Earthquake Prediction using Hybrid Machine Learning Techniques

Mustafa Abdul Salam[1]
Artificial Intelligence Dept., Faculty of Computers and Artificial Intelligence, Benha, Egypt

Lobna Ibrahim[2]
Scientific Computing Dept., Faculty of Computers and Artificial Intelligence, Benha, Egypt

Diaa Salama Abdelminaam[3]
Information Systems Dept., Faculty of Computers and Artificial Intelligence, Benha, Egypt
Faculty of Computer Science, Misr International University, Cairo

***Abstract*—This research proposes two earthquake prediction models using seismic indicators and hybrid machine learning techniques in the region of southern California. Seven seismic indicators were mathematically and statistically calculated depending on pervious recorded seismic events in the earthquake catalogue of that region. These indicators are namely, time taken during the occurrence of n seismic events (T), average magnitude of n events (M_mean), magnitude deficit that is the difference between the observed magnitude and expected one (ΔM), the curve slope for n events using inverse power law of Gutenberg Richter (b), mean square deviation for n events using inverse power law of Gutenberg Richter (η), the square root of the released energy during T time (DE1/2) and average time between events (µ). Two hybrid machine learning models are proposed to predict the earthquake magnitude during fifteen days. The first model is FPA-ELM, which is a hybrid of the flower pollination algorithm (FPA) and the extreme learning machine (ELM). The second is FPA-LS-SVM, which is a hybrid of FPA and the least square support vector machine (LS-SVM). These two models' performance is compared and assessed using four assessment criteria: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), and Percent Mean Relative Error (PMRE). The simulation results showed that the FPA-LS-SVM model outperformed the FPA-ELM, LS-SVM, and ELM models in terms of prediction accuracy.**

Keywords*—*Extreme learning machine; least square support vector machine; flower pollination algorithm; earthquake prediction

I. Introduction

Earthquake is movements and shaking inside the ground that produce energy in rocks. Like many natural disasters, Earthquake causes many damages, financial loss, and injuries [1]. Earthquakes happen daily in various regions around the world. The more susceptible regions to earth-shaking are japan [2], Indonesia, south California, turkey, Iran, and Taiwan [3]. People can feel the Earthquake if its magnitude is more than 2.5, but if the magnitude is less than 2.5, the Earthquake will not be felt. The magnitude of highly caused damaged earthquakes is more than 4.5 [4]. Sometimes earthquakes are responsible for huge numbers of deaths.

So, scientists work hard in this field to prevent these severe effects. The best effort is done to Alert people in time because a wrong alert will cause unnecessary losses. Certainly, people cannot stop the occurrence of earthquakes, but they can adopt protective measures and precautions to minimize the deleterious effects by predicting earthquake magnitude using machine learning techniques. There are a lot of methods that can be applied to predict earthquake magnitude using different sensors, devices, magnetic and electrical waves, or seismic indicators obtained from the processing of the historical data of earthquakes [1]. Really, there is no perfect model that results from 100%, but at least a trial is made to improve the accuracy as much as possible [3].

Artificial intelligence plays an essential role in predicting and classifying problems. A neural network is a very effective tool to solve complicated non-linear issues [1]. Technology and machine learning provide many robust mechanisms to study seismic data and indicators. Data mining and machine learning are highly successful instrument in the prediction domain, especially if massive data is required as weather forecasting, stock prediction, and so on [5]. Dataset plays an essential role in determining the purposed model's quality and performance, so we search a lot to choose the more exact dataset [4].

There are thousands of machine learning algorithms, but no one is always suitable for all issues because there are many factors that affect that, like the number of features (input indicators), number of records of the dataset, and type of problem (classification or regression), so we will apply some machine learning algorithms and compare their results with each other to determine the more suitable algorithm for this issue. The comparison will be applied here to compare our work with other researchers that applied different machine learning algorithms on the same dataset [6]. In this article, the magnitude will be predicted using historical records of south California. Depending on the seismic indicators as an input for machine learning algorithms, which are obtained from performing some statistical and mathematical equations on raw data (time and magnitude) [7]. Machine learning techniques can predict well if provided by a suitable dataset, which is divided into (70%for train and 30%for test) [8]. In this paper, ELM, LS-SVM, and FPA will be used. ELM (Extreme Learning Machine) is a straightforward algorithm because it depends on a feed-forward neural network. The data moves in one direction (forward direction) with only one hidden layer. ELM is a well-known and well-resulted algorithm in classification and regression domains. Unlike other ANN algorithms that face the overfitting problem and the long-running time problem, ELM overcomes these problems and can achieve high accuracy and high-speed results. ELM often uses a sigmoid function, which will be applied in this work.

SVM (Support Vector Machine) is a well-known deep learning tool for regression that depends on kernel methods giving high prediction results. SVM is a developed algorithm of machine learning to avoid ANN shortages. SVM gives optimal global solutions avoiding local minima problems [9].

LS-SVM (Least Square Support Vector Machine) is the edited version of the SVM algorithm. LS-SVM simplifies the SVM method, but it needs kernel parameters, which are important for regression problems. So we need an appropriate way for the optimal choice of LS-SVM parameters, So optimizing LS-SVM with FPA is done [9].

FPA (Flower Pollination Algorithm) was developed in 2012 by Yang, which uses pollination of flowers. FPA was applied for many non-linear problems giving high results [9].

ELM and LS- SVM are optimized by FPA (Flower Pollination Algorithm), enhancing the accuracy and minimizing errors. ELM and LS-SVM are considered supervised learning where they take seven seismic parameters (time is taken during the occurrence of n seismic indicators (T), average magnitude during n events (M_mean), magnitude deficit, which is the difference between the observed magnitude and expected one (∆ M), the curve slope for n events using inverse power law of Gutenberg Richter (b), mean square deviation for n events using inverse power law of Gutenberg Richter (η), the square root of the released energy during T time (DE1/2)), average (mean) time between events (μ) [10] as network inputs and produces the future magnitude as a network output to train well and be able to test after that. No algorithm can predict 100%, but some shortages of ELM algorithm will be removed by using optimization (swarm intelligence) algorithm FPA to get FPA _ELM (the result of optimizing ELM with Flower Pollination algorithm, which optimizes weights of input nodes and biases of a hidden layer of ELM). The optimization of SVM with FPA is applied to enhance the performance of LS-SVM and get the FPA-LS-SVM model that yields high results.

This work is divided into four parts; the first part is processing the raw data to find the seismic indicators. The second part is to input these indicators into machine learning algorithms to predict or output the expected future magnitude. The third part is to optimize the output using artificial intelligence techniques (swarm intelligence). The fourth part is comparing different results using different algorithms.

In the rest of this paper, the related work is showed in Section II, Applied Algorithms in Section III. Data and parameters calculations in Section IV, methodology in Section V, Results and discussion in Section VI, conclusion and future work in Section VII, and finally the references in machine learning and earthquake fields that is helpful in this work.

II. Related Work

Of Course, the earthquake catastrophe takes a large space of scientists' interest, so there are many kinds of research and scientists' efforts which are spent in this domain. Researchers worldwide do their best to predict where and when the Earthquake occurs depending on seismic indicators and other seismic electric signals using machine learning techniques and optimization algorithms.

Adeli et al. applied the probabilistic neural network on seismic indicators to predict the earthquake magnitude. This model predicts well for magnitude from 4.5 to 6 in the southern California region [11]. Moustra et al. who use the artificial neural network to predict earthquake occurrence using time series data and seismic electric signals in the Greece region, there are two case studies which finally led to when appropriate data presented to NN, it can predict accurately [12]. Hegazy et al. optimized ELM with FPA (flower pollination algorithm), which shows a better accuracy when applied to prediction task [13]. In this paper, Ma et al. proposed a genetic algorithm-based neural network giving GA_NN model is applied to six seismic indicators to find the relation between these indicators and the maximum earthquakes in china [14]. Maceda et al. here SVM is applied to earthquake problem. This paper decided that SVM is perfect for solving classification problems using a small-size training dataset [15]. Li et al. depended on the collected data from earthquake stations to be used in machine learning techniques to distinguish between Earthquake and non-earthquake [16]. Rajguru et al. optimized ANN (artificial neural network) with GA (genetic algorithm) to predict the source location and dimensions of the earthquakes giving good results although the complexity of the problem [17]. Wang et al. introduced a deep learning algorithm (LSTM) to find the relation among the earthquakes in different places that is useful in earthquake prediction [18]. Reyes et al. used an ANN model to predict earthquake magnitude in a limited interval or bounded by threshold during the following five days. Applying statistical tests and experiments showed the higher success rate of that method than other machine learning classifiers [19]. Martínez-Álvarez et al. used various seismic indicators as inputs for ANN in different seismic zones, and this showed that the seismic indicators are the best features for earthquake prediction [20]. Zhou et al. showed how efficiently the neural network could predict the earthquake magnitude using the LM_BP algorithm [21]. Morales-Esteban et al. used statistical methods and applied ANN on Earthquake in two seismic regions The Western Azores–Gibraltar fault and the Alborán Sea whether the magnitude is limited in an interval or determined by threshold [22]. Wang et al. proposed a RBF(Radial Basis Function) neural network to earthquake prediction, and the results showed that RBF is an effective tool for that non-linear problem [23]. Alarifi et al. used ANN for earthquake prediction in the Red Sea region, and the results showed the high ability of ANN forecasting than any other statistical models [24]. Asencio–Cortés et al. proposed a machine learning model to predict the earthquake magnitude after the next seven days using cloud infrastructure [25]. Tan et al. proposed support vector machines (SVM) and the results were good and added a new method for predicting earthquakes [26]. Florido et al. interested in processing earthquake data in

