Journal of Engineering Technology and Applied Sciences, 2024 *e-ISSN: 2548-0391*
Received 17 October 2023, Accepted 24 April 2024 *Vol. 9, No. 2, 47-62*
Published online 30 April 2024, Research Article *doi: 10.30931/jetas.1377481*

Citation: Emeç, M., Özcanhan, M. H., "Application of Artificial Neural Network Methods to Anatolian Plate Earthquake Magnitude and Location Prediction". Journal of Engineering Technology and Applied Sciences 9 (2) 2024 : 47-62.

APPLICATION OF ARTIFICIAL NEURAL NETWORK METHODS TO ANATOLIAN PLATE EARTHQUAKE MAGNITUDE AND LOCATION PREDICTION

Murat Emeç[a], Mehmet Hilal Özcanhan[b]

[a] Department of Computer Science,
Istanbul University, Turkey
*murat.emec@istanbul.edu.tr (*corresponding author)*

[b] Department of Computer Engineering
Dokuz Eylul University, Turkey
hozcanhan@cs.deu.edu.tr

Abstract

Same-region earthquakes usually have a pattern that is difficult to identify clearly. Therefore, time series analysis methods have been proposed for earthquake prediction. Our work attempts to predict three earthquake parameters in the Anatolian Peninsula using pure artificial neural network methods. An optimized BP-NN model and optimally hyper-parameterized LSTM Model have been designed to predict earthquake magnitude, latitude, and longitude. The two models are compared with previous works for their prediction performances using four well-accepted metrics: mean squared error, mean absolute error, median absolute error, and standard deviation. The time, depth, sun, and moon distances to Earth were identified as the most contributing factors in earthquake occurrence through analysis by five different feature extraction algorithms. The date harmed the prediction accuracy. The LSTM model outperformed the BP-NN Model in magnitude prediction with 0.062 MSE. Latitude predictions of both methods were satisfactory and close. However, BP-NN had lower error rates in latitude prediction. However, longitude prediction errors were significant in both models. Therefore, our designs did not successfully predict the exact location of the earthquakes. However, multi-variate, stacked LSTM models are promising in predicting Anatolian Peninsula earthquake magnitudes, but future work is necessary for location and timing predictions.

Keywords: Artificial neural networks, earthquakes, prediction methods, back propagation, LSTM

1. Introduction

An earthquake is a seismic event due to the energy accumulated due to plate movements in the earth's crust. Earthquakes are natural disasters that can devastate human life and sometimes the entire ecosystem where they occur. The recordings of the seismic waves unleashed during an earthquake reveal the earthquake's location, time, and magnitude. People know avoiding earthquakes is impossible but wish to be warned. Thus, it is commonly believed that predicting earthquakes can help reduce losses. Therefore, predicting the timing, magnitude, and location of significant earthquakes has become an essential field of research. Unfortunately, due to the complicated underlying factors, a solution to determine the earthquake's time, magnitude, and location is yet to come [1]. Due to region-specific and complexly interrelated geophysical factors, a simple cause-result analysis is not viable. Hence, earthquake prediction is still a significant research area, even in today's age of Information Technology. Due to region-specific and complexly interrelated geophysical factors, a simple cause-result analysis is not viable. As a result of the highly non-linear cross-correlations between earthquakes and their factors, traditional mathematical, statistical, and machine-learning methods have failed to make sound predictions [2]. Contemporary data mining approaches and Artificial Neural Networks (ANN) have been tried in earthquake prediction [3]. Deep Learning methods and hybrid multi-layer ANN methods have also been used [4]. Some methods, such as multi-layer perceptron (MLP) and back-propagation Neural Networks (BP-NN), have become common [5–7]. While some research focused on fundamental component analysis and data dimensionality reduction [8], most works chose to apply a single algorithm. The problem with one-stage algorithm efforts is the low prediction performance. In our study, however, we use improved ANN models. Instead of the original, single Long Short Term Memory (LSTM) model, we designed a stacked, multi-variate, multi-input, and multi-output unique LSTM model for earthquake prediction. LSTM possesses a strong non-linear learning capability suitable for analyzing correlations in long-term data, such as earthquakes [6]. In the rest of this study, previous earthquake prediction methods are discussed in Section 2. In Section 3, the materials and methods of our model are described. Results and comparisons with earlier works are presented in Section 4. Finally, we conclude in Section 5.

2. Related works

In a highly valued work, Gutenberg and Richter provided a universal mathematical model for the magnitude and frequency of earthquake occurrence after a long period of non-standard earthquake reporting [9]. Many works used the Richter scale in their seismology and earthquake prediction research. However, modern scientific earthquake prediction studies date back to the Haicheng earthquake case in the 1970s [9]. A "Four-Stage Prediction Timeline" approach prevented a much larger casualty number by ordering the evacuation of a city. The four-stage prediction model consisted of a few years long-term, one or two years middle-term, a few months short-term, and daily imminent stages. However, the same prediction success was not obtained in other world regions. Therefore, much new research emerged in the latest earthquake prediction literature, which can be classified under four categories [2]:

- Mathematical and statistical methods [11].
- Studies using precursor signals [12–19].

- Machine learning methods [18–21].
- Deep Learning methods [22–28].

However, previous work showed that earthquake prediction cannot be reduced into simple, mathematical, or statistical non-linear models. Similarly, good results could not be obtained only by using precursors. However, the spreading of computers and sensors has boosted the research of modern techniques, with increasing computing power and abundant data. Our literature research revealed that Deep Learning works are more widespread [22–24]. Namely, BP-NN and ANN models are widely used in many time series analyses and recommended for earthquake-related works. One of the variants of ANN is the popular LSTM [26]. Although some ANN models have been applied to earthquake prediction, to our knowledge, there has been no work comparing BP-NN and LSTM models for earthquake prediction for the Anatolian Plate [27–29].

