ACTA INFOLOGICA 2022;6(2):265-281
dergipark.org.tr/acin

İSTANBUL UNIVERSITY PRESS

DOI: 10.26650/acin.1146097

RESEARCH ARTICLE

Comparison of Different Heuristics Integrated with Neural Networks: A Case Study for Earthquake Damage Estimation

Yapay Sinir Ağı ile Entegre Farklı Sezgisel Yöntemlerin Karşılaştırılması: Deprem Hasar Tahmini için bir Vaka Çalışması

Ayşe Berika Varol Malkoçoğlu[1], Zeynep Orman[2], Rüya Şamlı[3]

[1](Lect.) Beykoz University, Vocational School of Logistics, School of Business, Istanbul, Turkiye
[2](Assoc. Prof.) Istanbul-Cerrahpasa University, Faculty of Engineering, Department of Computer Engineering, Division of Computer Sciences, Istanbul, Turkiye
[3](Prof.) Istanbul-Cerrahpasa University, Faculty of Engineering, Department of Computer Engineering, Division of Software Engineering, Istanbul, Turkiye

ORCID: A.B.V.M. 0000-0003-1856-9636; Z.O. 0000-0002-0205-4198; R.Ş. 0000-0002-8723-1228

Corresponding author:
Ayşe Berika VAROL MALKOÇOĞLU
Beykoz University, Vocational School of Logistics, School of Business, Istanbul, Turkiye
E-mail address:
ayseberikavarolmalkocoglu@beykoz.edu.tr

Submitted: 22.07.2022
Revision Requested: 25.11.2022
Last Revision Received: 28.12.2022
Accepted: 19.12.2022

Citation: Varol Malkocoglu, AB., Orman Z., & Samli R. (2022). Comparison of different heuristics integrated with neural networks: A case study for earthquake damage estimation. *Acta Infologica, 6*(2), 265-281. https://doi.org/10.26650/acin.1146097

ABSTRACT

Earthquakes are among the most challenging natural phenomena to predict. Most of these unpredictable earthquakes result in the loss of human lives and property. Seismologists can estimate the probable location and magnitude of such earthquakes. However, the actual time and extent of their impact remain unknown. If the effects of possible earthquakes can be predicted, quick and accurate decisions can be made. For this purpose, developing predictive models about earthquakes is a prevalent and vital issue in the literature. In this study, various Machine Learning (ML) algorithms were compared on a public dataset of earthquakes, which had occurred worldwide and had a local magnitude $Ml \geq 3$, and the algorithm with the highest performance was selected and optimized with various other algorithms. The performances of the models were compared using different performance evaluation metrics such as accuracy, Mean Square Error, Root-Mean Square Error, precision, recall, and f1 score. As a result, it was observed that the Artificial Neural Network (ANN) algorithm optimized with the Particle Swarm Optimization (PSO) algorithm produced the most successful result with an accuracy value of 0.82. Based on the obtained results, it is believed that this model can be used in different earthquake damage prediction studies and as a guide in emergency planning.

Keywords: Earthquake, Damage prediction, Machine learning, Optimization algorithms, Artificial neural networks, Particle swarm optimization

ÖZ

Depremler, tahmin edilmesi en zor doğa olayları arasında yer almaktadır. Bu öngörülemeyen deprem-lerin ardından çoğu zaman can ve mal kayıpları meydana gelmektedir. Depremler önceden kesin olarak belirlenemese bile deprem bilimciler tarafından olası konumları ve büyüklükleri yaklaşık olarak tahmin edilebilmektedir. Ancak, bu depremlerin zamanı ve bırakacağı etkinin boyutu bilinme-mektedir. Eğer olası depremlerin etkileri önceden tahmin edilebilirse, arama kurtarma çalışmaları sırasında ekiplerin hızlı ve doğru kararlar alması sağlanabilir ve bu sayede özellikle can kayıplarının önüne geçilebilir. Bu amaç doğrultusunda depremlerle ilgili tahmin modelleri geliştirmek günümüzde oldukça yaygın ve hayati bir konudur. Bu çalışmada ise dünya genelinde gerçekleşmiş yerel büyük-lüğü $Ml \geq 3$ olan açık kaynaklı deprem verileri kullanılarak farklı Makine Öğrenmesi algoritmaları karşılaştırılmış ve en yüksek performansa sahip olan algoritma seçilerek çeşitli algoritmalar ile opti-mize edilmiştir. Modellerin performansı doğruluk, Ortalama Kare Hata, Kök-Ortalama Kare Hata, kesinlik, geri çağırma ve f1 puanı gibi farklı performans değerlendirme metrikleri kullanılarak karşı-laştırılmıştır. Sonuç olarak PSO algoritması ile optimize edilmiş ANN algoritmasının 0.82 oranında doğruluk değeri ile en başarılı sonucu ürettiği gözlemlenmiştir. Elde edilen sonuçlara bakıldığında bu modelin farklı deprem hasar tahmin çalışmalarında ve acil durum planlamasında yol gösterici olarak kullanılabileceği düşünülmektedir.

Anahtar Kelimeler: Deprem, Hasar tahmini, Makine öğrenmesi, Optimizasyon algoritmaları, Yapay sinir ağları, Parçacık sürüsü optimizasyonu

1. INTRODUCTION

Earthquakes occur due to the fact that sudden vibrations, which emerge as a result of fractures in the earth's crust, spread in waves and shake the ground. This natural phenomenon, which is difficult to predict today, has caused many casualties and property loss over the centuries. Therefore, human beings have been trying to detect and predict earthquakes with the help of various signs since primitive times to take precautions when necessary. Thanks to the developing science and technology, scientists who base these predictions on mathematical and statistical methods have tried to make earthquake predictions by probability-related methods, especially by using the location, time, and magnitude parameters of previous earthquakes. In addition, damage caused by earthquakes has been examined similarly, with attempts to predict possible damage using artificial intelligence-based algorithms, especially in recent years. In order to develop earthquake damage prediction models by using artificial intelligence-based methods, data of the effects of previous earthquake in the designated region are used. This data usually consists of such information as the location of earthquakes, their severity, the number of people lost, the number of people who died, the number of buildings destroyed, and the value of material damage emerging in the location of the earthquake (NOAA, 2021). The earthquake intensity in the collected data is measured indirectly based on certain standards (Table 1). These values are calculated by examining the effects of earthquakes, with various magnitude values which identify the earthquake being obtained through multiple methods (Kandilli Observatory and Earthquake Research Institute, 2021).

