Turkish Journal of Computer and Mathematics Education *Vol.12 No.14 (2021), 2397– 2403*

Research Article

An Enhanced Earthquake Prediction Model Using Long Short-Term Memory

Dr.SivaiahBellamkonda[a], LavanyaSettipalli[b], Dr.Ramachandran Vedantham[c], Manoj Kumar Vemula[d]

[a]Assistant Professor of Computer Science & Engineering, Indian Institute of Information Technology Kottayam, Kerala, India.
[b] Research Scholar of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India.
[c]Associate Professor of Information Technology, VasireddyVenkatadri Institute of Technology, Nambur, Guntur, India.
[d]Assistant Professor of Computer Science & Engineering, Vardhaman College of Engineering, Hyderabad, India.

Abstract: Seismology is a branch of science which deals with the prediction of earthquake in specification with magnitude, location and time of future earthquakes with required parameters and within the stated limits along with exact region of occurrence. Exact earthquake prediction is an active research area but prediction of results with high accuracy is important which are carried out by evaded scientist. In the present work exact prediction of earthquake with high accuracy is tried to achieve using RNN called Long Short Term Memory (LSTM), where the rigorous trained model is helpful for exact prediction of earthquake.

Keywords: Seismology, Earthquake, Long Short Term Memory, Deep Learning.

1. Introduction

One among many natural disasters is earthquake, which eventually occurs lacking warning where people did not find ample of time even to reach. These earthquakes lead towards tremendous loss with respect to serious human injuries, loss of life there by leading to a great loss of economy of a particular country where they occur. Common cause for this kind of destruction produced due to tectonic forces in plates of earth, this leads for production energy which travels from the inner of the earth surface to the top and they are termed as active faults. There exist three different kinds of seismic waves which are labeled as S, P and destructive waves where the first two are referred as body waves**P. Shearer (1999).**

Earthquake prediction for the existing work is classified into four categories:

1) MA (Mathematical Analysis),

2) ML Algorithms (Machine Learning),

3) DL (Deep Learning) and

4) PS investigations (Precursor Signal).

In the first category, a remarkable contribution towards earthquake forecast is given by MA tools **Boucouvalas, M. Gkasios, N. Tselikas, and G. Drakatos(2015)**such as FDL. In the Second category, work escalates information related to earth's atmosphere such as aerosol optical depth (AUD), electromagnetic signals, animal abnormal behavior, cloud images **M. Hayakawa (2015),M. Akhoondzadeh and F. J. Chehrebargh(2016), J. Fan, Z. Chen, L. Yan, J. Gong, and D. Wang (2015), and M. Hayakawa, H. Yamauchi, N. Ohtani, M. Ohta, S. Tosa, T. Asano, A. Schekotov, J. Izutsu, S. M. Potirakis, and K. Eftaxias (2016).** In the Third category, work contribution is taken from analysis of time series methods, M-IFN, ANN, k-NN, SVM, Multi-objective info-fuzzy network, Artificial Neural Network, Support Vector Machine, k-nearest neighbors **M. Last, N. Rabinowitz, and G. Leonard (2016),G. Asencio-Cortes, F. Mart ınez-Alvarez, A. Morales-Esteban, J. Reyes, and A. Troncoso (2015)**. These are useful in prediction of largest earthquakes which are going to occur in coming years based on the seismic events which are recorded previously. In fourth category of work where prediction of both the time and magnitude of seismic events are carried out successfully with the help of deep learning algorithms. This is achieved by the effective involvement of neural networks such as MLP, NN, BP, RNN and FFNN (Multi- layered Perceptron, Neural Network, Back Propagation, Recurrent Neural Network, Feed Forward Neural Network) **J. Mahmoudi, M. A. Arjomand, M. Rezaei, and M. H. Mohammadi (2016), C. Li and X. Liu (2016), S. Saba, F. Ahsan, and S. Mohsin (2016)**.