2.1. Motivation

The motivation behind our work is threefold. People are expecting early warnings about pending earthquakes due to the availability of modern-day technologies. Therefore, further research to satisfy the public's earthquake warning expectations is our responsibility. Secondly, we used Artificial Intelligence Feature Extraction Algorithms to study the contribution of the sun and moon distances because many major Anatolian Plate earthquakes coincide with sun and moon eclipses or phases. Thirdly, previous related works recommend further research into earthquake prediction using modern data processing tools. Therefore, we used pure ANN methods, such as the state-of-the-art BP-NN and LSTM models.

2.2. Contribution

Neural networks can handle large data sets, even if some are missing or corrupted. In addition, some input data have more impact on the outcome than others. Therefore, we decided to identify the most contributing factors or features that can improve the prediction of earthquake magnitudes and locations. In brief, we make the following contributions to Anatolian Plate earthquake prediction:

- A pure ANN approach to earthquake prediction.
- Extensive feature extraction analysis is used to identify the most contributing celestial factors.
- A comparison of modern ANN methods, BP-NN and LSTM, was applied to Anatolian Plate earthquakes.
- The Use of state-of-the-art multi-variate stacked LSTM architecture in earthquake data analysis.
- Prediction of earthquake magnitudes and approximate locations.

3. Materials and methods

The primary material of deep learning research is data. We obtained earthquake data from different sources. The data was first studied for completeness and errors and compared. Data apart from earthquakes were also collected and time-merged with the earthquake data. Feature selection was done to find the most relevant data on earthquakes that should be used in earthquake data analysis. For the methods, we used and compared two modern ANN methods, namely BP-NN and LSTM. The details of the materials and techniques used are elaborated below.

3.1. Materials

3.1.1 Data acquisition

Most earthquakes occur along well-mapped fault lines, which reveal the world's tectonic plate borders. The world's tectonic plates and major fault lines are shown in Fig. 1. Our region of interest (ROI) is the Anatolian Plate, squeezed by the Eurasian, African, and Arabian Plates. Therefore, data on the ROI was obtained from two sources. However, detailed data from the period before the 1970s is not available because seismic sensors were not widespread worldwide. The start of detailed local data collection on Anatolian Plate earthquakes was in 1970, and regular detailed data announcements began in 1975. Fortunately, worldwide earthquake data became globally available in the United States Geological Survey (USGS). Comparing the data from both sources showed that local data mostly matched USGS data. However, the data on earthquakes under magnitude 2.5 is more at USGS. Therefore, we decided to use the USGS earthquake data between 1970 and 2019 in our study.

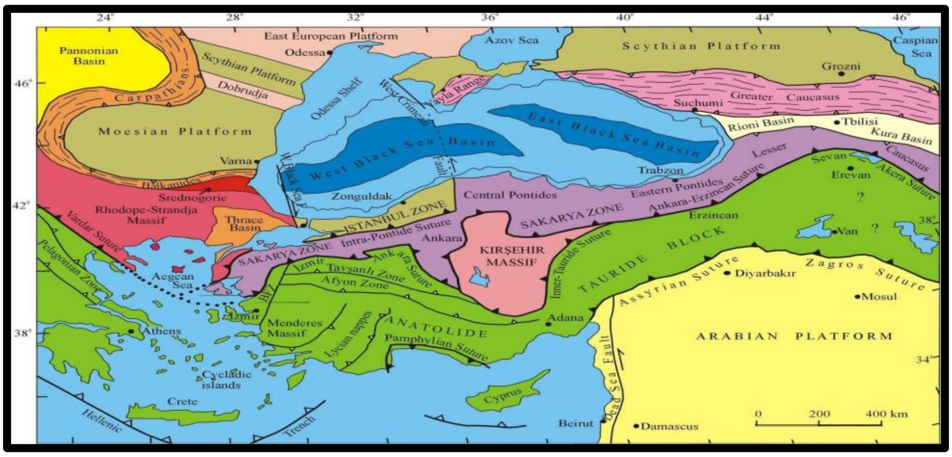

Figure 1 World tectonic plate map, including Turkey's Anatolian plate.

The fault lines are the natural geological borders of the plates. We first traced the world fault lines by breaking them down into none non-overlapping, small-sized zones. Then, the geologic systems affecting Turkey were studied to determine our ROI [28, 29]. Due to the thrust zone in the east, the Anatolian Plate was divided into Eastern and West Anatolia Regions.

Fig.2 shows Turkey's major fault lines, thrust zones, and grabens. The area in Fig. 2 only covers Turkey, between 25°- 45° longitude and 33°- 43° latitude. However, due to the African and Arabian Plate's thrust force on the Anatolian Plate, the scope of our work was adjusted to cover the area bounded by 25°-50° longitude and 35°-47° latitude [30]. Although our ROI can be expanded or shifted, trials with slightly different ROI did not reveal significant changes in the results of our work.

3.1.2 Our ROI data pre-processing

Our ROI data from 1970 to 2019 includes the date, time, latitude, longitude, magnitude, region name, and depth of earthquakes. On the other side, the Sun Distance-Altitude-Azimuth and Moon Distance-Altitude-Azimuth at the time of the earthquakes were obtained with custom-made software [31]. All earthquakes of magnitude larger than 2.5 were time merged with sun and moon data in a single database. Of 18,917 records, 80% were used for training and 20% for testing the designed neural networks. No corrupted (empty, infinite values) or repeated data was used during the analyses.