Table 1

Earthquake magnitude scales and their tasks

Earthquake magnitude scales	**Symbol**	**Explanation**
Earthquake Duration Magnitude	Md	Measured by using the vibration time on the seismometer.
Local Magnitude	Ml	Measured by using the amplitude of the sound wave.
Surface Wave Magnitude	Ms	Measured by using wave amplitude spread from the epicenter to the environment.
Body Wave Magnitude	Mb	Measured from the early portion of the body wave train that is usually associated with the P-wave.
Momentum Magnitude	Mw	Calculated by performing the mathematical model of the earthquake.

In some studies (Epstein & Lomnitz, 1966; Bath, 1979; Moustra, Avraamides & Christodoulou, 2011; Reye, Morales-Esteban & Martínez-Álvarez, 2013), data were given to statistical or artificial intelligence-based models, and the magnitude of possible casualties in earthquakes was predicted. The magnitude of the damages and casualties vary depending on the area where the earthquake occurred and the structure of the existing buildings. By learning the data related to the properties and effects of earthquakes that occurred in various countries through the developed ANN-based model, this study aims to predict and prevent human casualties that may arise due to a possible earthquake in any country in the world. To this end, preprocessing processes were first applied to the considered dataset. Then, we compared the results of some traditional ML algorithms, such as Decision Tree (DT), Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and also ANN according to various metrics, to find the best Machine Learning (ML) method used on the preprocessed dataset. ANN has been shown to obtain the best accuracy result among the other methods. After that, various optimization algorithms, such as Gradient Descent (GD), Mini-Batch Stochastic Gradient Descent (MBSGD), RMSProp, Adaptive Moment Estimation (ADAM) optimizer, and heuristic algorithms, such as Genetic Algorithm (GA) and PSO, were applied to improve the performance of the ANN method. ANN optimized with PSO (PSO-ANN) achieved the best results based on various performance evaluation metrics, such as: accuracy, MSE, RMSE, precision, recall, and f1-score. The main contributions of the study can be highlighted as follows:

A detailed literature review on earthquake prediction is presented, and the well-known ML methods used in these studies were determined and evaluated.

As a result of the literature review, it has been determined that ANN and PSO are among the frequently used ML methods for earthquake prediction. However, it has also been observed that there are a limited number of studies using ANN and PSO for earthquake damage and casualty prediction in the literature. Therefore, the proposed PSO-ANN model in this study is novel for earthquake damage prediction.

The dataset utilized in the model training of this study has not been encountered in earthquake damage prediction models in the literature.

Various ML models and different optimization and heuristic algorithms have been applied. The results show that the developed PSO-ANN model outperforms the established counterparts based on various performance evaluation metrics.

The developed PSO-ANN model was determined as an appropriate model to predict earthquake damages, according to its high accuracy.

2. LITERATURE REVIEW

Various studies in the literature use evolutionary algorithms within the scope of earthquake studies. These studies, which estimate possible human casualties after an earthquake, are summarized in Table 2 according to: the datasets used, the models used for the prediction, the metrics used to evaluate the model, and the model performance evaluations.

Table 2

Models used in earthquake prediction studies and performance evaluation of these models

Studies	Dataset	Target	Models	Metrics	Model Performance Evaluation
Aghamohammadi et al., 2013	Bam in 2003	Damage assessment	BPNN	RMSE	Dead predicted-RMSE:0.021 Injured predicted-RMSE:0.042
Xing et. al., 2015	China in 1970-2015	Damage assessment	RW v-SVM, SVM, BPNN	MSE	Dead predicted-MSE:0.0412 Injured predicted-MSE:0.0211
Cui et. al., 2021	China in 1966-2017	Damage assessment	GBDT XGBoost	MAE MSE	GBDT-MAE:0.441, MSE:0.343 XGBoost- MAE:0.445, MSE:0.346
Xia Wang et al., 2011	China in 1990-1995	Damage assessment	BPNN	-	Close results were produced
Gul and Guneri, 2016	(Mw⩾5) earthquakes in Turkey in 1975-2016	Damage assessment	LM- ANN	R^2	Able to correctly predict the number of survivors
Turkan and Ozel, 2014	(Ms⩾5) earthquakes in Turkey in 1900-2012	Damage assessment	LR, BR, SAR, SBR	MSPE R^2	LR-MSPE: 3.30700, R^2:0.53 BR-MSPE: 0.00052, R^2:0.65 SAR-MSPE: 0.00044, R^2:0.91 SBR-MSPE: 0.00041, R^2:0.61
Asim et al., 2018	Pakistan in 1980-2016	Earthquake prediction	GP-Adaboost	Accuracy	Hindikush 87%, Chile 84.5%, Pannakat 86%
Tao, 2015	(Mw⩾6.5) Himalaya and Nepal	Earthquake prediction	BPNN, BPNN-GA	MSE	BPNN-Nepal: 0.010 Himalaya: 0.032
Saba et al., 2017	Pakistan in 2002-2012	Earthquake prediction	BA-ANN, BPNN	MSE	BA-ANN-Azad 0.0091, Balochistan 0.015, Hindikush 0.027
Li and Liu, 2016	Coastal areas	Earthquake prediction	BPNN, PSO-BPNN	MAE	PSO-BPNN:0.031
Abraham and Rohini, 2019	Japan in 2010-2016	Earthquake prediction	PSO-BPNN	MSE	PSO-BPNN model is more successful than a simple BPNN model.
Xi et al., 2019	Ludian region of China	Earthquake prediction	ANN, PSO-ANN	Accuracy	ANN-76.5%, PSO-ANN-82.5%
Moayedi et al., 2019	Laleh valley in western Iran	Earthquake prediction	ANN, PSO-ANN	R^2 RMSE	ANN- RMSE: 0.111, R^2: 0.9733 PSO-ANN-RMSE: 0.104, R^2 : 0.9717
Gordan et al., 2016	699 FOS data	Earthquake prediction	ANN, PSO-ANN	R^2 RMSE	ANN-RMSE: 0.057, R^2: 0.915 PSO-ANN-RMSE: 0.022, R^2 :0.986
Shiuly et al., 2020	Himalayan region of India	Earthquake prediction	ANN, GA	-	Correlation coefficient of GA is lower
Jena and Pradhan, 2020	Aceh, Indonesia	Earthquake risk assessment	AHP-TOPSIS	-	Showed that 10,252 and 44,443 people belonged to very high and high-risk zones
Alizadeh et al., 2018a	Tabriz City, Iran	Earthquake hazard assessment	ANN	Pearson Correlation	Developed a novel computational framework
Alizadeh et al., 2018b	Tabriz City, Iran	Earthquake vulnerability assessment	ANP-ANN	Pearson Correlation	A new ANP-ANN model was established
Ahmad et al., 2014	Pakistan	Earthquake loss estimation	Probabilistic framework	-	Two methods for structures assessment are found comparable
Ahmad et al., 2012	Pakistan	Seismic vulnerability	A new model	-	The aim of the study was to understand the damage mechanism of the model.
Ahmad, 2019	-	Fragility Functions	Probabilistic framework	-	Seismic fragility functions were derived.
Yuan, 2021	Global earthquake data	Earthquake magnitude prediction	K-means	PPV, NPV, Sn, Sp, Avg	A seismic prediction model using clustering of global earthquake data is presented.
Shan et al., 2020	Qiabuqia Geothermal Field, China	Earthquake risk assessment	ANN	-	The regional tectonic evolution based on the survey data