In this paper, a new approach has been proposed which is an advanced version of RNN using LSTM, a one dimensional input (time stamp series) is given as input to this network which is helpful to carryout predictions showing a correlation for earthquake at different time with respect to latitude and longitude. It is also associated with a strong learning nonlinear capability along with the data having a long term correlation interval

2.Proposed Model

2.1 Flow diagram of proposed model

The proposed model is depicted in the figure 1 below. The model consists of four layer as shown in figure 1 such as LSTM layer followed by dropout layer then LSTM layer followed by activation layer such as SOFTMAX which outputs the predictions.

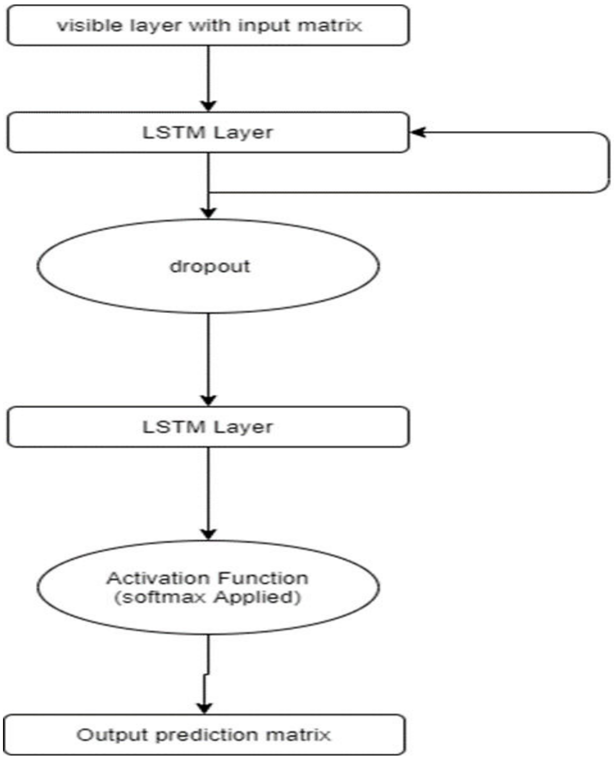

Figure.1 Flow diagram of proposed model

2.2 Attributes

The list of attributes or features are shown in figure 2 and are briefly explained below.

Time Stamp: Time when the event occurred. Times are reported in milliseconds. We indicate the time when the earthquake initiates rupture, which is known as the "origin" time.

Latitude: represented in degrees. Negative values for southern latitudes. An earthquake begins to rupture at a hypocenter which is defined by a position on the surface of the earth (epicenter) and a depth below this point (focal depth). The latitude is the number of degrees north (N) or south (S) of the equator and varies from 0 at the equator to 90 at the poles. The longitude varies from 0 at Greenwich to 180 and the E or W shows the direction from Greenwich.

Longitude: represented in degrees' longitude. Negative values for western longitudes. The longitude is the number of degrees east (E) or west (W) of the prime meridian which runs through Greenwich, England. The longitude varies from 0 at Greenwich to 180 and the E or W shows the direction from Greenwich.

Magnitude: Earthquake magnitude is a measure of the size of an earthquake at its source. It is a logarithmic measure. There are various ways that magnitude may be calculated from seismograms. Different methods are effective for different sizes of earthquakes and different distances between the earthquake source and the recording station.