Chile. The data were labeled by applying to clustering algorithms. It was easy to identify Earthquakes with a magnitude larger than 4.4 [27]. Last et al. used many data mining methods and time series to predict the magnitude of the highest earthquake event in the following year depending on data from the previous records using a multi-objective info-fuzzy network algorithm [28]. Rafiei et al. was interested in giving early alarm weeks before maximum earthquake event using classification algorithm (CA) combining with mathematical optimization algorithm (OA) whose role is to find the location of the Earthquake with maximum magnitude [29]. Kerh et al. used a genetic algorithm combined with a neural network to evaluate the response of the Earthquake in Taiwan that produces high results comparing to neural network model only [30]. Mirrashid et al. used the system of adaptive neuro-fuzzy inference to predict the coming earthquakes with magnitude 5.5 or higher developed by three algorithms subtractive clustering (SC), grid partition (GP), and fuzzy C-means (FCM). The results showed that the ANFIS-FCM model gives the highest accuracy in predicting the magnitude of the Earthquake [31]. Asim et al. used many models of computational intelligence for earthquake prediction in northern Pakistan, and the feed-forward neural network showed the highest performance compared with other models [32]. Umar et al. combined the Logistic Regression (LR) with Frequency Ratio (FR) to overcome the limitations of each one individually. This model achieved high success and good earthquake prediction in Indonesia [33]. Asim et al. focused on computing the seismic indicators mathematically and then applying the tree classifiers on them to predict the earthquake magnitude in Hindu Kush, showing that the rotation forest gave a good prediction than random forest [34].

Machine learning (ML) methods such as artificial neural networks (ANNs), support vector machines (SVMs), extreme learning machine (ELM), are considered the most commonly used ML models in classification, regression. But these methods may suffer from local minima and overfitting problems due to using local optimization training algorithms such as gradient descent algorithm in ANN [35]. Swarm Intelligence algorithms such as particle swarm optimization (PSO), follower pollination algorithm (FPA), ant colony optimization (ACO), and artificial bee colony (ABC), can solve the problems or drawbacks of machine learning models such as ANN, SVM, and ELM methods [36,37]. Using swarm intelligence or meta-heuristic algorithms in optimizing and training classical machine learning models can enhance the accuracy and generalization ability of these methods [38-43].

III. Preliminaries

A. ELM (Extreme Learning Machine)

This model is a feed-forward neural network with only one hidden layer, a very rapid learning method with solutions for many problems caused by traditional neural network algorithms like overfitting, local minima, slowness, achieving better performance higher accuracy. ELM is a simple model that is performed through three steps.

- First step: ELM chooses the weights of input nodes and hidden biases randomly**.**
- Second step: ELM performs calculations to generate the output matrix of the hidden layer.
- Third step: calculations of the output weight.

ELM uses a single hidden layer with N hidden nodes and f(x) the activation function to learn distinct samples (M) $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i}$=$[\mathbf{x_{11}}, \mathbf{x_{12}} \dots . \mathbf{x_{1k}}]^{\mathbf{T}} \in \mathbf{R^k}$.

$\mathbf{t_i} = [\mathbf{t_{11}}, \mathbf{t_{12}} \dots . \mathbf{t_{1d}}]^{\mathbf{T}} \in \mathbf{R^{kd}}$ after that, the non-linear problem is turned to the linear problem:

$$H\beta=T \tag{1}$$

the hidden output matrix H is defined as follow:

$$H=\begin{bmatrix} \mathbf{f}(\mathbf{w_1}.\mathbf{x_1}+\mathbf{b_1}) & \cdots & \mathbf{f}(\mathbf{w_N}.\mathbf{x_1}+\mathbf{b_N}) \\ \vdots & \ddots & \vdots \\ \mathbf{f}(\mathbf{w_1}.\mathbf{x_M}+\mathbf{b_1}) & \cdots & \mathbf{f}(\mathbf{w_N}.\mathbf{x_M}+\mathbf{b_N}) \end{bmatrix} \tag{2}$$

Where $\mathbf{w_j} = [\mathbf{w_{j1}}, \mathbf{w_{j2}} \dots . \mathbf{w_{jk}}]^{\mathbf{T}}$ is the weight vector between input nodes and hidden ones, **b** is the bias of hidden nodes where j= 1,2 ,..N.

$\mathbf{w_j} . \mathbf{x_i}$ where i=1…M is the inner product of $\mathbf{w_j}$, $\mathbf{x_i}$. β= $[\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \boldsymbol{\beta_3} \dots . \boldsymbol{\beta_N}]^{\mathbf{T}}$ is output weights' matrix, but $\boldsymbol{\beta_j} = [\boldsymbol{\beta_{j1}}, \boldsymbol{\beta_{j2}}, \boldsymbol{\beta_{j3}} \dots . \boldsymbol{\beta_{jd}}]^{\mathbf{T}}$ where j= 1,2,3…..N is the vector of weights connecting the output neuron and a jth hidden one. $\mathbf{T}$=$[\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3} \dots . \mathbf{t_M}]^{\mathbf{T}}$ is targets' matrix. The errors between the estimated value $\mathbf{t_i}$ and the real value $\mathbf{t_i}$ equal to zero using the following formula:

$$\mathbf{t_i} = \sum_{\mathbf{j=1}}^{\mathbf{N}} \boldsymbol{\beta_j}\ \mathbf{f}(\mathbf{w_j}.\mathbf{x_i}+\mathbf{b_j}) \tag{3}$$

weights that link between the hidden layer and output one are estimated by the least square solution to the linear problem using the following formula:

$$B=\mathbf{H^{-1}}T \tag{4}$$

Where $\mathbf{H^{-1}}$ is the inverse of the H matrix using the Moore Penrose method that makes ELM perform better and faster [13].

B. LS-SVM (Least Square Support Vector Machine)

It is one of deep and supervised learning that can classify data and predict values LS-SVM is the new version of SVM that solves SVM issues. Using LS-SVM, the solution can be founded by solving some linear equations rather than a Quadratic programming problem used in SVM. Let x is the matrix of input data. Using training data, LS-SVM plays a good role in creating the function that shows the dependence of the output on the input. The formula of this function:

$$f(x)=\mathbf{W^T}\ \varphi\ (x)+ b \tag{5}$$

where W, φ (x): $\mathbf{R^p} \rightarrow \mathbf{R^n}$ are column vectors with size n*1, and b ∈ **R**. Now LS-SVM calculates that function using the same minimization problem in SVM. Showing the important difference that LS-SVM contains equality constraints, unlike inequality ones in SVM. LS-SVM depends on the least square function [9]. We can minimize the error using this optimization formula:

$$\min j(w,e,b)= \frac{\mathbf{1}}{\mathbf{2}}\mathbf{w^t}w + C\,\frac{\mathbf{1}}{\mathbf{2}}\mathbf{e^t} \tag{6}$$

$$\mathbf{y_i}=\mathbf{W^T}\ \varphi\ (\mathbf{x_i})+ b +\mathbf{e_i} \qquad (7)$$

Where e is n*1 column vector, C ∈ $\mathbf{R^+}$ is the parameter between training errors and solution size. In 2 Lagranian is formed that differs according to w,b,e, a where a is Largrangian multiplier, we have.

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & -\mathbf{Z^T} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1^T} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & -\mathbf{I} \\ \mathbf{Z} & \mathbf{1} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{b} \\ \mathbf{e} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{y} \end{bmatrix}. \qquad (8)$$

Identity matrix and Z= $[\varphi\ (\mathbf{x_1}), \varphi\ (\mathbf{x_2}), \dots \varphi\ (\mathbf{x_n})]^{\mathbf{T}}$

Using four from row 1, we get W= $\mathbf{Z^T}$and from row 3, we get Ce=a.