PPV is a predictive positive value, NPV is a negative predictive value, Sn is sensitivity, Sp is specificity, and Avg is average, BPNN is Backpropagation Neural Network

To show the general beneficial usage of ANN in natural disaster prediction, the recent literature is given in below and summarized in Table 3.

Table 3

Models used in natural disaster prediction studies and performance evaluation of these models

Studies	Dataset	Target	Models	Metrics	Model Performance Evaluation
Gessang and Lasminto, 2020	Jenelata Sub-watershed	flood mitigation	ANN	accuracy, RMSE, Corr. coeff.	accuracy:71.19 % RMSE:1.45, Corr. coeff.:0.6
Dhunny et al., 2020	Mauritius	flood prediction	ANN	accuracy	High-level accuracy in flood prediction.
Sahoo et al., 2021	Barak River	flood prediction	ANN, RBFNN, SVM, FA,	R^2, MSE, RMSE	SVM MSE: 0.00792, RMSE: 0.03064, R^2:0.9818 RBF-FFA MSE: 0.00776, RMSE: 0.03078, R^2:0.9712 FFBPN MSE: 0.00698, RMSE: 0.03311, R^2:0.8821
Rani et al., 2020	Karnataka and Maharashtra	flood monitoring	Linear Regression, ANN, SVM	MAE	Linear Regression MAE:40.2467874 ANN MAE: 90.606787 SVM MAE: 21.8097545
Obasi et al., 2020	Anambra-Imo River	river discharge forecasting	ANN	R^2	Average 0.95
Ranit and Durge, 2019	Wardha river	flood prediction	ANN	-	By using the forecasted inflow, rate of inflow in reservoir can decide the time of operation
Bano et al., 2021	Upper Yamuna Basin	flood prediction	ANN	R^2, SSE, MSE, RMSE	Showed that model has less SSE, MSE and RMSE.
Hadid et al., 2020	north of France	flood prediction	LSTM and piecewise func.	-	Usage of PWARX systems in the flood forecast field.
Boutkhamouine et al., 2020	Salat river	flood prediction	Bayesian networks	-	The model showed good performances.
Dazzi et al., 2021	Parma River	flood prediction	SVR, MLP, LSTM	RMSE, NSE	RMSE < 15 and NSE > 0.99
Zhou et al., 2020	Yangtze River	flood prediction	Kalman Filter with RNN	-	Hybridizes Kalman Filter with RNN.
Zhan et al., 2020	Yangtze River	flood prediction	VBNN	-	VBNN obtained more accurate forecast results
Anupam and Pani, 2020	Brahmani river	flood prediction	ELM-PSO	R^2, MSE	Considerable accuracy in terms of R^2 and MSE.
Chawla and Singh, 2021	North-Western Himalaya	avalanche forecasting	Random forest	-	Random Forest technique for avalanche forecasting.
Kaur et al., 2020	North-Western Himalaya	avalanche forecasting	HMM, NN, ANN	-	Different models have been developed with same input data.
Joshi et al., 2020	North-Western Himalaya	avalanche forecasting	ANN	RMSE, standard deviation	RMSE of all parameters has been found.
Choubin et al., 2020	Taleghan watershed	avalanche forecasting	GAM, MARS, BRT, SVM	Accuracy Kappa PrecisionRecall AUC	Accuracy > 0.88, Kappa > 0.76, Precision > 0.84, Recall > 0.86, AUC > 0.89
Adjei et al., 2021	western region of Ghana	rainfall forecasting	LSTM	MSE, RMSE	Precipitation with parameters affect rainfall forecast efficiency of the LSTM model.

SVR is Support Vector Regression, MLP is Multi-Layer Perceptron, RNN is Recurrent Neural Networks, VBNN is Variational Bayesian Neural Network, ELM-PSO is Extreme Learning Machine-Particle Swarm Optimization, HMM is Hidden Markov Model, AUC is area under ROC curve, and LSTM is Long Short-Term Memory.

When these studies are evaluated, it is seen that most of the studies conducted on earthquakes relate to predicting them. The ANN structure, bio-inspired algorithms, such as PSO and GA, and their combinations with hybrid models are widely used in such studies, and high-performance results are obtained according to performance evaluation metrics. On the other hand, it is also observed that the use of ANN and other bio-inspired algorithms in earthquake damage prediction studies are limited in number. Most damage prediction studies rely on traditional ML methods. Various datasets with different features and data types are used in these papers.