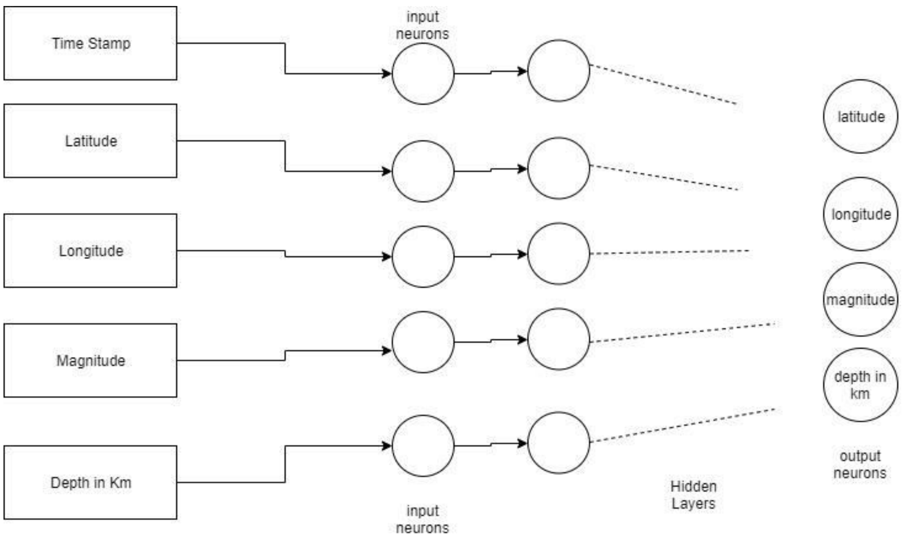

Figure.2 The five input attribute for the model.

Depth: Depth of the event in kilometers. The depth where the earthquake begins to rupture. This depth may be relative to the WGS84 geoid, mean sea-level, or the average elevation of the seismic stations which provided arrival-time data for the earthquake location. The choice of reference depth is dependent on the method used to locate the earthquake, which varies by seismic network.

2.3 Preprocessing

The normalization makes easier both optimization of the loss function and also regularization, because all feature values are at same scale. The Min Max Scaler is the most famous scaling algorithm, and follows the following formula for each feature:

$$\mathrm{Xi} = \frac{\mathrm{Xi} - \min(\mathrm{x})}{\max(\mathrm{x}) - \min(\mathrm{x})} \tag{1}$$

This estimator scales and translates each feature individually such that it is in the given range on the training set in range [0, 1].

2.4 The Typical RNN Architecture

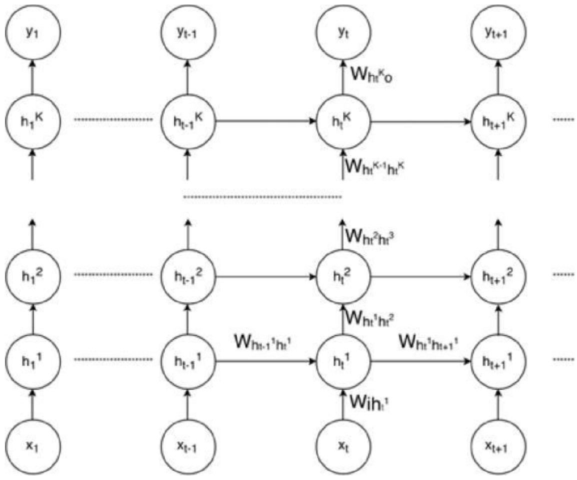

Figure.3RNN Architecture

Above figure 3 depicts the architecture of RNN with n hidden layers. Large differences are traced in the working of both RNN and ANN. While working with RNN the output not only depends on the present input but also depends on the past input also. Let xt denotes the input vector for the time “t”, with the help of its hidden layers h1, h2 .., hk, in order to compute the output yt with respect to the input xt.hkt indicates at a time “t” kth hidden layer, representing number of elements which are associated with its input elements. With the help of nonlinear functions and weight matrices past input data is also effecting the data horizontally to some extent.

2.5 Long Short Term Memory (LSTM)

JurgenSchmidhuber and Sepp Hochreiter proposed an RNN based LSTM model [13]. With the help of network cycles dynamic sequences are captured by RNN, whereas there occur certain constraints with respect to exploding gradients and vanishing problems where gradients are squashed to increase or zero without any bounding to many time stamps during back propagation. A major problem of gradients is overcome with the help of LSTM. Logical functions are implemented in LSTM which consists of three or four network layers or gates.