Note that: the kernel matrix K = Z $\mathbf{Z^T}$And the parameter λ = $\mathbf{C^{-1}}$, the conditions for optimality give the following solution

$$\begin{bmatrix} \mathbf{0} & \mathbf{1^T} \\ \mathbf{1} & \mathbf{K}+\lambda\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}. \qquad (9)$$

Types of the kernel function K:

First: linear kernel

$$K(x, \mathbf{x_i}) = \mathbf{x_i}^{\mathbf{T}}\mathbf{x}$$

Second: the polynomial kernel of d degree

$$K(x, \mathbf{x_i}) = (\mathbf{1}+\mathbf{x_i}^{\mathbf{T}}\mathbf{x}/\mathbf{c})^{\mathbf{d}}. \qquad (10)$$

Third: RBF kernel (radial basis function)

$$K(x, \mathbf{x_i}) = \mathbf{exp}(-\|\mathbf{x}-\mathbf{x_i}\|^{\mathbf{2}}/\ \sigma^{\mathbf{2}}). \qquad (11)$$

Forth: MLP kernel

$$K(x, \mathbf{x_i}) = \mathbf{tanh}(\mathbf{k}\mathbf{x_i}^{\mathbf{T}}\mathbf{x}+\theta). \qquad (12)$$

C. FPA (Flower Pollination Algorithm) is an optimization technique that uses the idea of pollination of flowers. The pollination process has two directions: biotic pollination, which occurred in 90% of flowers by the spread of pollens using various insects, and abiotic pollination, which occurred in 10% of flowers where the pollens move through water or wind. Two methods do the pollination Self-pollination: occurs when the flower's pollen moves to the same flower or different flowers at the same plant.

Cross-pollination: occurs when the pollen of one flower of a specified plant moves to another flower in a different plant. The steps of the FPA algorithm:

- The cross-pollination (biotic) is called global pollination, where the Levy fight takes place
- Self-pollination is considered as local pollination
- The constancy of the flower can be treated as the probability of reproduction is proportional to two flowers similarity
- There is a switch probability p ∈ {0,1} that determines the pollination process, even local or global.

In global pollination, the pollinators can carry pollens for long distances. This ensures creating the fittest (Best) that obeys levy distribution:

$$x_i^{t+1}= x_i^{t}+ L(Best-x_i^{t}) \qquad (13)$$

In local pollination, the pollen can be carried by other factors, and this can be represented using the formula:

$x_i^{t+1}= x_i^{t}+ U(x_j^{t}-x_k^{t})$ (14) [13],[9]

```
Algorithm 1: Flower Pollination Algorithm (FPA)
1) Initialization of:
    • Population size (n) and the flower population Xi (i = 1, 2,..., n) randomly chosen solutions.
    • switch probability p ∈ {0,1}
    • Maximum iterations' number.
    • The best solution in the initial population (Best)
    • Function (fn) applied to each flower
2) randomly initial population generated.
3) while (t < Max_iteration)
    for each flower do
        if (rand<p)
            Draw L (d-dimensional step vector) that undergoes a Levy distribution.

            Global pollination for the solution i
            $x_i^{t+1}= x_i^{t}+ L(Best-x_i^{t})$
        Else
            Draw U from uniform distribution in [0,1].

            From all the solutions, choose □, k randomly.

            Local pollination for the solution i
            $x_i^{t+1}= x_i^{t}+ U(x_j^{t}-x_k^{t})$.
        endif
        Calculate the fitness of each flower.

        Pass each flower(solution) to fn to evaluate the new solution.
    end for
        find Best (current best solution)
end while
```

IV. Data and Parameters Calculations

Seven parameters are mathematically and statistically calculated during a specific period of time [10] from the earthquake catalog. These parameters are the inputs for the network to predict the future expected magnitude as the output of the network.

1) Earthquake data: The Earthquake catalog source for south California is available to be downloaded for free using the website (www.data.scec.org.). The historical earthquake data of Southern California is between.

1st January 1950 and 31st May 1978 is divided into 693 periods. Each period consists of fifteen days.

2) Seismic parameters: The seven earthquake indicators are computed for each period of time. This part will show these indicators and their mathematical calculations. The first indicator is the time during the range of n events, which called T.

$$T = tn\text{- } t1 \qquad (15)$$

T1 is the time of the first event, and tn is the time for the period's nth event.

The second seismic parameter is the average magnitude of the last n events of the period, which is calculated as the following.

$$M\_mean=\frac{\sum Mi}{n} \qquad (16)$$

Mi is the magnitude of the ith event, and n is the number of events.

The third parameter is $DE^{1/2}$ the square root of released seismic energy is during time T. The $DE^{1/2}$ the parameter can be computed as follow:

$$DE^{1/2} = \frac{\sum \sqrt{E}i}{T} \qquad (17)$$

$\sqrt{E}i$ is the square root of the seismic energy (E) of the ith event where E can be calculated from the following formula.

$$Ei=10^{(11.8+1.5Mi)}ergs \qquad (18)$$

The fourth indicator is b_value, that is, the slope of the curve between the log of frequency of occurred earthquakes and the earthquake magnitudes given from Richter inverse power law.

$$\log_{10} N = a - bM \qquad (19)$$

N is the number of events with M magnitude a,b values are constants.

where b can be calculated as follow

$$b= \frac{(n\sum(Mi\log Ni)-\sum Mi\sum \log Ni)}{((\sum Mi)^2-n\sum Mi^2)} \qquad (20)$$

and a can be calculated from the following formula:

$$a= \frac{\sum(\log_{10} Ni+bMi)}{n} \qquad (21)$$

The fifth parameter is (η value), which is the sum of mean square deviation based on inverse power law. η value can be computed as follow.

$$\eta= \frac{\sum(\log_{10} Ni-(a-bMi))^2}{n-1} \qquad (22)$$

The sixth parameter is the ∆M value, defined as the magnitude deficit (the difference between the observed magnitude and the expected one). ∆M is computed as follow:

$$\Delta M= M_{observed} - M_{excepected} \qquad (23)$$

Where $M_{excepected}$ is calculated from the formula:

$$M_{excepected}={}^{a}/_{b} \qquad (24)$$

The last one μ in the meantime among the characteristic events which is calculated as follow:

$$\mu = \frac{\sum(t_{i\ characteristic})}{n_{characteristic}} \qquad (25)$$

[10].

Now a sample of the dataset from the period from 1st January 1950 to 30th May 1950 that represents ten periods of time will be presented. Each period consists of fifteen days as shown in Table I.

V. Proposed Models

The proposed methods depend on the study of historical earthquake data in earthquake catalogs. By processing these data, seismic indicators that are used as inputs for the network can be obtained. Then the ELM algorithm is optimized by FPA to enable us from an optimal prediction of the occurrence of the Earthquake, and also optimizing LS-SVM with FPA to enhance the accuracy of earthquake magnitude prediction. The network architecture contains seven input indicators, which represent the seismic indicators, and the output shows the predicted magnitude during fifteen days. The description of proposed FPA-ELM, and FPA-LS-SVM algorithms are shown in algorithm 2, and algorithm 3, respectively. The data is used in three manners. First, data is divided into 70% for training and 30% for testing. Then, data is divided into 80% for training and 20% for testing. At last data is divided into 90% for training and 10% for testing. The phases of the used models are shown in Fig. 1. The earthquake indicators which have been calculated from the datasets are shown in Table I.

TABLE I. Earthquake Indicators Sample

time period	**input seismic indicators**							**output**
	T	DE1/2	b	H	ΔM	M_mean	μ	target_mag
1/1/1950-15/1/1950	3655	1197086.433	0.585242	0.006859	0.437819	2.9305	182	3.37
16/1/1950-30/1/1950	3668	816522.7778	0.663648	0.005888	0.307195	2.59	135	2.16
31/1/1950-14/2/1950	3684	294005.8278	0.469148	0.013345	0.497108	2.255333	245	1.65
15/2/1950-1/3/1950	3699	1205497.389	0.569933	0.012245	0.034071	2.946316	194	3.05
2/3/1950-16/3/1950	3714	231294.1789	0.588185	0.009469	0.266256	2.6675	464	2.68
17/3/1950-31/3/1950	3730	483924.9513	0.566164	0.022646	0.52207	2.705333	248	2.84
1/4/1950-15/4/1950	3743	848726.3986	0.460094	0.002465	0.068989	2.713333	249	2.14
16/4/1950-30/4/1950	3758	437602.1004	0.684234	0.018921	0.311847	2.65625	234	2.44
1/5/1950-15/5/1950	3775	558145.6542	0.739229	0.026896	0.555382	2.882667	251	3.34
16/5/1950-30/5/1950	3789	534786.1837	0.52383	0.019356	0.545958	2.817692	291	3.68