In this paper, the prediction of human casualties that might occur in earthquakes was performed using the PSO-ANN structure, an evolutionary-based ANN model. For this purpose, an open-source dataset was used which was obtained from the National Environmental Information Center (NOAA) (2021). This dataset contains such information as: latitude, longitude, Ml size, depth, loss of life, and property of earthquakes occurring worldwide. The results obtained using different parameters were evaluated and compared to the results obtained with other methods based on: accuracy, MSE, RMSE, precision, recall, and f1-score metrics.

The dataset utilized in the model training of this study has not been encountered in the earthquake damage prediction models recommended in the literature. As a result, the dataset and the proposed model used in this earthquake damage prediction study are unique and will contribute to both the literature and the field of practical application.

3. MATERIALS and METHODS

3.1 Dataset and Preprocessing

The dataset used in the study was taken from NOAA (2021) database. In this study, all earthquakes that occurred in the world and caused deaths are included. In the dataset, there are 2,317 pieces of data and 48 features (date, time, location, depth of focus, longitude, latitude magnitude, intensity, damage predictions such as total deaths, injuries, damaged houses, and destroyed houses). However, although a large number of features are described in this dataset, it remains unbalanced because it does not have equal or close numbers of data from each class. In addition, much of the data has no value above 50%. Therefore, such data was removed from the dataset for the study, leaving 14 features available for inspection. These features are shown in Table 4.

Table 4
Features and description of the dataset used in the study

Data Feature	Explanation	Data Type	Empty value (%)
Year	The year the earthquake occurred	Numeric	0.0
Country	The country of the earthquake	Text/Nominal	0.0
Region	Codes of the regions	Nominal	0.0
Location Name	The city of the earthquake	Text/Nominal	0.0
Latitude	Line segments dividing the earth into cross-sections for positioning	Numeric	0.302
Longitude	Line segments dividing the earth into longitudinal sections for positioning	Numeric	0.302
Focal Depth (km)	Depth of the earthquake as km	Numeric	44.15
Mag (Ml)	Local magnitude of the earthquake	Numeric	19.03
Mag (Ms)	Earthquake surface wave size	Numeric	41.21
MMI Int	Modified Mercalli intensity scale	Numeric	43.33
Deaths	The number of casualties	Numeric	17.09
Damage Description	Damage size	Nominal	0.0
Houses Destroyed Description	Destroyed house size	Nominal	49.58
Deaths Description	Category of casualties	Nominal	0.0

Missing data in some of these features used in the dataset may adversely affect the learning ability of the model. In order to avoid this situation, missing data in this study were completed using the interpolation method. It is known that when the model is trained by selecting the useful features in the dataset, a decrease in training time, an increase in interpretation skills, and an increase in performance can be achieved by preventing overfitting. Therefore, feature selection was performed after the deficiencies in the dataset were completed. In the study, the correlation method, which is a statistical technique used to evaluate the relationship between each input variable and the target variable, was chosen. Thus, the learning speed and performance of the model were increased by eliminating unnecessary features from the dataset.

The correlation relationship between the features can be seen in the temperature map given in Figure1 below. When the map was examined, it was observed that there were high correlations between some features. The threshold value was determined as T=0.45 and one of the features with a value above this threshold value was removed from the dataset. Accordingly, the features that have higher correlations than the threshold value are listed below:

(1) Damage Description and Death Description

(2) Longitude and Region

(3) Mag (Ms) and Mag (Ml)

(4) Location Name and Country

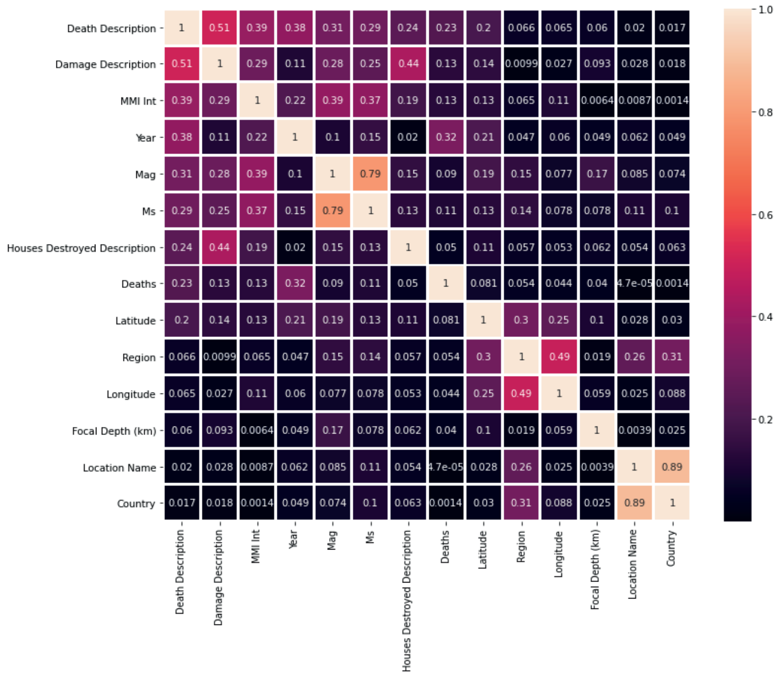

Figure 1. Correlation map of properties

It was necessary to remove a feature in each group from the dataset. Then, data with text/nominal data type was converted by the algorithm to numeric data in order to produce a more significant result. This process was carried out by the label-encoder method, which converted each value into a number. The properties in the new dataset obtained are shown in Table 5.