2.5.1 Architecture of LSTM

The LSTMs are complex neural network structures. LSTMs forward and backward passes information is depicted in [14]. When to start activation in the internal state is initiated in forward pass, a zero value at the gate indicates no activation. During the closure state of both the gates memory cell captures the present activation state, which neither grows nor shrinks. The gradient is enabled with the constant carousel error during the backward pass, in this state the gradient neither explodes not vanishes. The figure 4 shows a chuck of neural networks A, where xt is the input and ht defines its corresponding output values. The loop symbol indicates the passage of information from one step to another step in the network.

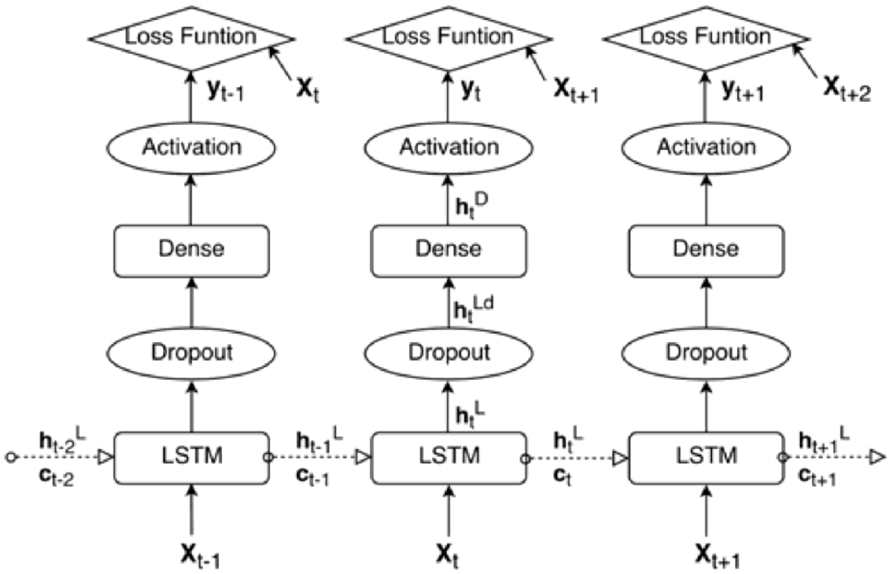

Figure.4 LSTM Architecture

2.5.2 LSTM Single memory cell

LSTMs are simple yet effective structures that use simple concepts of gates for identifying long term memory (LTM) and short term memory (STM). To understand the underlying mechanism, a single memory cell of an LSTM is taken as shown in figure 5. The following gates are used for this task:

1. Forget Gate: Decides which information to keep and which one to forget based on previous LTM.

2. Learn Gate: Decides which information to keep based on Event (current input) and previous STM.

3. Remember Gate: generates the output based on current event and previous STM.

4. Use Gate: Combine important information from Previous Long Term Memory and Previous Short Term Memory to create STM for next and cell and produce output for the current event.

The calculations of the LSTM for each gate are as follows:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_i) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = O_t * \tanh C_t \quad (7)$$