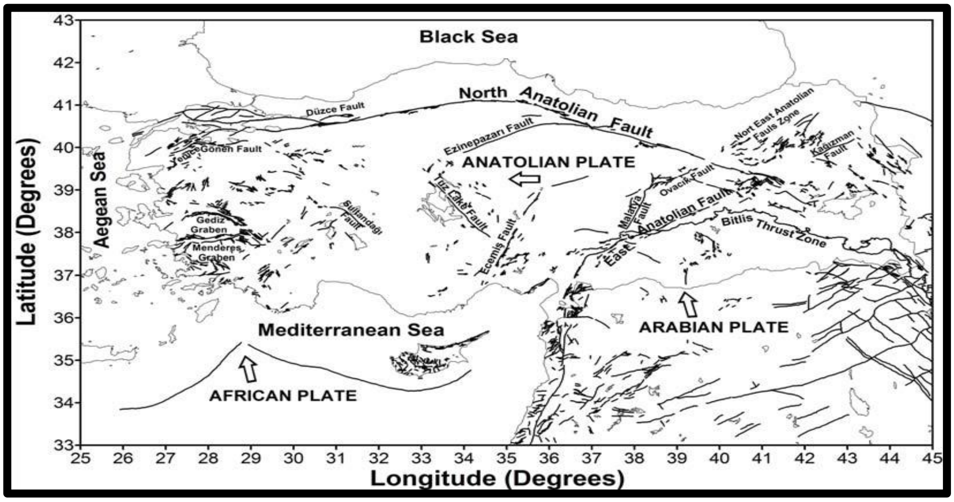

Figure 2 Major fault lines, thrust zones, and grabens in Turkey

3.1.3 Tools used in the analysis

All collected data was inserted in MySQL database table structures. The following computer configuration was used for training the data and testing our models: Intel Core i5 8250U CPU, AMD Radeon 530 GPU, 8 GB of RAM, and Windows 10™ operating system. Cuda tool kit v9.0.176 was used for debugging and optimizing to observe AMD Radeon GPU compatibility. Python programming language was used to design our two custom models. For additional functionalities, the following libraries were used:

- Keras is responsible for the development and training of models.
- ScikiT-learn scans the model and finds the best input parameters by 5-fold cross-validation.
- Tensorflow by Google, Keras, and library for faster training and testing.
- Numpy for performing matrix operations.
- Matplotlib for plotting trend graphs.
- MATLAB R2017b Academic version for BP-NN modeling.
-

3.2. Methods

3.2.1 Our ROI feature selection strategy

As a first step, feature selection was applied to the prepared data using three commonly preferred methods: Decision Tree, CHI^{2,} and PCA. The feature selection results revealed nine features out of the 12 available in each technique. Of the nine features, the scores of the six standard features in the Decision tree and CHI^2 methods are given in Table 1. The most critical or related features were the Sun and Moon distances. In conclusion, feature extraction using three popular methods pointed to six inputs for our model: Sun distance-altitude-azimuth and Moon distance-altitude-azimuth. Our outputs are the expected prediction values: Earthquake longitude, latitude, and magnitude.

Table 1. Decision Tree, CHI^{2,} and PCA top 9 feature extraction scores

Features	**Decision Tree**	**CHI^2**
Sun distance	0.1623	6174.66
Moon distance	0.1617	74.97
Sun altitude	0.1573	1.25
Moon azimuth	0.1555	3.91
Moon altitude	0.151	0.61
Date	0.1092	41.71
Time	0.1019	22.68
Depth	0.0666	15.27

3.2.2 Correlation matrix

In the second step, the correlation matrix of the features against the outputs has been analyzed. The results are shown in Table 2. According to the table, date and moon altitude negatively correlate to our outputs. Meanwhile, time and depth are positively correlated. Therefore, time and depth need to be included in our analysis.

Table 2. Correlation matrix for our outputs: earthquake latitude, longitude, and magnitude

Features	Latitude	Longitude	Magnitude
Date	-0.016	0.001	-0.001
Time	-0.007	0.013	0.033
Moon altitude	-0.015	-0.009	-0.018
Moon azimuth	0.010	0.017	0.027
Moon distance	0.019	0.000	-0.001
Sun altitude	0.066	-0.048	-0.179
Sun azimuth	-0.009	0.012	0.064
Sun distance	0.039	-0.012	-0.006
Depth	**0.051**	**0.259**	**0.278**
Latitude	1.000	0.227	-0.29
Longitude	0.227	1.000	0.516
Magnitude	-0.29	0.516	1.000

3.2.3 Statistical analysis

Factor analysis is a statistical method usually applied to find the correlation between the prepared data and the expected outputs of the designed ANN model [40]. In other words, factor analysis is used to determine new features or verify the most contributing features of an ANN model. Detailed mathematical derivations and explanations can be found in [41]. The results of our factor analysis are given in section 4.1.

3.2.4 Method1: BP-NN model

ANNs are computational models inspired by biological neural networks [32]. Most ANNs have input, hidden, and output layers, as shown in Fig. 3. The number of input nodes depends on the number of features in the data set. While the user determines the number of hidden layer nodes, the number of output nodes depends on the output result set [33]. Training the model can start with a single hidden layer comprising ten nodes. Gradually, hidden neurons are added if the designed network cannot learn the studied data [34]. Search for the correct number of hidden neurons continues until the prediction error is reduced under a predefined value. This methodology is called the adaptive hidden neuron algorithm. The activation function is usually chosen as the non-linear sigmoid function. The scope of this work deliberately does not include the geological reasoning of the seismic events to investigate if purely ANN methods can predict earthquakes. ANNs are computational models inspired by biological neural networks [32]. Most ANNs have input, hidden, and output layers, as shown in Fig. 3. The number of input nodes depends on the features determined in the previous stage. While the user selects the number of hidden layer nodes, output nodes rely on the result set [33]. Training the model can start with a single hidden layer of ten nodes. Gradually, new hidden neurons or layers are added if the designed network cannot learn the studied data [34]. Search for the correct number of hidden neurons continues until the prediction error is reduced under a predefined value. This methodology that we accepted is called an adaptive hidden neuron algorithm. The activation function is usually chosen as the non-linear sigmoid function. Some neurons are fed back to reduce the prediction errors by adjusting the weights or biases of internal neurons, hence the term back-propagation.