Algorithm2: The Hybrid Earthquake prediction FPA_ELM

```
Input:  n number of flowers
        N_gen number of iterations
        d dimensions of search variables
        fn function applied to each flower
        Trn Training, Vld Validation datasets
Output:
        fbest optimal hidden weights and biases
        f(fbest) Sum square error for the NN
        over the validation set fbest
1) Initialization:
    • N,N_gen
    • switch probability $p \in \{0,1\}$
    • Function (fn) applied to each flower
2) randomly initial population generated.
3) while Stopping criteria not met, do
    for each flower, do
        if (rand<p)
            Draw L (d-dimensional step vector)
            that undergoes a Levy distribution.

            Global pollination for a solution i
            ${x_i}^{t+1}$= ${x_i}^{t}$+ L(Best-${x_i}^{t}$)
        Else
            Draw U from a uniform distribution
            In [0,1]

            from all the solutions choose $j,k$
            randomly

            Local pollination for a solution i
            ${x_i}^{t+1}$= ${x_i}^{t}$+ U(${x_j}^{t}$-${x_k}^{t}$).
        endif
        Calculate the fitness of each flower

        Pass each flower(solution) to fn to
        evaluate the new solution

            construct NN given the hidden layer
            weights and biases of flower i

            Calculate the output layer's weights using
            the MP matrix using training dataTrn and
            hidden layer weights and Biases.

            Use the validation data Vld to evaluate the
            NN model
            Compute prediction accuracy
    end for
end while
4) Pass Best to fn
5) Apply ELM.
6) Return prediction accuracy
```

Algorithm3: The hybrid FPA_SVM for Earthquake prediction

```
Input:  n number of flowers
        N_gen number of iterations
        Ub data Upper bound
        Lb  data Lower bound
        d dimensions of search variables
        fn function that applies to each flower to
        improve SVM parameters.
output: prediction accuracy
        Best global solution
1) Initialization:
    • Population size (n) and the flower population
      Xi (i = 1, 2..., n) randomly chosen solutions.
    • switch probability $p \in \{0,1\}$
    • Maximum iterations' number.
    • Data limits Lb, Ub with dimension d.
    • The best solution in the initial population
      (Best)
    • Function (fn) applied to each flower
2) randomly initial population generated.
3) while (t < Max_iteration)
    for each flower do
        if (rand<p)
            Draw L (d-dimensional step vector)
            that undergoes a Levy distribution.

            Global pollination for a solution i
            ${x_i}^{t+1}$= ${x_i}^{t}$+ L(Best-${x_i}^{t}$)
        Else
            Draw U from a uniform distribution
            in [0,1].

            From all the solutions choose $j$,
            randomly.

            Local pollination for a solution i
            ${x_i}^{t+1}$= ${x_i}^{t}$+ U(${x_j}^{t}$-${x_k}^{t}$).
        endif
        Calculate the fitness of each flower.

        Pass each flower(solution) to fn to
        evaluate the new solution

        Pass output to SVM parameters

        Apply SVM

        Compute prediction accuracy
    end for
    t=t+1
end while
4) Pass Best to fn
5) Pass output to SVM parameters.
6) Apply SVM.
7) Return prediction accuracy.
```

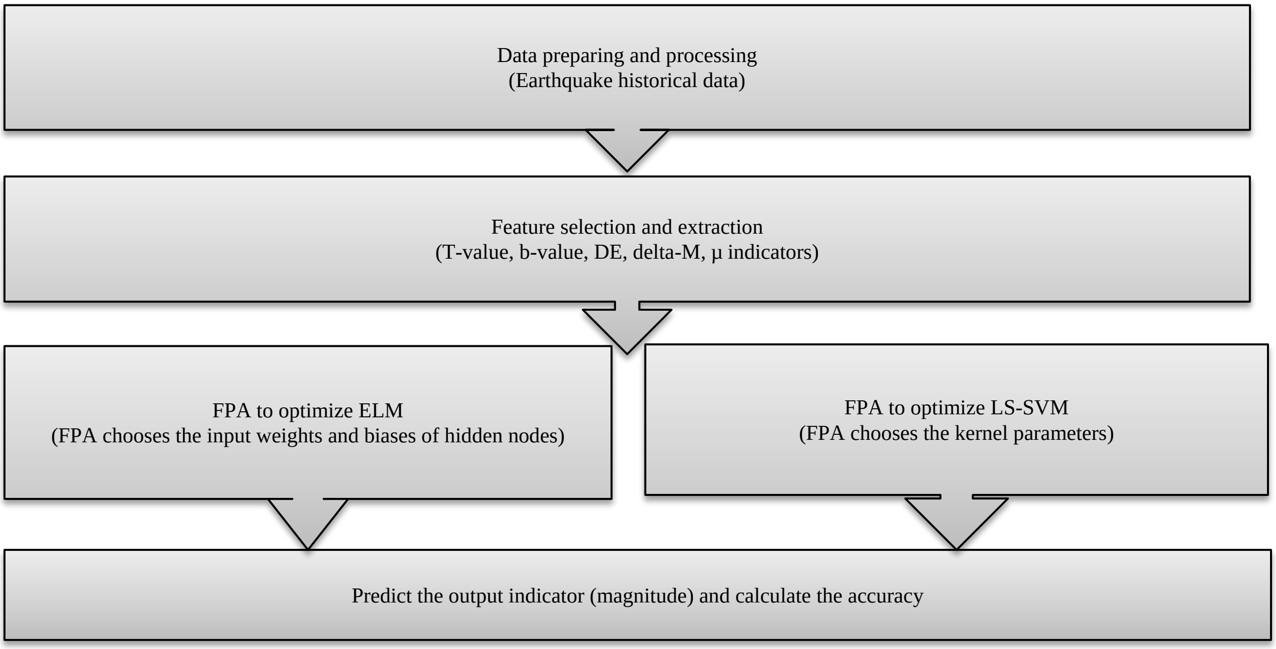
Data preparing and processing
(Earthquake historical data)
Feature selection and extraction
(T-value, b-value, DE, delta-M, μ indicators)
FPA to optimize ELM
(FPA chooses the input weights and biases of hidden nodes)
FPA to optimize LS-SVM
(FPA chooses the kernel parameters)
Predict the output indicator (magnitude) and calculate the accuracy

Fig. 1. The Phases of Proposed Model.

VI. Results and Discusssion

After processing data and introducing the indicators to the proposed models, the performance of each model is estimated using four performance evaluation criteria RMSE, MAE, SMAPE, and PMRE, which can be calculated through the following formulas [13]:

$$\text{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{Ai}-\text{Fi})} \qquad (26)$$

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}|\text{Ai}-\text{Fi}| \qquad (27)$$

$$\text{SMAPE}=\frac{\sum_{i=1}^{n}|\text{Fi}-\text{Ai}|}{\sum_{i=1}^{n}\text{Ai}+\text{Fi}} \qquad (28)$$

$$\text{PMRE}=\frac{100}{n}\sum_{i=1}^{n}\left|\frac{\text{Ai}-\text{Fi}}{\text{Fi}}\right| \qquad (29)$$

First data is divided into 70% for training and 30% for testing and the results showed that the accuracy of FPA-ELM is higher comparing to solely using ELM, as shown in Fig. 2.

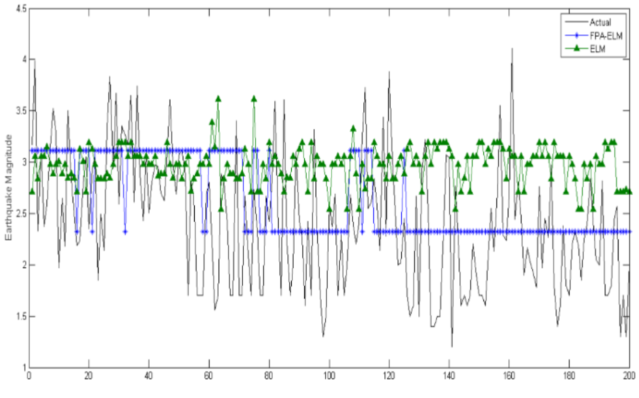

Fig. 2. ELM Accuracy VS FPA-ELM Accuracy when Data is divided into 70% for Training and 30% for Testing.

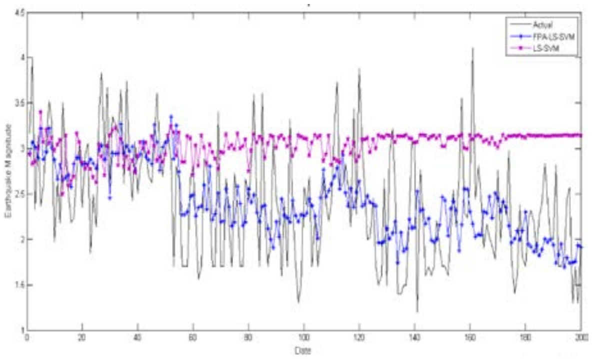

Fig. 3. LS-SVM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 70% for Training and 30% for Testing.