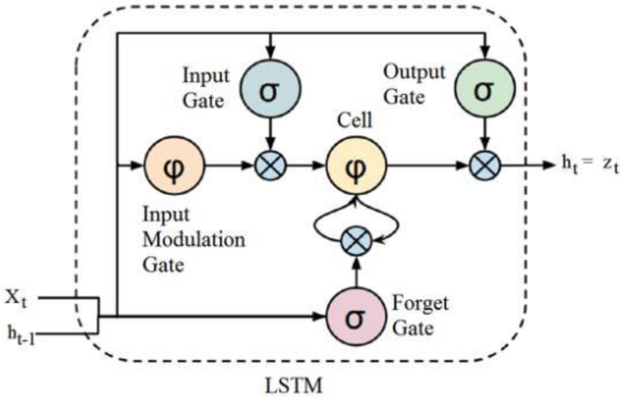

Figure.5 LSTM Memory Cell

Cell state, output gate, forget gate and input gate are denoted as C, O, F, I. φ, σ are the nonlinear and sigmoid which are being mapped with the inputs [-1,1]. W_f,W_i,W_C,W_O are the connection matrices in peephole which are helpful for the interconnection between input, forget output gates with respect to the cell states. In the same manner W_fx,W_ix,W_Cx,W_Ox are the weight matrices which are connected between input vector xt and forget gate, input gate, cell state and output gate respectively. Both the inputs and gates are having the dimension as M ×1 and N ×1 respectively.

2.5.3 Dropout Layer

Applying dropout method at the output of LSTM layer prevents overfitting in the system. This may sometimes affect the performance of the system in a way to get high during training and low during testing. During the occurrence of overfitting the focus of the system is much on historical data due to which the results are too rigid in order to give satisfactory results for the given inputs. The solutions to this overfitting can be adjusted in the proposed system with an addition of dropout [15]. Due to this a randomly selected nodes are turned off temporarily during each training along with its conjoint connections so here we prefer to apply dropout between dense and LSTM layer.

2.5.4 Dense Network

A fully connected dense neural network which consists data required to make necessary prediction gets the input from an LSTM. Here in each layers every neuron is connected to all other neural of the beyond layer. A bias is added after multiplying a matrix with the output of LSTM. In order to learn the nature between the prediction and feature data set dense networks are more suitable compared to other networks.

3.Results and Discussion

From figure 6 and 8, shows MSE after each epoch for LSTM. In FFNN model MSE converges with a low value this resembles that the error rate is a bit low. It is also observed from the figure 7and 9 that R^2 score with respect to each variable is represented in the test set. All obtained positive score values show that a drastic improvement of score for the timestamp is pulled over -0.252 having a score of 59% above the score obtained from FFNN. This seems that LSTMs results are much better than FFNN. Score of R^2 is having a best possible value of 1.0 where it can be negative for some time. Value of y can be predicted with a constant model while neglecting some of the input feature with R^2 score having a value of 0.0. As MSE has been scaled with in value of 0-1, R^2 is observed to be better. When MSE is unable to be scaled then R^2 is considered better.