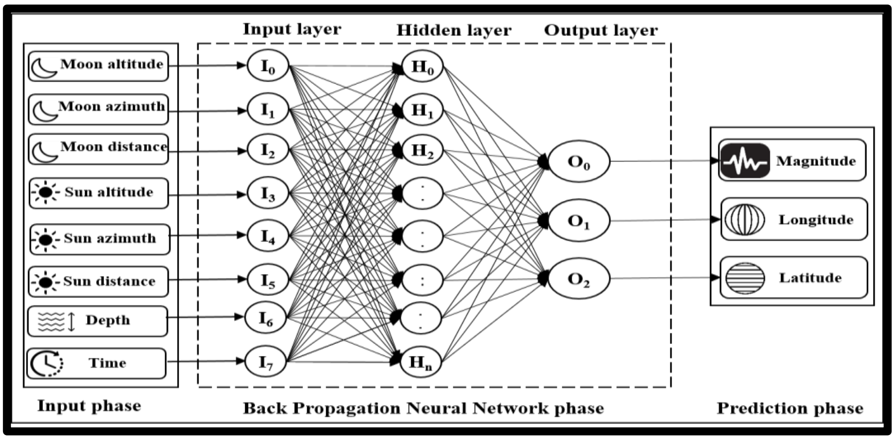

Figure 3. Our BP-NN structure for forecasting earthquakes

The BP-NN architecture of our first method is depicted in Fig. 3. MATLAB R2017b program was utilized to build our model. From the data, 15133 records were allocated for training the model. Three thousand seven hundred eighty-four records were used for testing, not part of the training set. Before feeding the neural network, all data were normalized with the MATLAB" pressed" method. The neural network was configured with one hidden layer and 20 hidden neurons. Training function was used as the adaptive learning training method for back-propagation. Traininggdx is a network training function that updates the weight and bias values of the designed neural network according to gradient descent momentum. The maximum iteration was set at 5000 iterations, and the minimum error rate was set at 0.01.

3.2.5 Method1: Stacked LSTM model

Recently, ANN has been improved by adding extra hidden layers and interconnections between the nodes. With the invention of deeper, interconnected feedback layers, newer approaches were named Recurrent Neural Networks (RNN). LSTM is an RNN variant, shown in Fig. 4. LSTM has been used successfully in handwriting and speech recognition and anomaly detection in computer intrusion detection systems [35–37]. There can be lags of unknown duration between significant events in time series. Still, LSTM can process entire data sequences and cope with the vanishing gradient problem in sequence learning methods [38]. A basic LSTM unit comprises an input, output, and forget gate (Fig. 4). The gates regulate the flow of information into and out of the unit. The unit remembers values over arbitrary time intervals.

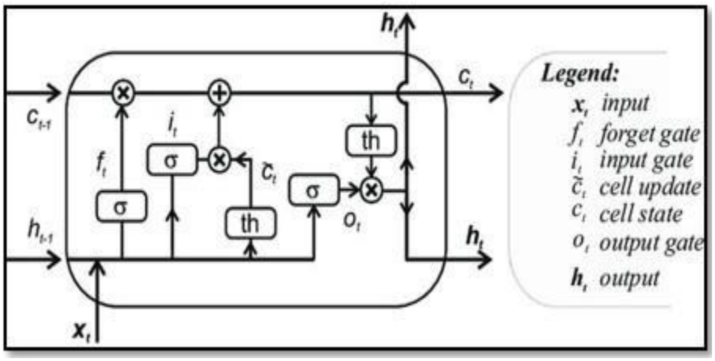

Figure 4. The general construct of an LSTM memory unit

The equations of status and output values of Fig. 4's LSTM unit are as follows [39]:

ft = (Wfhht1 + Wfxxt + bf) (1)

it = (Wihht1 + Wixxt + bi) (2)

c^t = tanh(W c^hht1 + W c^xxt + bc^) (3)

ct = ftct1 + itc^t (4)

ot = (Wohht1 + Woxxt + bo) (5)

ht = otxtanh(ct) (6)

The value ct is the LSTM, where Wi, Wc, and Wo are the weights. Bi, bf, and bo are the biases of the input gate, forget gate, and output gate, respectively. Operator 'x' denotes the pointwise multiplication of two vectors. The symbol represents the sigmoid activation function. The input gate decides the information stored in the unit's new state. The output gate decides what information can be outputted. The forget gate decides the information that will be disregarded in the next state. A value of "1" in the forget gate ft keeps the present data, while 0 drops the info from the LSTM unit. LSTM variant studies resulted in four LSTM types [41]:

- Univariate LSTM Models
- Multivariate LSTM Models
- Multi-Step LSTM Models
- Multivariate Multi-Step LSTM Models

Our detailed inspection of the categories showed that the category two multi-variate model accommodates our study due to the multiple interrelated inputs and outputs [40]. However, early trials showed that a single LSTM layer is inadequate for accurate predictions. Therefore, stacked LSTM layers used in univariate models were integrated into our design. Thus, the innovation in our design is the multi-variate stacked LSTM structure used for earthquake prediction, as shown in Fig. 5. Stacking layers of LSTM cells on top of each other is possible, as shown in our architecture. Our layered model has increased complexity and prediction power, but training times have also increased. The "Adam" layer is added to normalize the output of neurons. For training the model 80% and testing, the remaining 20% of the data was used. The same performance metrics of Method 1 were used in Method 2.