And the performance of LS-SVM became higher and better after optimizing LS-SVM with FPA as shown in Fig. 3.

After the experiment, the results showed that the performance of LS-SVM is better than the performance of ELM, according to this, FPA-LS-SVM accuracy is higher than FPA-ELM one. This is obvious from the following Fig. 4.

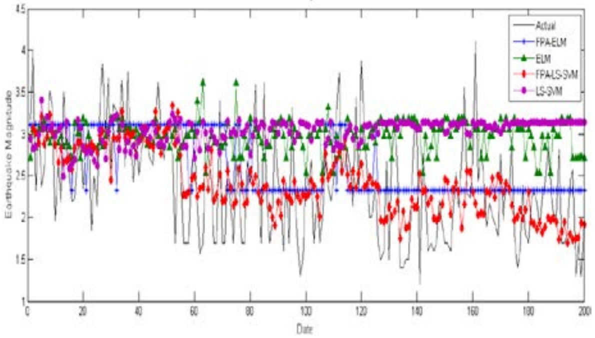

Fig. 4. FPA-ELM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 70% for Training and 30% for Testing.

TABLE II. EVALUATION CRITERIA FOR THE MODELS WHEN DATA IS DIVIDED INTO 70% FOR TRAINING AND 30% FOR TESTING

evaluation test	Models			
	ELM	FPA- ELM	LS-SVM	FPA-LS-SVM
RMSE_test	0.88057	0.645895	0.93642	0.540368
MAE_test	0.73833	0.525306	0.789076	0.434484
SMAPE_test	0.02346	0.030137	0.022373	0.034936
PMRE_test	38.0774	25.12757	40.83974	19.47929

The values of evaluation tests were as follow in Table II:

The column chart that provides the rate of RMSE evaluation criteria for all models when data is divided into 70% for training and 30% for testing is applied as shown in Fig. 5.

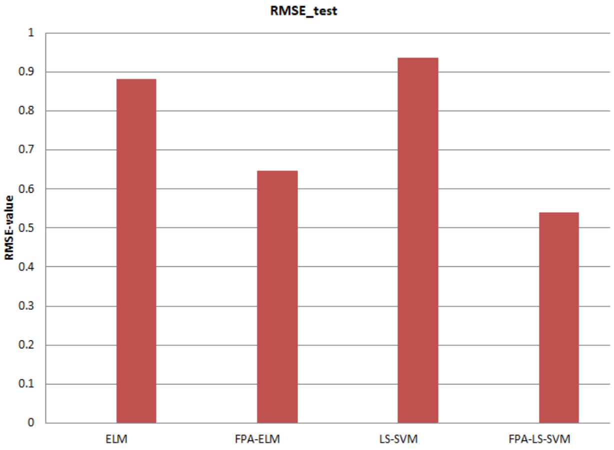

Fig. 5. RMSE Evaluation Criteria for All Models when Data is divided into 70% for Training and 30% for Testing.

The column chart that provides the rate of MAE evaluation criteria for all models when data is divided into 70% for training and 30% for testing is applied as shown in Fig. 6.

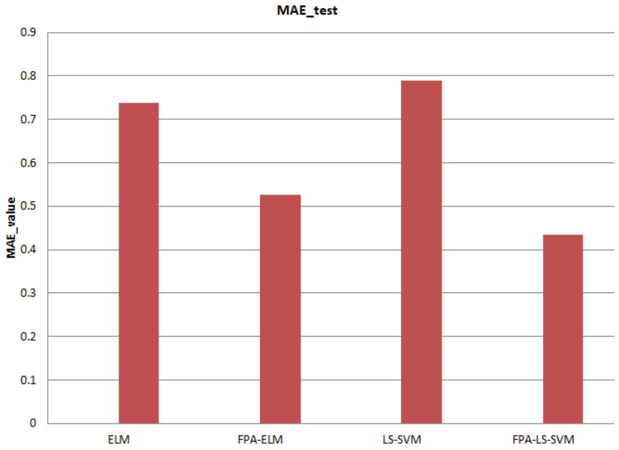

Fig. 6. MAE Evaluation Criteria for All Models when Data is divided into 70% for Training and 30% for Testing.

The column chart that provides the rate of SMAPE evaluation criteria for all models when data is divided into 70% for training and 30% for testing is applied as shown in Fig. 7.

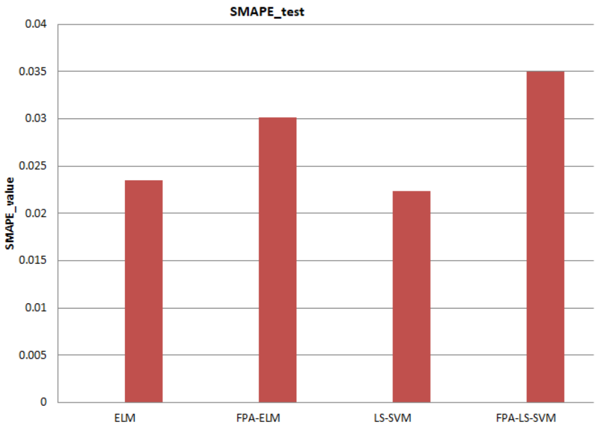

Fig. 7. SMAPE Evaluation Criteria for All Models when Data is divided into 70% for Training and 30% for Testing.

The column chart that provides the rate of PMRE evaluation criteria for all models when data is divided into 70% for training and 30% for testing is applied as shown in Fig. 8.

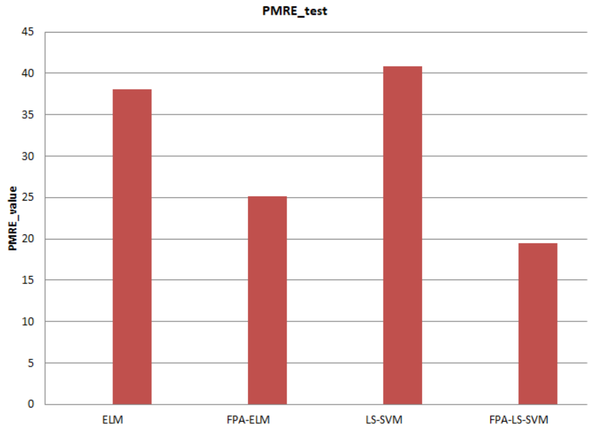

Fig. 8. PMRE Evaluation Criteria for All Models when Data is divided into 70% for Training and 30% for Testing.

Second, data is divided into 80% for training and 20% for testing and the results also showed that the accuracy of FPA-ELM is higher comparing to solely using ELM, as shown in Fig. 9.

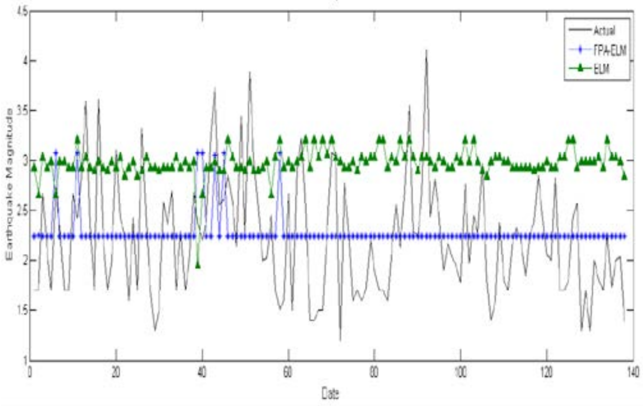

Fig. 9. ELM Accuracy VS FPA-ELM Accuracy when Data is divided into 80% for Training and 20% for Testing.

And the performance of LS-SVM became higher and better after optimizing LS-SVM with FPA as shown in Fig. 10.

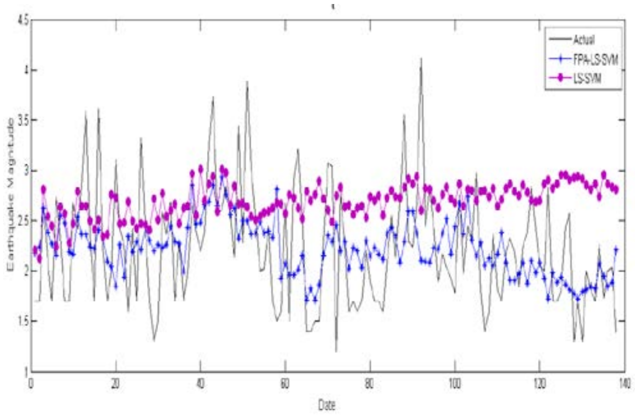

Fig. 10. LS-SVM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 80% for Training and 20% for Testing.