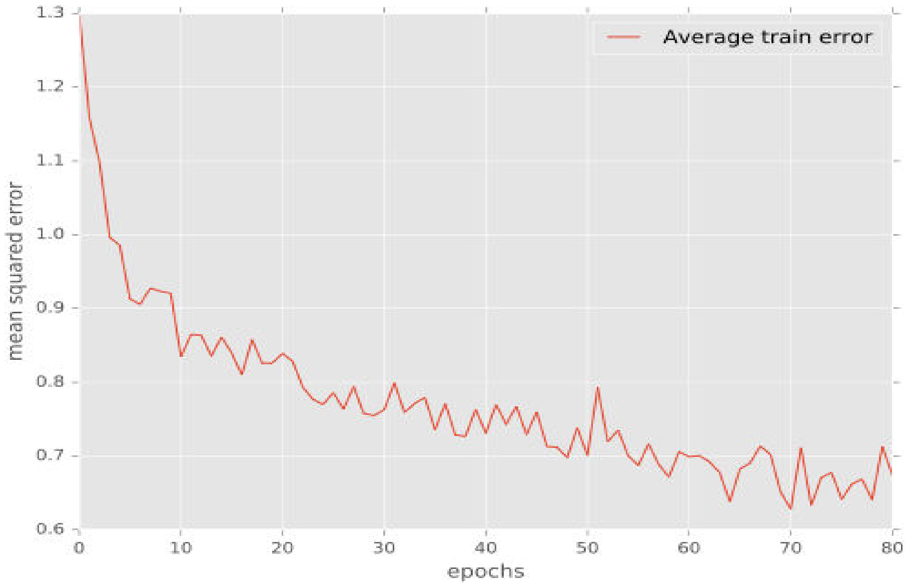

Figure.6MSE after every epoch for LSTM

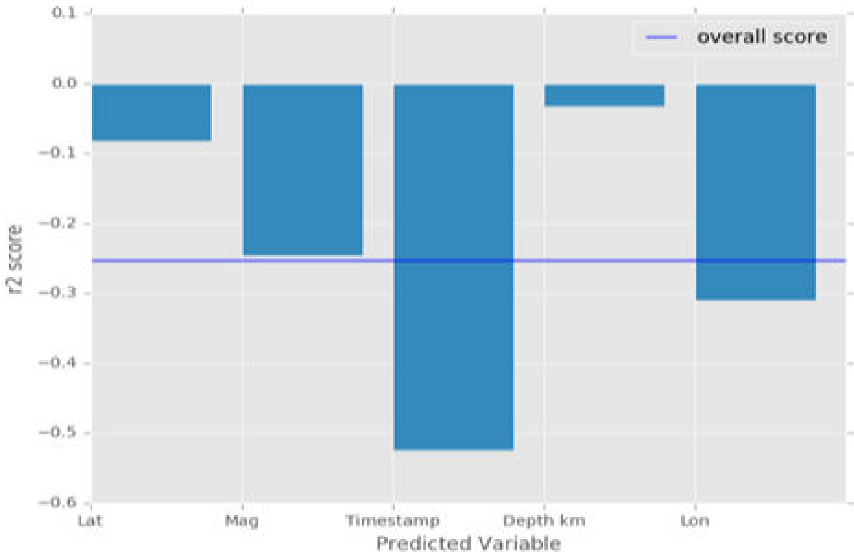

Figure.7 R^2 score for each variable at the end oftraining of LSTM

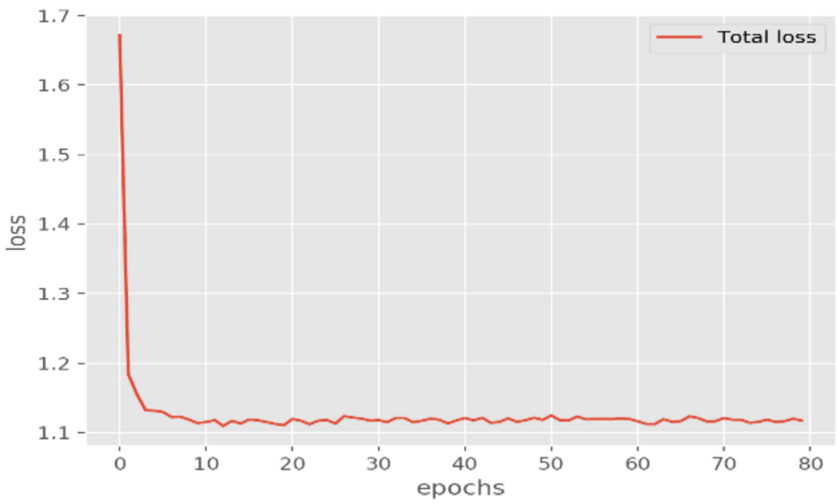

Figure.8 MSE after every epoch for FFNN

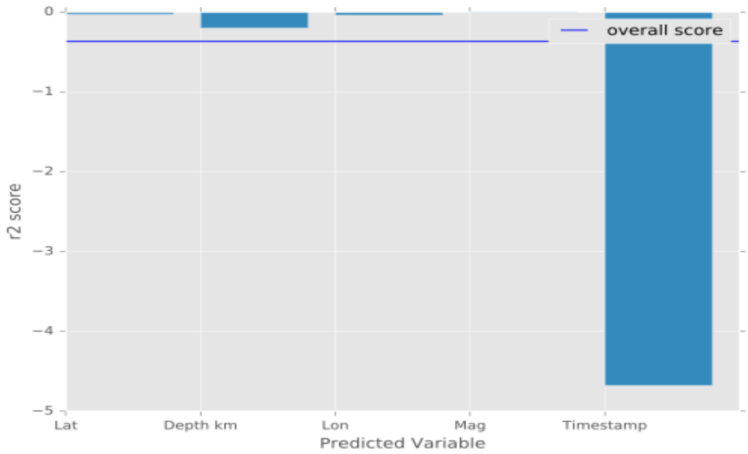

Figure.9 R^2 score for each variable at the end of Training of FFNN

4.Conclusion

Parameters associated with earthquake such as longitude, latitude, time, depth and magnitude of earthquake is predicted with significant results by LSTM model. The obtained results with respect of LSTM are superior when compared with FFNNs results. LSTMs R^2 score obtained are 59% more than the one obtained by FFNN's. Future work can be carried out for the exact prediction of earthquakes in states like Sikkim which are more prone to earthquake. The results can be finely tuned if the present method is combined with deep learning methods.

References

P. Shearer (1999). Introduction to Seismology. Cambridge University Press.

Boucouvalas, M. Gkasios, N. Tselikas, G. Drakatos(2015). Modified Fibonacci-dual-Lucas method for earthquake prediction. In Third International Conference on Remote Sensing and Geoinformation of the Environment. International Society for Optics and Photonics. 95351A–95351A.

M. Hayakawa (2016). Earthquake prediction with electromagnetic phenomena. In The Irago Conference 2015: 360 Degree Outlook on Critical Scientific and Technological Challenges for a Sustainable Society. 1709, 020002.

M. Akhoondzadeh, F. J. Chehrebargh (2015). Feasibility of anomaly occurrence in aerosols time series obtained from modis satellite images during hazardous earthquakes. Advances in Space Research.