Table 5

The latest version of the dataset

Data Feature	Explanation	Data Type	Variable Type	Empty value (%)
Year	The year the earthquake occurred	Numeric	Dependent	0.0
Country	The country of the earthquake	Numeric	Dependent	0.0
Latitude	Line segments dividing the earth into cross-sections for positioning	Numeric	Dependent	0.0
Longitude	Line segments dividing the earth into longitudinal sections for positioning	Numeric	Dependent	0.0
Focal Depth (km)	Depth of the earthquake in km	Numeric	Dependent	0.0
Mag (Ml)	Local magnitude of the earthquake	Numeric	Dependent	0.0
MMI Int	Modified Mercalli intensity scale	Numeric	Dependent	0.0
Deaths	The number of casualties	Numeric	Dependent	0.0
Damage Description	Damage size	Nominal	Dependent	0.0
Houses Destroyed Description	Destroyed house size	Nominal	Dependent	0.0
Deaths Description (output)	1: ~ 1 -50 people 2: ~51-100 people 3: ~101-1000 people 4: ~1001 or more people	Nominal	Independent	0.0

3.2 Machine Learning Models

3.2.1. Decision Tree

Decition Tree (DT) is one of the most widely used methods for supervised learning. This method can handle both categorical and numerical data, whereas other techniques are specialized for only one type of variable. DTs used in data mining are mainly of two types: classification tree and regression tree. There are various DT algorithms in use today, such as: ID3, C4.5, CART, CHAID, and MARS (Rathee and Mathur, 2013; Hssina, Merbouha, Ezzikouri & Erritali, 2014).

3.2.2 Naive Bayes

Bayes' theorem is of fundamental importance for inferential statistics and many advanced ML. Bayesian reasoning is a logical approach to updating the probability of hypotheses in the light of new evidence, and it has a very important place in science (Berry, 1996).

A Neural Network (NN) consists of an input layer, hidden layers, and an output layer. In particular, given the input data $X=\{x_1,\dots,x_N\}$ and output data $Y=\{y_1,\dots,y_N\}$ with N data points, the input and output data can be modeled with the parameters ω as $Y=NN(X,\omega)$ where ω can be trained by backpropagation. Then the model output value y^* can be forecast by giving a new input point x^* through the network $y^*=NN(x^*,\omega)$. As for Bayesian Neural Networks (BNN), the values of the parameter ω are initialized following a prior distribution $p(\omega)$. Then the output and input training dataset, X,Y is used to obtain the optimal posterior distribution $p(\omega|X,Y)$ of the BNN model parameters.

3.2.3 Support Vector Machine

Support Vector Machine (SVM) is a binomial classification algorithm that builds computational classification models that assign samples into two or more classes, which can be applied to prediction or diagnosis. SVM is fundamental because of theoretical reasoning; it is robust to a large number of variables and small samples, can learn both simple and high complex classification models, avoids overfitting by using complex mathematical principles, and provides reliable results (Hardin, Duviella & Lecoeuche, 2011).

3.2.4 Logistic Regression

Logical Regression (LR) is a mathematical modeling approach that can be used to describe the relationship of several inputs to a dichotomous dependent variable. While other modeling approaches are also possible, LR is by far the most popular modeling procedure used to analyze, for example, epidemiologic data (Kleinbaum and Klein, 2010).

3.2.5 Artificial Neural Network

Artificial Neural Networks (ANN) are structures designed by the inspiration of the learning and remembering abilities of biological neurons in the human brain that imitates the synaptic connection between biological neuron cells and these cells. This structure learns by using existing examples. Based on this learning, these models can respond to reactions from the environment. Instead of storing the information in memory the way classical computers do, this model has a distributed structure that spreads the information it obtains to the whole network with weights.

Traditional neural networks are basically divided into two: single-layer and multi-layer perceptrons. Structures that produce output by passing the input parameters through the activation function are known as a single-layer perceptron, while structures that feed the input parameters to the hidden layers, transfer them from the hidden layers to other hidden layers, and then produce the output value are known as multi-layer perceptrons.

The multi-layer ANN algorithm, which emerged for the first time in the 1960s, became popular with an article published by Rumelhart, Hinton, and Williams (1986). This multi-layer perceptron consists of input, output, and hidden layers. Each hidden layer consists of numerous perceptron's, which are called hidden layers. This structure is divided into two, as forward and backward propagation. Thanks to the forward and backward propagation methods of multi-layer perceptrons (Fig. 2), the network performs the classification process by learning from the labeled data.

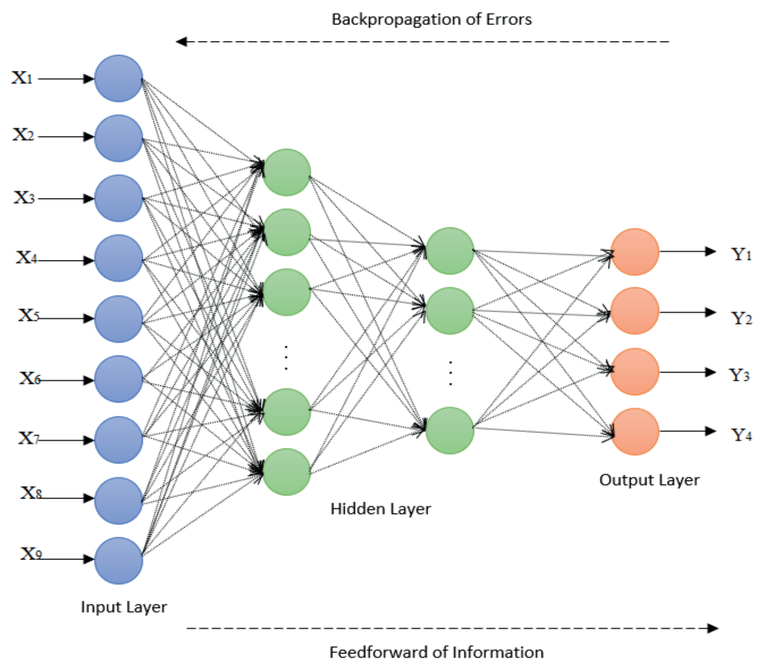

Figure 2. Forward and Backward propagation multi-layer neural network algorithm

The feedforward neural network (FFNN) is represented by the input layers X and the k^{th} neuron in the layers which is shown in *eq*.(1) and *eq*.(2). i^{th} step, k^{th} neuron output is represented by the Y output value. Each neuron in the intermediate layers receives the information from all neurons in the input layer with the effect of the connection weights (W). (j: weight value linking to middleware element)

$$Y_k^i = X_k \tag{1}$$

$$NET_j^a = \sum_{k=1}^{n} W_{kj} Y_k^i \tag{2}$$

Any f activation function whose derivative can be taken is used, which is shown in eq. (3).

$$Y_j^a = f(NET) \tag{3}$$

Using BPNN, the error is shared among the weights (Fig. 2), and thus the learning of the network is strengthened, which is shown in *eq*.(4). The error value of the network is obtained by taking the difference between the expected value T_m and the output value Y_m. The error for the m^{th} neuron is Em.

$$E_m = T_m - Y_m \tag{4}$$

Calculation of the difference between the expected value and the actual output after training the network according to the input-output data underlies the FFNN algorithm. The error is reduced by sharing the calculated error value proportionally to the neuron weights. This method can produce good results on linear and nonlinear problems (Goodfellow, 2016). In this study, an FFNN with two hidden layers and one output layer was used in the proposed model for earthquake damage estimation.