And FPA-LS-SVM accuracy is higher than FPA-ELM one. This is obvious from the following Fig. 11.

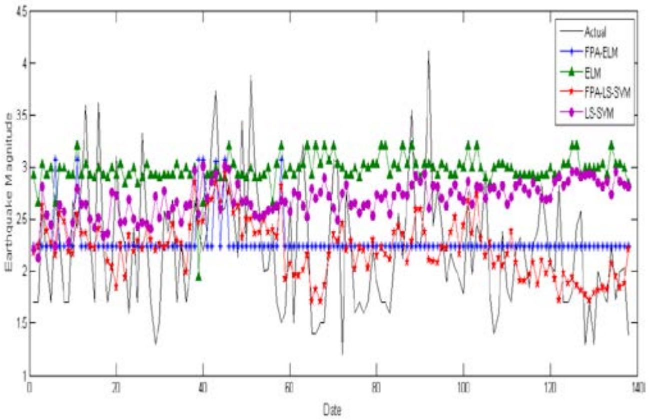

Fig. 11. FPA-ELM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 80% for Training and 20% for Testing.

In this case the values of evaluation criteria were as follow in Table III:

TABLE III. EVALUATION CRITERIA FOR THE MODELS WHEN DATA IS DIVIDED INTO 80% FOR TRAINING AND 20% FOR TESTING

evaluation test	**Models**			
	ELM	**FPA-ELM**	**LS-SVM**	**FPA-LS-SVM**
RMSE_test	0.962142	0.598239	0.769138	0.565476
MAE_test	0.845038	0.4836	0.649227	0.447226
SMAPE_test	0.03279	0.045286	0.03779	0.046786
PMRE_test	45.68055	23.49402	34.71175	20.931907

The column chart that provides the rate of RMSE evaluation criteria for all models when data is divided into 80% for training and 20% for testing is applied as shown in Fig. 12.

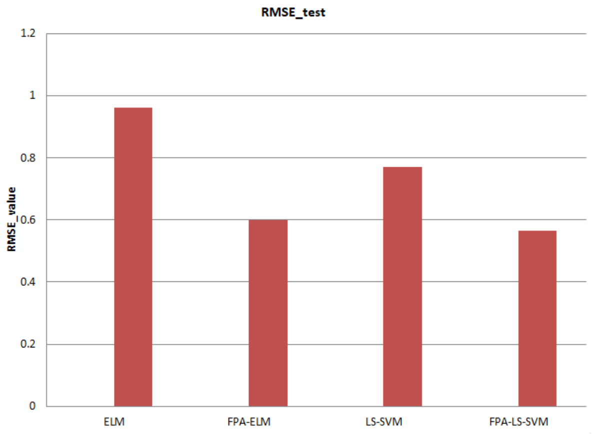

Fig. 12. RMSE Evolution Criteria for All Models when Data is divided into 80% for Training and 20% for Testing

The column chart that provides the rate of MAE evaluation criteria for all models when data is divided into 80% for training and 20% for testing is applied as shown in Fig. 13.

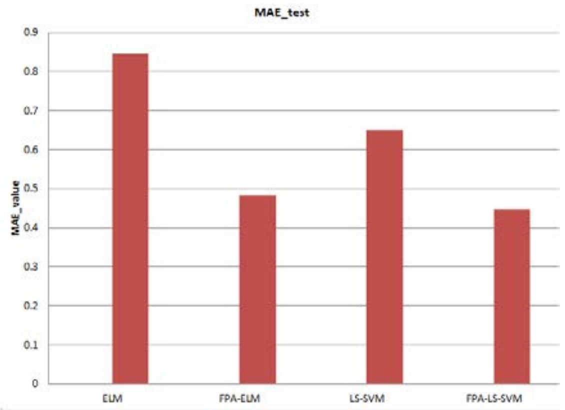

Fig. 13. MAE Evaluation Criteria for All Models when Data is divided into 80% for Training and 20% for Testing.

The column chart that provides the rate of SMAPE evaluation criteria for all models when data is divided into 80% for training and 20% for testing is applied as shown in Fig. 14.

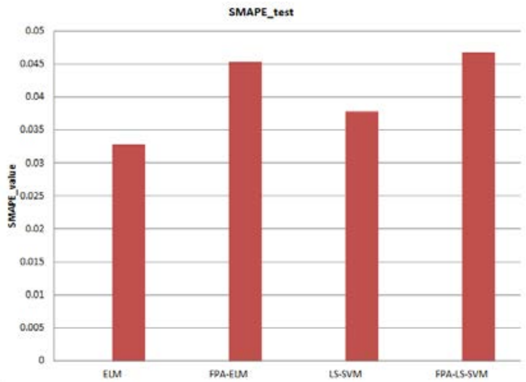

Fig. 14. SMAPE Evaluation Criteria for All Models when Data is divided into 80% for Training and 20% for Testing.

The column chart that provides the rate of PMRE evaluation criteria for all models when data is divided into 80% for training and 20% for testing is applied as shown in Fig. 15.

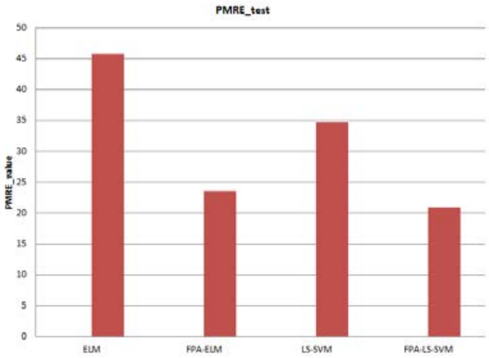

Fig. 15. PMRE Evaluation Criteria for All Models when Data is divided into 80% for Training and 20% for Testing.

At last data is divided into 90% for training and 10% for testing and the results also showed that the accuracy of FPA-ELM is higher comparing to solely using ELM, as shown in Fig. 16.

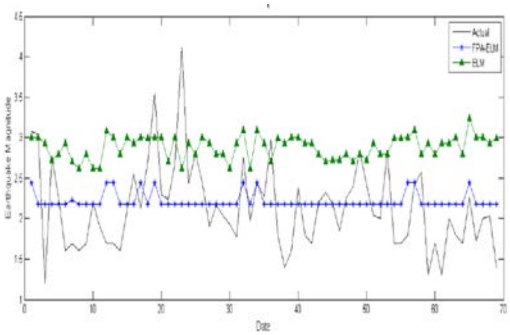

Fig. 16. ELM Accuracy VS FPA-ELM Accuracy when Data is divided into 90% for Training and 10% for Testing.

And the performance of LS-SVM became higher and better after optimizing LS-SVM with FPA as shown in Fig. 17.

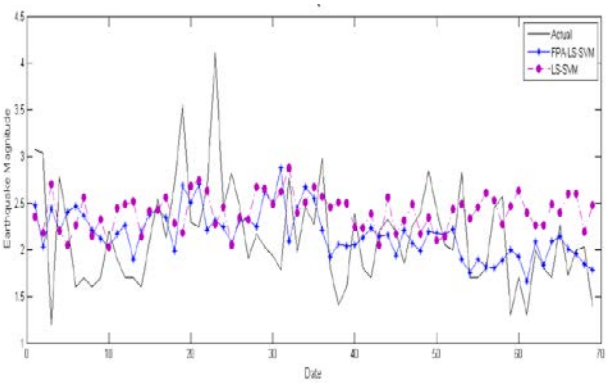

Fig. 17. LS-SVM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 90% for Training and 10% for Testing.

And also FPA-LS-SVM accuracy is higher than FPA-ELM one. This is obvious from the following Fig. 18.

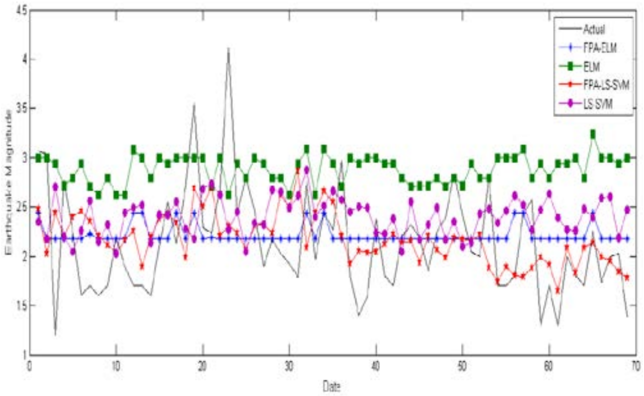

Fig. 18. FPA-ELM Accuracy VS FPA-LS-SVM Accuracy when Data is divided into 90% for Training and 10% for Testing.