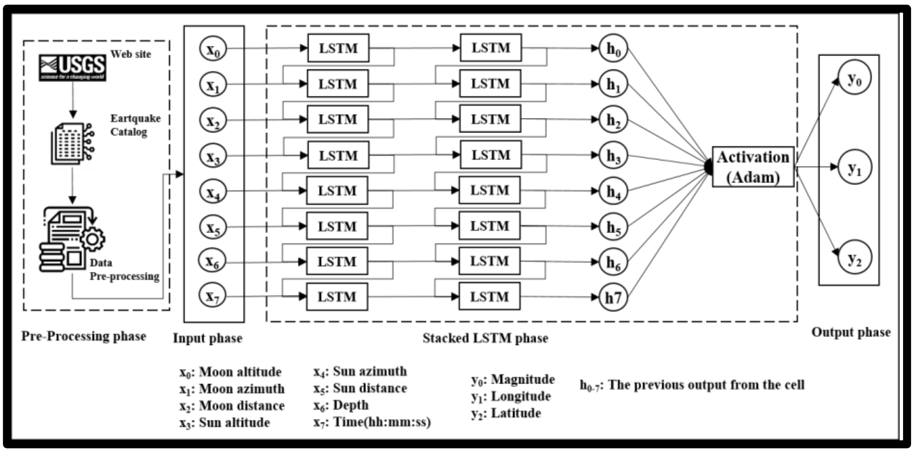

Figure 5. Our Stacked LSTM structure for predicting earthquakes

3.2.6 Performance evaluation metrics

Performance evaluation of prediction proposals is universally based on accuracy and different error parameters. In ANN prediction, performances are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), Median Absolute Error (MEDAE) [2], and Standard Deviation (STD) values [6]. To be in line with previous and future ANN earthquake analyses, we used the same metrics to declare and compare our results with each other and prior works.

4. Research Results and discussion

4.1. Statistical results

Factor analysis has been applied to the input data to determine the correlation between the features and the predicted values. The study investigates the 'weight' of a feature among different groups of features (factors) on the outputs. In Table 3, features positively affecting the outputs are signless, while features negatively affecting the outputs are depicted with negative sign and positive correlation values in Table 3. According to Table 3, time is the most critical contributing feature, while sun azimuth and moon distance are essential for further investigation. Surprisingly, the data negatively contributes to the earthquake prediction analysis in all three-factor groups.

Table 3. Factor analysis results of studied features

Features	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Time	**0.948**	-	-	-	-	-
Sun azimuth	**0.944**	-	-	-	-	-
Magnitude	-	0.814	-0.134	-0.122	-	-
Longitude	-	0.799	-	0.234	-	-
Depth	-	0.616	-	-	-	-
Sun altitude	-	-0.151	0.787	-	-	-
Sun distance	-	-	0.784	-	-	-
Latitude	-	0.146	0.180	0.772	0.124	-0201
Moon distance	-	-	-0.174	0.577	0.163	0.181
Moon azimuth	-	-	-	-	0.794	-0.278
Date	-	-	-	-0.197	-0.603	-0.378
Moon altitude	-	-	-	-	-	**0.848**
% of Variance	15.43	14.39	11.00	8.90	8.50	8.38
Cumulative %	15.43	29.82	40.82	49.72	58.22	66.60

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.

4.2. Artificial neural network method 1: Optimized BP-NN model results

The Hinton diagram method optimized our designed BP-NN model [34]. Hinton diagrams show the weights of the connection values of the input nodes to the hidden nodes in designed BP-NN models. The connection values are represented by the colors and sizes of the rectangles in the diagram. The adaptive optimization resulted in 20 hidden neurons in our design reaching the minimum training error of 0.01. Table 4 presents the performance of the BP-NN method. The BP-NN's MSE was 0.423 in magnitude prediction, 30.912 in longitude, and 1.528 in latitude prediction. The longitude prediction is inaccurate. The reason for the significant error in our longitude prediction originates from the close to zero weight values of the neurons reaching the longitude output. Close to zero weight values are known to produce significant errors in BP-NN. The large error may be due to the mistakes in the reported longitude values. The GPS coordinates of earthquakes are estimations because they occur in large depths, where earth curvature negatively affects the estimations. However, the BP-NN method has striking prediction results when closely predicting earthquake magnitudes, latitudes, and longitudes. Even the sporadic correct predictions can be an answer to people's early warning expectations.

Table 4. Performance metrics of Method 1: BP-NN.

Model Name	Neurons	Epochs	Feature(s)	MSE	STD	MAE
BP-NN	20	5000	Latitude	1.528	1.236	0.927
			Longitude	30.912	5.545	3.661
			Magnitude	0.413	0.649	0.510

4.3. Artificial Neural Network Method 2: Stacked LSTM Model Results

The same inputs and outputs used in Method 1 were used in the multi-variate, many-to-many connected, and stacked LSTM configuration of Fig.5. The model is trained with 80% of the data and tested in the remaining 20%. The model's performance depends on optimizing hyper-parameters (number of neurons, epoch, and batch size). The number of neurons decides the number of nodes in a single LSTM unit and affects the graphical fitting of the data points, preventing over-fitting or under-fitting. The number of complete passes through the model is the number of epochs. Since all of the data cannot be fed into the model in one go, it is divided into batches that can be passed through one training episode of the model. The best performance is found by varying the hyper-parameters and the inputs [31]. Many hyper-parameter combinations have been tried for optimization. The lowest error rates, hence highest performances, were obtained in configurations with (number of neurons, epoch, batch size) = (32, 32, 32), (32, 64, 32), and (8, 64, 64). The impact of the used inputs on the prediction results was tested in the next step. Changing the inputs was most significant in the above three configurations.