3.3 Optimization Algorithms for ANN

3.3.1 Gradient Descent

Gradient Descent (GD) is an iterative algorithm whose purpose is to make changes to a set of parameters to reach an optimal set of parameters that leads to the lowest loss function value possible. A loss, cost, or objective function is the function whose value we seek to minimize. The form of the loss function looks as eq. (5):

$$lossfunction = f(w) \tag{5}$$

When performing GD, each time we update the parameters, we expect to observe a change in *minf*(*w*). At each iteration, the gradient of the function that contains parameters in w is taken so that changes in the function with respect to parameters bring us closer to the goal of reaching an optimal set of parameters that will ultimately lead to the lowest possible loss function value.

3.3.2 Mini-Batch Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a variation of GD that randomly samples one training sample from the dataset to be used to compute the gradient per iteration. SGD works well because we are using just one data point to calculate the gradient, update the weight vector *w*, and compute the loss function value. Sampling more than one sample to compute the gradient for SGD such that 1 < b < n is referred to as Mini-Batch Stochastic Gradient Descent (MBSGD) (Botton, 2010)

3.3.3 RMSProp

RMSProp is a variant of the Gradient Descent Algorithm. It is an unpublished, adaptive learning rate method proposed by Geoff Hinton (Mcmahan and Streeter, 2014).

3.3.4 Adaptive Moment Estimation

ADAM optimizer is a method that computes adaptive learning rates for each parameter. ADAM stores an exponentially decaying average of past squared gradients v_t like Adadelta and RMSprop. It also keeps an exponentially decaying average of past gradients m_t, similar to momentum eq. (6). g_t represent gradients at timestep t. β represent exponential decay rates for the moment estimates.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{6}$$

m_t and v_t are estimates of the first moment and the second moment e) of the gradients, respectively, hence the name of the method (Kingma and Ba, 2015).

3.4. Heuristic Algorithms for ANN

3.4.1 Genetic Algorithm

Similar to other evolutionary algorithms, the main operators of the Genetic Algorithm (GA) are selection, crossover, and mutation. Every solution corresponds to a chromosome, and each parameter represents a gene. GA evaluates the fitness of each individual in the population using a fitness function. In order to improve weak solutions, the most suitable individuals are selected and their genes are passed on to the next generation. This operator is more likely to select the good solution since the probability is proportional to the fitness value. What increases local optima avoidance is the probability of selecting poor solutions. This means that if good solutions are trapped in a local solution, they can be pulled out with other solutions. Because GA is stochastic, it is understandable to question its efficiency and reliability. What makes this algorithm reliable and able to estimate the global optimum for a given problem is the process of maintaining the good solution in each generation and using them to improve other solutions (Mirjalili, 2019).

Five phases are considered in a genetic algorithm.

Initial population: A set of individuals, called a population, is characterized by a set of parameters (variables) known as genes. Genes are combined into a string to form a Chromosome (solution).

Fitness function: The fitness function determines an individual’s ability to compete with other individuals. By giving each individual a fitness score, the probability of the individual being selected for reproduction is determined.

Selection: It selects the two most suitable individuals (parents) and transfers its genes to the next generation. Individuals with high fitness have a greater chance of being selected for breeding.

Crossover: For each parent pair to be bred, a crossover point is chosen randomly from among the genes. Offspring are created by exchanging the parents' genes.

Mutation: In order to preserve diversity within the population and prevent premature convergence, some genes are sometimes mutated in new offspring with a low probability.

3.4.2 Particle Swarm Optimization

Particle Swam Optimization (PSO) is an heuristic algorithm which was designed by Eberhart and Kennedy in 1995 and based on inspiration from the behaviors of bird swarms. In other words, this algorithm is a population-based heuristic algorithm developed based on the ability of swarms of animals, such as fish, birds, and insects, to find food sources and survive.

The particles (p_i) represent the animals in the swarm, and each particle adjusts its position to the best position in the swarm by using its previous experience. Particles move at certain velocities (v_i) in each iteration and update their velocities and positions based on the information from the previous step. At each time step t in the simulation, the velocity of the i th particle is represented as v_i. The update process keeps their best positions in each step-in memory and adjusts their other movements according to this position. In this case, the best position (p_{best}) of the particles is found for each iteration. The best position of the particle is calculated with the *argmin* objective function, which gives the minimum which is shown in eq. (8).

$$p_i = (p_{i1}, p_{i2}, \dots, p_n) i = 1,2,3, \dots N \tag{7}$$

$$p_{best}(t) = arg_t min f\left(p_i(t)\right) \tag{8}$$

The particle that has the best position in the swarm is followed by other particles. Therefore, among the (*pbest*)s, the minimum value is calculated using the *argmin* function, and the global best position (*gbest*) is obtained, which is shown in eq. (9).

$$g_{best}(\mathrm{t}) = arg_i min f\left(p_i(t+1)\right) \tag{9}$$

Thus, in each iteration, *pbest* and *gbest* can be obtained, and the particle's position and velocity can be updated. This process continues until the goal is reached, which is shown in eq. (10) and (11). c_1, c_2 represent learning constants; R_1, R_2 represent randomly generated 0~1 random number; and *xij* represent current position of particle.