In this case the values of evaluation criteria were as follow in Table IV:

TABLE IV. EVALUATION CRITERIA FOR THE MODELS WHEN DATA IS DIVIDED INTO 90% FOR TRAINING AND 10% FOR TESTING

evaluation test	**Models**			
	ELM	**FPA- ELM**	**LS-SVM**	**FPA-LS-SVM**
RMSE_test	0.929831	0.529094	0.651456	0.537101
MAE_test	0.813929	0.417803	0.53825	0.428999
SMAPE_test	0.065945	0.096061	0.082919	0.097089
PMRE_test	44.65868	21.13068	28.38148	20.80531

The column chart that provides the rate of RMSE evaluation criteria for all models when data is divided into 90% for training and 10% for testing is applied as shown in Fig. 19.

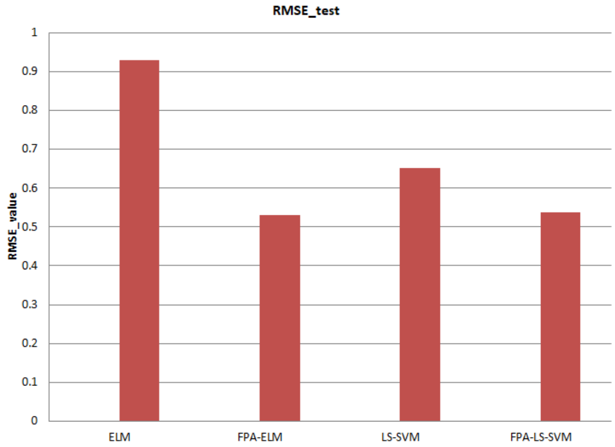

Fig. 19. RMSE Evolution Criteria for All Models when Data is divided into 90% for Training and 10% for Testing.

The column chart that provides the rate of MAE evaluation criteria for all models when data is divided into 90% for training and 10% for testing is applied as shown in Fig. 20.

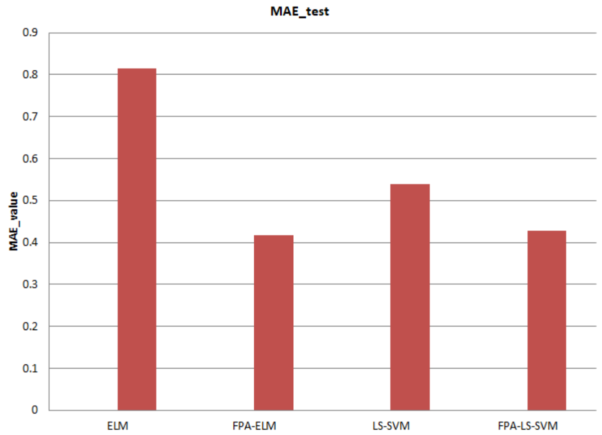

Fig. 20. MAE Evaluation Criteria for All Models when Data is divided into 90% for Training and 10% for Testing.

The column chart that provides the rate of SMAPE evaluation criteria for all models when data is divided into 90% for training and 10% for testing is applied as shown in Fig. 21.

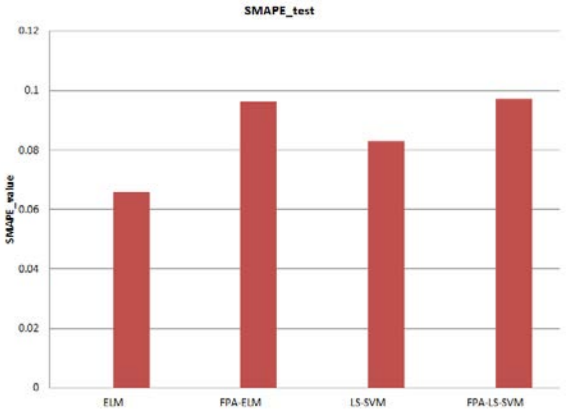

Fig. 21. SMAPE Evaluation Criteria for All Models when Data is divided into 90% for Training and 10% for Testing.

The column chart that provides the rate of PMRE evaluation criteria for all models when data is divided into 90% for training and 10% for testing is applied as shown in Fig. 22.

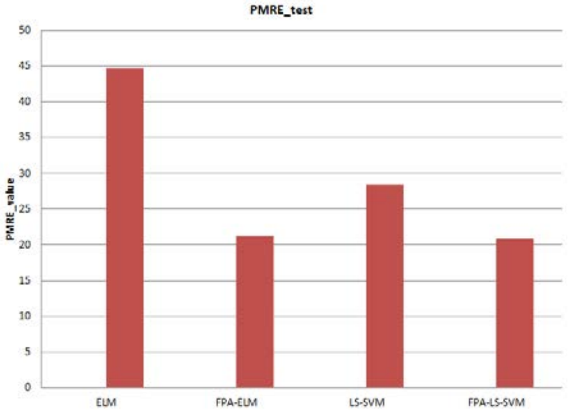

Fig. 22. PMRE Evaluation Criteria for All Models when Data is divided into 90% for Training and 10% for Testing.

VII. Conclusion

In this paper, two hybrid models, FPA-ELM and FPA-SVM, were proposed to forecast earthquake magnitude in the southern California region. Some seismic indicators were generated mathematically and statistically from the dataset to be employed as inputs for the proposed models. The proposed models were evaluated using four criteria. These criteria are RMSE, MAE, SMAPE, and PMRE. The experimental results showed that the accuracy of both ELM and LS-SVM were increased after optimizing it by FPA algorithm. The performance of proposed FPA-LS-SVM outperformed the FPA-ELM model according to all compared criteria. Also, FPA-LS-SVM is the best in reducing the false alarm ratio in earthquake prediction.

References

[1] K. M. Asim, F. Martínez-Álvarez, A. Basit, and T. Iqbal, “Earthquake magnitude prediction in Hindukush region using machine learning techniques,” Nat. Hazards, vol. 85, no. 1, pp. 471–486, 2017, doi: 10.1007/s11069-016-2579-3.

[2] G. Asencio-Cortés, F. Martínez-Álvarez, A. Troncoso, and A. Morales-Esteban, “Medium–large earthquake magnitude prediction in Tokyo with artificial neural networks,” Neural Comput. Appl., vol. 28, no. 5, pp. 1043–1055, 2017, doi: 10.1007/s00521-015-2121-7.

[3] T. L. Chin, C. Y. Huang, S. H. Shen, Y. C. Tsai, Y. H. Hu, and Y. M. Wu, “Learn to Detect: Improving the Accuracy of Earthquake Detection,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 11, pp. 8867–8878, 2019, doi: 10.1109/TGRS.2019.2923453.

[4] S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza, “STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI,” IEEE Access, vol. 7, pp. 179464–179476, 2019, doi: 10.1109/ACCESS.2019.2947848.

[5] R. Mallouhy, C. A. Jaoude, C. Guyeux, and A. Makhoul, “Major earthquake event prediction using various machine learning algorithms,” 6th Int. Conf. Inf. Commun. Technol. Disaster Manag. ICT-DM 2019, pp. 1–7, 2019, doi: 10.1109/ICT-DM47966.2019.9032983.

[6] M. Maya and W. Yu, “Short-term prediction of the earthquake through neural networks and meta-learning,” 2019 16th Int. Conf. Electr. Eng. Comput. Sci. Autom. Control. CCE 2019, pp. 1–6, 2019, doi: 10.1109/ICEEE.2019.8884562.

[7] M. Monterrubio-Velasco, J. C. Carrasco-Jimenez, O. Castillo-Reyes, F. Cucchietti, and J. De La Puente, “A Machine Learning Approach for Parameter Screening in Earthquake Simulation,” Proc. - 2018 30th Int. Symp. Comput. Archit. High Perform. Comput. SBAC-PAD 2018, pp. 348–355, 2019, doi: 10.1109/CAHPC.2018.8645865.

[8] A. Vahaplar, B. T. Tezel, R. Nasiboglu, and E. Nasibov, “A monitoring system to prepare machine learning data sets for earthquake prediction based on seismic-acoustic signals,” 9th Int. Conf. Appl. Inf. Commun. Technol. AICT 2015 - Proc., pp. 44–47, 2015, doi: 10.1109/ICAICT.2015.7338513.

[9] O. Hegazy, O. S. Soliman, and M. A. Salam, “Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction,” Int. J. Adv. Comput. Res., vol. 5, no. 18, pp. 35–45, 2015.

[10] A. Panakkat and H. Adeli, “Neural network models for earthquake magnitude prediction using multiple seismicity indicators,” Int. J. Neural Syst., vol. 17, no. 1, pp. 13–33, 2007, doi: 10.1142/S0129065707000890.

[11] H. Adeli and A. Panakkat, “A probabilistic neural network for earthquake magnitude prediction,” Neural Networks, vol. 22, no. 7, pp. 1018–1024, 2009, doi: 10.1016/j.neunet.2009.05.003.