In the proposed LSTM model, hyperparameter tuning was accomplished through a random grid search, programmer's heuristics, and previous experiments and literature reports [20-21].

Our best-performing hyperparameters are as follows:

- The number of LSTM hidden layers: 1
- The number of neurons in each LSTM layer: 8
- The number of neurons in each Dense layer: 8
- Dropout rate: 0.1
- Learning rate: 0.001
- Batch size: 64
- Epoch size: 64
- Optimizer: Adam
- Cost function: cross-entropy

The results of the three configurations for varying inputs have been summarized in Tables 5, 6, and 7. Table 5 shows the test results when all of the inputs are included, including all inputs in the first test, which allowed for obtaining a comparison baseline for different input combinations. The averages of the errors and standard deviation were taken to assess the performance of the three configurations. Configuration Test 1 had the worst performance with the highest average errors. With the lowest error averages, the performance of configuration Test 3 was the best. Test 2 had a medium average error.

Table 5. Results of the LSTM tests, when all inputs are used

Test(s)	Neur.	Epochs	Batch size	Feature (s)	MSE	STD	MAE	MEDAE	Avg.	Perfor. Order
Test 1	32	32	32	Latitude	4.848	1.395	1.703	1.002	2.337	Poor
				Longitude	105.500	4.111	9.415	10.030	32.264	
				Magnitude	0.067	0.122	0.229	0.232	0.163	
Test 2	32	64	32	Latitude	4.799	1.414	1.673	1.224	2.278	Medium
				Longitude	86.390	4.605	8.074	6.507	26.394	
				Magnitude	0.072	0.106	0.246	0.252	0.169	
Test 3	**8**	**64**	**64**	Latitude	3.670	1.382	1.327	0.766	**1.786**	Successful
				Longitude	52.990	3.436	6.417	5.772	**17.154**	
				Magnitude	0.062	0.115	0.222	0.244	**0.161**	

In the second testing phase, the moon data recorded during the earthquakes was removed from the inputs. The performance results of the three configurations for this set-up are shown in Table

5. The average error and standard deviation performance of configuration Test 3 were again the best. However, the performance order of Tests 1 and 2 was reversed. Another observation is that the performances in Tables 5 and 6 are very close. For example, every performance metric of configuration Test 3 in Table 5 is very close to its corresponding value in Table 6. Therefore, the absence of the moon variables does not appear to impact predictions critically. Only sun variables were removed from the training data in the third testing phase.

Table 6. Results of the designed model tests without moon, depth, and time parameters.

Test(s)	Neur.	Epochs	Batch size	Feature(s)	MSE	STD	MAE	MEDAE	Avg.	Perfor. Order
Test 1	32	64	32	Latitude	4.917	1.163	1.888	1.851	2.454	Poor
				Longitude	101.5	6.01	8.08	6.05	30.41	
				Magnitude	0.152	0.257	0.300	0.247	0.239	
Test 2	32	32	32	Latitude	4.841	1.155	1.872	1.812	2.42	Medium
				Longitude	112.7	6.53	8.36	5.81	33.35	
				Magnitude	0.174	0.246	0.293	0.250	0.241	
Test 3	**8**	**64**	**64**	Latitude	4.966	1.166	1.891	1.803	**2.457**	Successful
				Longitude	99.96	5.997	8.003	5.906	**29.967**	
				Magnitude	0.168	0.245	0.297	0.257	**0.242**	

Table 6 shows that configuration Test 2 is again the worst performer. The configuration performance Test 3 was again the best. Out of three tests, configuration Test 2 had two worst performances and, therefore, was classified as the non-optimal architecture. As the best performer in all three tests, configuration Test 3 was declared the optimal architecture compared to the BP-NN model. Another observation is the poor performance in longitude prediction in all tests. Because of the significant errors in longitude prediction, our designed models cannot successfully predict the exact location of the earthquakes.

Table 7. Results of the designed model tests without sun, depth, and time parameters.

Test(s)	Neur.	Epochs	Batch size	Feature(s)	MSE	STD	MAE	MEDAE	Avg.	Perfor. Order
Test 1	32	32	32	Latitude	4.909	1.146	1.896	1.768	2.43	Poor
				Longitude	114.3	6.61	8.39	5.51	33.703	
				Magnitude	0.146	0.247	0.291	0.249	0.233	
Test 2	32	64	32	Latitude	4.764	1.131	1.867	1.813	2.394	Medium
				Longitude	106.2	6.28	8.16	5.61	31.563	
				Magnitude	0.151	0.246	0.300	0.256	0.238	
Test 3	**8**	**64**	**64**	Latitude	4.847	1.139	1.884	1.81	**2.42**	Successful
				Longitude	98.49	5.947	7.945	6.042	**29.606**	
				Magnitude	0.144	0.236	0.297	0.221	**0.225**	

5. Comparison of BP-NN and stacked LSTM model results

Table 8 shows the best performance values of the optimized BP-NN model and the Stacked LSTM model with configuration (number of neurons = 8, epoch = 64, Batch size = 64). The table reveals exciting results. The magnitude of significant earthquakes in Turkey was predicted with an MSE performance of 0.144 using our Stacked LSTM model. Latitude prediction of the Stacked LSTM model was also satisfactory, with an average error value of 2.420, but BP-NN performance was better with 1.629. Comparing the two models' performances shows that the Stacked LSTM model has more minor errors in magnitude prediction, and the predicted magnitudes deviate less in LSTM. Therefore, it is safe to declare that our Stacked LSTM performed better in predicting earthquake magnitudes in Turkey. However, the same is not true of latitude predictions. The latitude prediction performance of the BP-NN Model was better than that of the Stacked LSTM Model. However, the latitude performance difference between the two models is insignificant.