$$v_{ij}(t+1) = v_{ij}(t) + c_1 R_1 \left(p_{best_{ij}}(t) - x_{ij}(t)\right) + c_2 R_2 \left(g_{best_{ij}}(t) - x_{ij}(t)\right) \tag{10}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

$$i = 1,2,3, \dots N \qquad j = 1,2,3, \dots n \tag{11}$$

i=1,2 ,3 , …N j=1,2 ,3 , …n

3.5. Performance Evaluation Metrics

Accuracy: It is the ratio of correct predictions to the total number of predictions, and it shows how well the model performs. Accuracy is defined by the following formula in eq. (12) (TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives):

$$(TP + TN)/(TP + TN + FP + FN) \tag{12}$$

MSE: It measures the average of the squares of the errors. It is defined as follows in eq. (13):

$$\frac{1}{n} \sum_{t=1}^{n} e_t^2 \tag{13}$$

RMSE: It shows the error distribution from a broad perspective. It is the square root of MSE and is defined by the following formula in eq. (14) (*e:* the error between the actual and predicted values, *n:* the number of observations)

$$\sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2} \quad (14)$$

Precision: It shows how close the model's predictions are to the observed values, and it is the ratio of correct positive predictions to the total number of positive samples. It is calculated as follows in eq. (15):

$$(TP)/(TP + FP) \quad (15)$$

Recall: It quantifies the number of correct positive predictions made out of all positive predictions, and it is defined as follows in eq. (16):

$$(TP)/(TP + FN) \quad (16)$$

$f1$-*score*: It combines both precision and recall into a single measure that captures both properties, and it is calculated by the following formula in eq. (17):

$$2 * ((Precision * Recall)/(Precision + Recall)) \quad (17)$$

4. RESULTS and DISCUSSION

In this study, a preprocessed open-source earthquake dataset which had been commonly used for earthquake prediction studies in the literature was used for the purpose of earthquake damage prediction. Firstly, DT, NB, SVM, LR, and the ANN algorithm were applied to the preprocessed dataset to determine the best ML method for the study. All these simulations were carried out in the Google Colaboratory environment by using the Python language. For these models, such libraries as sklearn, pandas, numpy, and matplotlib were used. For each models, the initial parameters were as follows:

DT : (criterion='entropy', splitter='random')

NB : (var_smoothing=1e-9)

SVM: (kernel='poly', C=0.01)

LR : (penalty='l2', tol=1e-4, C=1.0)

ANN : (activation='tanh',hidden_layer_sizes=(8, 8), solver='adam')

Table 6 presents the Accuracy, MSE, RMSE, precision, recall, and f1-score results of the conducted experiments for different ML models for both Ml ≥ 3 and Ml ≥ 5.

Table 6

Experimental results for different ML models

Algorithms	Accuracy		MSE		RMSE		Precision		Recall		F1-Score	
	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5
DT	0.63	0.62	0.95	0.95	0.975	0.97	0.63	0.62	0.62	0.62	0.63	0.62
NB	0.60	0.59	1.44	1.54	1.20	1.24	0.57	0.53	0.60	0.59	0.52	0.50
SVM	0.65	0.63	1.04	1.14	1.02	1.07	0.58	0.56	0.65	0.63	0.60	0.57
LR	0.56	0.55	1.40	1.48	1.18	1.21	0.48	0.45	0.57	0.55	0.47	0.45
ANN	**0.67**	**0.67**	**0.95**	**0.82**	**0.96**	**0.90**	**0.54**	**0.60**	**0.67**	**0.67**	**0.59**	**0.60**

Based on Table 6, it is evident that ANN obtained the best performance results among the other methods. The ANN produces an average of 0.67 accuracy for Ml ≥ 3 and 0.67 for Ml ≥ 5 as the best values. Optimization and heuristic algorithms can modify the attributes of an ANN, such as weights and learning rate, in order to reduce the losses and improve the performance of the model. For this purpose, we applied various optimization and heuristic algorithms to the ANN model, such as: GD, MBSGD, RMSProp, ADAM optimizer, GA, and PSO. Table 7 presents the accuracy, MSE, RMSE, precision, recall, and f1-score results of the experiments conducted on the ANN model with different optimizers for both Ml ≥ 3 and Ml ≥ 5. According to Table 7, ANN optimized with PSO achieved the best results based on evaluation metrics.

Table 7

Experimental results for different optimization and heuristic algorithms for ANN

Optim. & Heur. Algorithms +ANN	Accuracy		MSE		RMSE		Precision		Recall		F1-Score	
	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5	ML≥3	ML≥5
GD	0.41	0.40	0.19	0.18	0.42	0.42	0.54	0.52	0.41	0.39	0.41	0.42
MBSGD	0.54	0.53	0.16	0.17	0.39	0.40	0.59	0.57	0.54	0.53	0.52	0.51
RMSPprop	0.74	0.72	0.12	0.12	0.32	0.33	0.72	0.71	0.74	0.72	0.72	0.71
Adam	0.75	0.74	0.11	0.11	0.30	0.31	0.73	0.72	0.75	0.74	0.74	0.73
GA	0.76	0.72	0.48	0.88	0.69	0.75	0.84	0.88	0.76	0.73	0.76	0.79
PSO	**0.79**	**0.79**	**0.09**	**0.09**	**0.30**	**0.31**	**0.82**	**0.79**	**0.79**	**0.79**	**0.80**	**0.78**

4.1. PSO-ANN Model

When the results in Table 6 and Table 7 are examined, it is seen that the PSO-ANN approach produces the best results. For this reason, an earthquake damage estimation model was developed using the PSO-ANN approach in the study. Thanks to the PSO with high-speed convergence ability, the weights of the ANN model were optimized, and the performance was increased. The selection of particles and other initial parameters were determined randomly. The initial values of the particles in this study are given in Table 8.

Table 8

Initial values of the PSO algorithm

Parameter	Value
Number of particles (p)	30
Number of iteration (N)	100
pbest	2.05
gbest	2.05
Velocity (v)	*Rand* (*p, len*(*data*))
c_1,c_2	0.72

len(data) represents the number of data. *Rand* is used to randomly select the initial speed.