J. Fan, Z. Chen, L. Yan, J. Gong, D. Wang (2015). Research on earthquake prediction from infrared cloud images. In Ninth International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR2015). International Society for Optics and Photonics. 98150E–98150E.

M. Hayakawa, H. Yamauchi, N. Ohtani, M. Ohta, S. Tosa, T. Asano, A. Schekotov, J. Izutsu, S. M. Potirakis, K. Eftaxias (2016). On the precursory abnormal animal behavior and electromagnetic effects for the kobe earthquake. Open Journal of Earthquake Research, 5(03), 165.

M. Last, N. Rabinowitz, G. Leonard (2016). Predicting the maximum earthquake magnitude from seismic data in israel and its neighboring countries. PloS one, 11(1), e0146101.

G. Asencio-Cortes, F. Mart ınez-Alvarez, A. Morales-Esteban, J. Reyes, A. Troncoso (2015). Improving earthquake prediction with principal component analysis: application to chile. In International Conference on Hybrid Artificial Intelligence Systems, 393–404.

J. Mahmoudi, M. A. Arjomand, M. Rezaei, M. H. Mohammadi (2016). Predicting the earthquake magnitude using the multilayer perceptron neural network with two hidden layers. Civil Engineering Journal, 2(1), 1–12.

C. Li, X. Liu (2016). An improved pso-bp neural network and its application to earthquake prediction. In Control and Decision Conference (CCDC), China, 3434–3438.

S. Saba, F. Ahsan, S. Mohsin (2016). Bat-ann based earthquake prediction for pakistan region. Soft Computing, 1–9.

Panakkat, H. Adeli (2009). Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. Computer-Aided Civil and Infrastructure Engineering, 24(4), 280–292.

Sepp Hochreiter, Jürgen Schmidhuber (1997). Long Short-Term Memory. Neural Comput. 9(8), 1735– 1780.

Zachary Chase Lipton (2014). A Critical Review of Recurrent Neural Networks for Sequence Learning. CoRR, abs/1506.00019.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.