Table 8. Comparison of earthquake magnitude prediction errors and the SD of the two methods.

Model Name	Neurons	Epochs	Feature(s)	MSE	STD	MAE
BP-NN	Latitude	1.528	1.236	0.927	0.744	1.108
	Longitude	30.912	5.545	3.661	2.054	10.558
	Magnitude	0.413	0.649	0.510	0.430	0.503
Stacked LSTM	Latitude	3.670	1.382	1.327	0.766	1.786
	Longitude	52.990	3.436	6.417	5.772	17.154
	Magnitude	**0.062**	**0.115**	**0.222**	**0.244**	**0.161**

It is evident from the table that longitude prediction has an unsatisfactorily high error in both models. Therefore, our work did not provide a satisfactory prediction of the exact location of earthquakes in Turkey. However, our latitude range is geographically between 35° and 47°. Hence, the latitude along the quake may be predicted with an error of 20.17% using the LSTM model. Matching the leeway with the present fault lines can approximate the possible location of an earthquake, as in the Haicheng case [9]. Briefly, the magnitude and latitude of earthquakes in Turkey were predicted to the extent that may satisfy the Turkish people's standard "earthquake warning" concern.

6. Conclusion

Earthquake data belonging to Turkey was obtained from USGS from 1970 to 2019. At first, five different feature extraction methods were used to determine the most contributing features. After selecting the best features, the conditioned data were analyzed using two custom-designed ANN methods. The first method used an optimized BP-NN model, and the second used a Stacked, Multi-variate, Many-to-Many LSTM model. A Hinton diagram proved the optimization of the BP-NN model. The optimization of the LSTM model was obtained by varying the configuration hyper-parameters. The best performance values of MSE, MAE, MEDAE, and STD metrics of the two models were compared. When compared, the LSTM model outperformed the BP-NN model in earthquake magnitude prediction. The magnitude prediction of the LSTM model was found to be very satisfactory. However, although very close, the BP-NN model outperformed the LSTM model in latitude prediction. Longitude prediction was way off by a wide margin. However, matching the predicted latitude with the present fault lines can approximate the possible location of an earthquake. In brief, the LSTM method was found to predict the magnitude and approximate latitude of earthquakes in Turkey. We plan to improve the latitude and longitude predictions in future work by adding more seismic inputs and geological precursors to the models.

References

[1] Sobolev, G.A., "Methodology, results, and problem forecasting earthquakes", Her. Russ. Acad. Sci. 85 (2015) : 107–111

[2] Wang, Q., Guo, Y., Yu, L., Li, P., "Earthquake prediction based on spatiotemporal data mining: an LSTM network approach", IEEE Transactions on Emerging Topics in Computing 8 (1) (2017) : 148-158.

[3] Narayanakumar, S., Raja, K., "A BP artificial neural network model for earthquake magnitude prediction in Himalayas, India", Circuits and Systems 7 (11) (2016) : 3456-3468.

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

[5] Mahmoudi, J., Arjomand, M. A., Rezaei, M., Mohammadi, M. H., "Predicting the earthquake magnitude using the multilayer perceptron neural network with two hidden layers", Civil engineering journal 2 (1) (2016) : 1-12.

[6] Li, C., & Liu, X., "An improved PSO-BP neural network and its application to earthquake prediction", Chinese Control and Decision Conference (CCDC) IEEE (2016) : 3434-3438

[7] Saba, S., Ahsan, F., Mohsin, S., "BAT-ANN based earthquake prediction for Pakistan region", Soft Computing 21 (2017) : 5805-5813.

[8] Asencio-Cortés, G., Martínez-Álvarez, F., Morales-Esteban, A., Reyes, J., & Troncoso, A., Improving earthquake prediction with principal component analysis: application to Chile", In: Hybrid Artificial Intelligent Systems: 10th International Conference, HAIS 2015, Bilbao, Spain, Springer International Publishing 10 (2015) : 393-404.

[9] Scholz, C. H., "A physical interpretation of the Haicheng earthquake prediction", Nature 267(5607) (1977) : 121-124.

[10] Dahmen, K., Ertaş, D., Ben-Zion, Y., "Gutenberg-Richter and characteristic earthquake behavior in simple mean-field models of heterogeneous faults", Physical Review E 58(2), (1998) : 1494.

[11] Boucouvalas, A. C., Gkasios, M., Tselikas, N. T., & Drakatos, G., "Modified-Fibonacci-Dual-Lucas method for earthquake prediction", In: Third international conference on remote sensing and geoinformation of the environment (RSCy2015) 9535 (2015) : 400-410. SPIE.

[12] Akhoondzadeh, M., Chehrebargh, F. J., "Feasibility of anomaly occurrence in aerosols time series obtained from MODIS satellite images during hazardous earthquakes", Advances in Space Research, 58(6) (2016) : 890-896.

[13] Hayakawa, M., "Earthquake prediction with electromagnetic phenomena", In: AIP Conference Proceedings, AIP Publishing 1709(1) (2016)

[14] Hayakawa, M., Yamauchi, H., Ohtani, N., Ohta, M., Tosa, S., Asano, T., ... Eftaxias, K., "On the precursory abnormal animal behavior and electromagnetic effects for the Kobe earthquake (M~ 6) on April 12, 2013", Open Journal of Earthquake Research, 5(03) (2016) : 165.