[12] M. Moustra, M. Avraamides, and C. Christodoulou, “Artificial neural networks for earthquake prediction using time series magnitude data or Seismic Electric Signals,” Expert Syst. Appl., vol. 38, no. 12, pp. 15032–15039, 2011, doi: 10.1016/j.eswa.2011.05.043.

[13] And M. A. S. Osman Hegazy, Omar S. Soliman, “FPA-ELM Model for Stock Market Prediction,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 5, no. 2, pp. 1050–1063, 2015.

[14] M. Li, Z. Lieyuan, and S. Yaolin, “Attempts at using seismicity indicators for the prediction of large earthquakes by genetic algorithm-neural network method,” Unpubl. Manuscr., no. 1, pp. 483–489, 1998.

[15] L. MacEda, J. Llovido, and A. Satuito, “Categorization of earthquake-related tweets using machine learning approaches,” Proc. - 2018 Int. Symp. Comput. Consum. Control. IS3C 2018, no. May 2011, pp. 229–232, 2019, doi: 10.1109/IS3C.2018.00065.

[16] W. Li, N. Narvekar, N. Nakshatra, N. Raut, B. Sirkeci, and J. Gao, “Seismic data classification using machine learning,” Proc. - IEEE 4th Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2018, pp. 56–63, 2018, doi: 10.1109/BigDataService.2018.00017.

[17] G. Rajguru, Y. S. Bhadauria, and S. Mukhopadhyay, “Estimation of Earthquake Source Parameters Using Machine Learning Techniques,” 2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018, pp. 1–7, 2018, doi: 10.1109/ICCCNT.2018.8493922.

[18] Q. Wang, Y. Guo, L. Yu, and P. Li, “Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach,” IEEE Trans. Emerg. Top. Comput., vol. 8, no. 1, pp. 148–158, 2020, doi: 10.1109/TETC.2017.2699169.

[19] J. Reyes, A. Morales-Esteban, and F. Martínez-Álvarez, “Neural networks to predict earthquakes in Chile,” Appl. Soft Comput. J., vol. 13, no. 2, pp. 1314–1328, 2013, doi: 10.1016/j.asoc.2012.10.014.

[20] F. Martínez-Álvarez, J. Reyes, A. Morales-Esteban, and C. Rubio-Escudero, “Determining the best set of seismicity indicators to predict earthquakes. Two case studies: Chile and the Iberian Peninsula,” Knowledge-Based Syst., vol. 50, pp. 198–210, 2013, doi: 10.1016/j.knosys.2013.06.011.

[21] F. Zhou and X. Zhu, “Earthquake prediction based on LM-BP neural network,” Lect. Notes Electr. Eng., vol. 270 LNEE, no. VOL. 1, pp. 13–20, 2014, doi: 10.1007/978-3-642-40618-8_2.

[22] A. Morales-Esteban, F. Martínez-Álvarez, and J. Reyes, “Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence,” Tectonophysics, vol. 593, pp. 121–134, 2013, doi: 10.1016/j.tecto.2013.02.036.

[23] Y. Wang, Y. Chen, and J. Zhang, “The application of RBF neural network in earthquake prediction,” 3rd Int. Conf. Genet. Evol. Comput. WGEC 2009, pp. 465–468, 2009, doi: 10.1109/WGEC.2009.81.

[24] A. S. N. Alarifi, N. S. N. Alarifi, and S. Al-Humidan, “Earthquakes magnitude predication using artificial neural network in northern Red Sea area,” J. King Saud Univ. - Sci., vol. 24, no. 4, pp. 301–313, 2012, doi: 10.1016/j.jksus.2011.05.002.

[25] G. Asencio-Cortés, A. Morales-Esteban, X. Shang, and F. Martínez-Álvarez, “Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure,” Comput. Geosci., vol. 115, pp. 198–210, 2018, doi: 10.1016/j.cageo.2017.10.011.

[26] K. Tan and X. Cai, “Prediction of earthquake in Yunnan region based on the AHC over sampling,” 2010 Chinese Control Decis. Conf. CCDC 2010, pp. 2449–2452, 2010, doi: 10.1109/CCDC.2010.5498782.

[27] E. Florido, F. Martínez-Álvarez, A. Morales-Esteban, J. Reyes, and J. L. Aznarte-Mellado, “Detecting precursory patterns to enhance earthquake prediction in Chile,” Comput. Geosci., vol. 76, pp. 112–120, 2015, doi: 10.1016/j.cageo.2014.12.002.

[28] M. Last, N. Rabinowitz, and G. Leonard, “Predicting the maximum earthquake magnitude from seismic data in Israel and its neighboring countries,” PLoS One, vol. 11, no. 1, pp. 1–16, 2016, doi: 10.1371/journal.pone.0146101.

[29] M. H. Rafiei and H. Adeli, “NEEWS: A novel earthquake early warning model using neural dynamic classification and neural dynamic optimization,” Soil Dyn. Earthq. Eng., vol. 100, no. February, pp. 417–427, 2017, doi: 10.1016/j.soildyn.2017.05.013.

[30] T. Kerh, Y. H. Su, and A. Mosallam, “Incorporating global search capability of a genetic algorithm into neural computing to model seismic records and soil test data,” Neural Comput. Appl., vol. 28, no. 3, pp. 437–448, 2017, doi: 10.1007/s00521-015-2077-7.

[31] M. Mirrashid, “Earthquake magnitude prediction by adaptive neurofuzzy inference system (ANFIS) based on fuzzy C-means algorithm,” Nat. Hazards, vol. 74, no. 3, pp. 1577–1593, 2014, doi: 10.1007/s11069-014-1264-7.

[32] K. M. Asim, M. Awais, F. Martínez-Álvarez, and T. Iqbal, “Seismic activity prediction using computational intelligence techniques in northern Pakistan,” Acta Geophys., vol. 65, no. 5, pp. 919–930, 2017, doi: 10.1007/s11600-017-0082-1.

[33] Z. Umar, B. Pradhan, A. Ahmad, M. N. Jebur, and M. S. Tehrany, “Earthquake induced landslide susceptibility mapping using an integrated ensemble frequency ratio and logistic regression models in West Sumatera Province, Indonesia,” Catena, vol. 118, no. September 2009, pp. 124–135, 2014, doi: 10.1016/j.catena.2014.02.005.

[34] K. M. Asim, A. Idris, F. Martinez-Alvarez, and T. Iqbal, “Short Term Earthquake Prediction in Hindukush Region Using Tree Based Ensemble Learning,” Proc. - 14th Int. Conf. Front. Inf. Technol. FIT 2016, pp. 365–370, 2017, doi: 10.1109/FIT.2016.073.

[35] O. Hegazy, O.S. Soliman, and M. Abdul Salam, "A Machine Learning Model for Stock Market Prediction", International Journal of Computer Science and Telecommunications, Vol. (4), Issue (12),pp. 17-23, December 2013.

[36] O. Hegazy, O.S. Soliman, and M. Abdul Salam, "LSSVM-ABC Algorithm for Stock Price Prediction", International Journal of Computer Trends and Technology (IJCTT), Vol. (7), Issue (2),pp. 81-92, Jan 2014.

[37] O. Hegazy, O.S. Soliman, and M. Abdul Salam, "Optimizing LS-SVM using Modified Cuckoo Search algorithm (MCS) for Stock Price Prediction", International Journal of Advanced Research in Computer Science and Management Studies, Vol. (3), Issue (2),pp. 204-224, February 2015.

[38] O. Hegazy, O.S. Soliman, and M. Abdul Salam, "Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction", International Journal of Advanced Computer Research Vol.(5), Issue (18),pp.35- 45, March-2015.

[39] H. Abdul-Kader, MA. Salam. Evaluation of Differential Evolution and Particle Swarm Optimization Algorithms at Training of Neural Network for Stock Prediction. Int. Arab. J. e Technol. 2012;2(3):145-51.

[40] R. Salem, M. Abdul Salam, H. Abdelkader and A. Awad Mohamed, "An Artificial Bee Colony Algorithm for Data Replication Optimization in Cloud Environments," in IEEE Access, vol. 8, pp. 51841-51852, 2020, doi: 10.1109/ACCESS.2019.2957436.

[41] EM Badr, M. Abdul Salam, M Ali, H Ahmed, Social Media Sentiment Analysis using Machine Learning and Optimization Techniques, International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 41, August 2019.

[42] MA. Salam, M. Ali, Optimizing Extreme Learning Machine using GWO Algorithm for Sentiment Analysis. International Journal of Computer Applications.;2020, 975:8887.

[43] O.S. Soliman, and M. A. Salam, "A Hybrid BA-LS-SVM Model and Financial Technical Indicators for Weekly Stock Price and Trend Prediction", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), vol. 4, issue 4, pp. 764-781, 2014.