC.2009.81

Zamani AS, Al-Arifi NS, Khan S (2012) Response prediction of earthquake motion using artificial neural networks. IJAR-CSIT

Zhang Q, Wang C (2008) Using genetic algorithm to optimize artificial neural network: a case study on earthquake prediction. In: Second International Conference on Genetic and Evolutionary Computing. doi:10.1109/WGEC.2008.96