[15] Fan, J., Chen, Z., Yan, L., Gong, J., Wang, D., "Research on earthquake prediction from infrared cloud images", In: MIPPR 2015: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications SPIE 9815 (2015) : 87-92.

[16] Thomas, J. N., Masci, F., Love, J. J., "On a report that the 2012 M 6.0 earthquake in Italy was predicted after seeing an unusual cloud formation", Natural Hazards and Earth System Sciences 15 (5) (2015) : 1061-1068.

[17] Florido, E., Martínez-Álvarez, F., Morales-Esteban, A., Reyes, J., Aznarte-Mellado, J. L., "Detecting precursory patterns to enhance earthquake prediction in Chile", Computers & geosciences 76 (2015) : 112-120.

[18] Asencio-Cortés, G., Martínez-Álvarez, F., Morales-Esteban, A., Reyes, J., "A sensitivity study of seismicity indicators in supervised learning to improve earthquake prediction", Knowledge-Based Systems 101 (2016) : 15-30.

[19] Morales-Esteban, A., Martínez-Álvarez, F., Troncoso, A., Justo, J. L., Rubio-Escudero, C., "Pattern recognition to forecast seismic time series", Expert systems with applications 37 (12), (2010) : 8333-8342.

[20] Panakkat, A., Adeli, H., "Neural network models for earthquake magnitude prediction using multiple seismicity indicators", International journal of neural systems 17 (1), (2007) : 13-33.

[21] Ikram, A., Qamar, U., "A rule-based expert system for earthquake prediction", Journal of Intelligent Information Systems 43 (2014) : 205-230.

[22] Moustra, M., Avraamides, M., Christodoulou, C., "Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals", Expert systems with applications 38 (12), (2011) : 15032-15039.

[23] Li, C., Liu, X., "An improved PSO-BP neural network and its application to earthquake prediction", In: Chinese Control and Decision Conference (CCDC) IEEE (2016) : 3434-3438.

[24] Narayanakumar, S., Raja, K., "A BP artificial neural network model for earthquake magnitude prediction in Himalayas, India", Circuits and Systems 7 (11) (2016) : 3456-3468.

[25] Kim, J., & Moon, N., "BiLSTM model based on multivariate time series data in multiple fields for forecasting trading area", Journal of Ambient Intelligence and Humanized Computing, (2019) : 1-10.

[26] Mignan, A., Broccardo, M., "Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations", Seismological Research Letters 91 (4) (2020) : 2330-2342.

[27] Mousavi, S. M., Zhu, W., Sheng, Y., Beroza, G. C., "CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection", Scientific reports 9 (1) (2019) : 10267.

[28] Kail, R., Burnaev, E., Zaytsev, A., "Recurrent convolutional neural networks help to predict location of earthquakes", IEEE Geoscience and Remote Sensing Letters 19 (2021) : 1-5.

[29] McHugh, C. M., Seeber, L., Cormier, M. H., Dutton, J., Cagatay, N., Polonia, A., ... Gorur, N., "Submarine earthquake geology along the North Anatolia Fault in the Marmara Sea, Turkey: a model for transform basin sedimentation", Earth and Planetary Science Letters 248 (3-4) (2006) : 661-684.

[30] Tan, O., Tapirdamaz, M. C., Yörük, A., "The earthquake catalogues for Turkey", Turkish Journal of Earth Sciences 17 (2) (2008) : 405-418.

[31] Reilinger, R. E., McClusky, S. C., Oral, M. B., King, R. W., Toksoz, M. N., Barka, A. A., ... & Sanli, I., "Global Positioning System measurements of present-day crustal movements in the Arabia-Africa-Eurasia plate collision zone", Journal of Geophysical

Research: Solid Earth 102 (B5) (1997) : 9983-9999.

[32] Sparavigna, A. C., "Software applied to archaeoastronomy: SunCalc and MoonCalc at the Torhouse Stone Circle", PHILICA (2017) : (1134).

[33] Yang, J., Liu, L., Jiang, T., Fan, Y., "A modified Gabor filter design method for fingerprint image enhancement", Pattern Recognition Letters 24 (12) (2003) : 1805-1817.

[34] Sztandera, L. M., "Tactile fabric comfort prediction using regression analysis", Wseas Transactions on Computers 2 (8) (2009) : 292-301.

[35] Masters, T., "Practical neural network recipes in C++", Morgan Kaufmann (1993).

[36] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J., "A novel connectionist system for unconstrained handwriting recognition", IEEE transactions on pattern analysis and machine intelligence 31 (5) (2008) : 855-868.

[37] Graves, A., Mohamed, A. R., Hinton, G., "Speech recognition with deep recurrent neural networks", In: IEEE international conference on acoustics, speech and signal processing IEEE (2013) : 6645-6649.

[38] Chauhan, S., Vig, L., "Anomaly detection in ECG time signals via deep long short-term memory networks", In: IEEE international conference on data science and advanced analytics (DSAA) IEEE (2015) : 1-7.

[39] Shewalkar, A., Nyavanandi, D., Ludwig, S. A., "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU", Journal of Artificial Intelligence and Soft Computing Research 9 (4) (2019) : 235-245.

[40] Yu, Y., Si, X., Hu, C., Zhang, J., "A review of recurrent neural networks: LSTM cells and network architectures", Neural computation 31 (7) 2019) : 1235-1270.

[41] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., "Empirical evaluation of gated recurrent neural networks on sequence modeling", arXiv preprint 1412.3555 (2014).