Using these values, the model is trained, tested, and the error between the predicted/actual values is calculated. The error is reduced by changing the positions of the particle at each iteration. This process continues until the MSE value of 1e-6 or the determined epoch value is reached. In this study, the weights of the ANN model used were optimized with PSO, and the estimation of human casualties that can be experienced in possible earthquakes was carried out.

Ten features of earthquakes were given as inputs to the developed model, which has two hidden layers and one output layer. The output layer consists of four classes, with the sigmoid function being used as the activation function. The network was trained by presenting the training data to the PSO-ANN (Figure 3) with random initial and weight values. By checking the convergence of the trained network, the error value (*Em*) between the expected value (*Bm*) and the predicted value (*Ym*) is calculated. The values of *Pbest* and *gbest* are used in order to update the positions of the particles to the best solution. These calculations and updates continue until the epoch count is completed or an MSE value of 1e-6 is obtained.

The *k-fold* cross-validation was applied to the dataset used in the model. In *k-fold* cross-validation, data is divided into *k* different subsets. *k*-1 subsets are used to train the data and to leave the last subset as the testing data. The average error value obtained as a result of *k* experiments indicates the validity of the model. The *k* value is usually chosen as 3 or 5. In our study, the training and the testing data were obtained by dividing the existing dataset into $k = 5$ layers. By dividing the data into layers, it was ensured that each layer was used as a testing set at one point. Table 6 and Table 7 show the average of the results produced in k layers. In this way, the performance of the model could be validated more accurately.

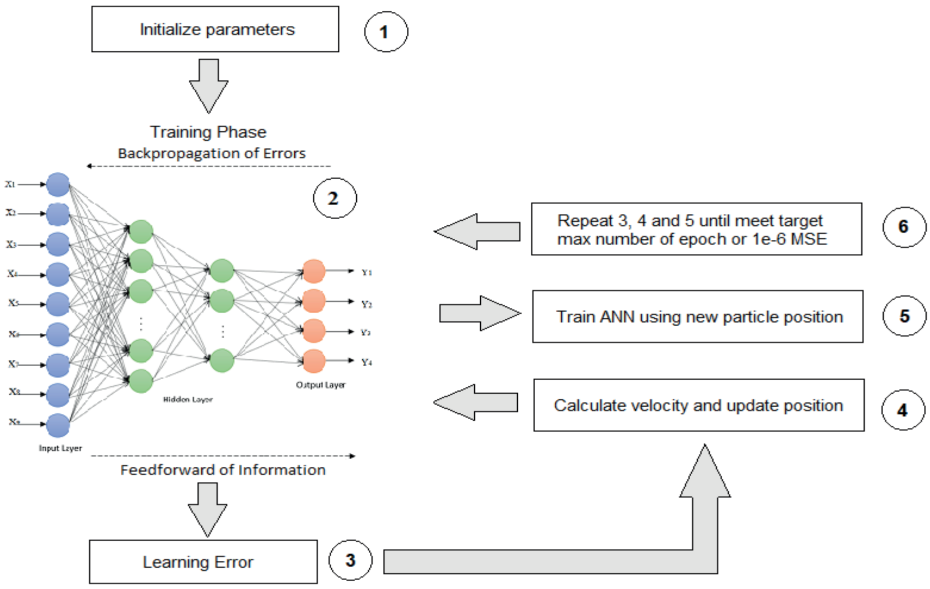

Figure 3. PSO-ANN representation model adapted from (Xi et. al.,2019; Moayedi et al.2019)

The proposed PSO-ANN model was trained with the preprocessed dataset of earthquakes, and the change in the number of neurons in its layers and the effect of the change in the epoch number on the model was observed. It is aimed to obtain the best results by changing the number of neurons and epoch values used in the first and second hidden layers of the experiments. The properties and results of the experiments carried out in this context are shown in Table 9. Based on the table, the dataset was divided into five layers, and each time one layer was used for testing, and the remaining four layers were used for training. In this way, each data in the dataset was used as both training and test data. Each *k* value represents the result of a test set. Accordingly, it was seen that the best average accuracy (acc = 0.82) was achieved at 500 epochs by using eight neurons in both hidden layers.

Table 9

Every kresult and average obtained from the PSO-ANN model

Number of neurons in the 1st hidden layer	Number of neurons in the 2nd hidden layer	Epoch	Accuracy					
			K1	K2	K3	K4	K5	K Avg.
8	8	300	0.54	0.74	0.86	0.87	0.93	0.79
8	8	400	0.58	0.76	0.81	0.88	0.92	0.79
8	**8**	**500**	**0.56**	**0.73**	**0.86**	**0.97**	**0.97**	**0.82**
16	8	300	0.54	0.61	0.77	0.93	0.96	0.76
16	8	400	0.52	0.72	0.78	0.94	0.94	0.78
16	8	500	0.54	0.76	0.83	0.92	0.96	0.80

In addition to the obtained accuracy, MSE, RMSE, Precision, Recall, and f1-score values were also calculated by taking the average of each k value. These values are included in Table 10.

Table 10

k- averages of results from the PSO-ANN model

Number of neurons in the 1st hidden layer	Number of neurons in the 2nd hidden layer	Epoch	MSE	RMSE	Precision	Recall	F1-Score
8	8	300	0.10	0.29	0.76	0.79	0.76
8	8	400	0.09	0.29	0.79	0.79	0.77
8	**8**	**500**	**0.08**	**0.26**	**0.82**	**0.82**	**0.80**
16	8	300	0.10	0.29	0.78	0.76	0.76
16	8	400	0.09	0.28	0.80	0.78	0.77
16	8	500	0.09	0.28	0.83	0.80	0.80

It was observed that the increase in the epoch number did not have a positive effect on the performance of the model. Although the change of neurons in the layers did not make a big difference, when accuracy and MSE were examined, it was seen that this model produced more successful results with hidden layers, which had eight neurons, and 500 epochs. When the results produced with these parameters were examined in terms of other performance evaluation metrics, it was observed that a value of 0.82 was obtained with the precision metric, and the ability to identify the correct samples for each class was high. Similarly, it was also observed that the process of finding all the correct examples per class was successful, with a value of 0.82 in the recall metric.

In addition to these metrics, the Receiver Operating Characteristic (ROC) curve produced by the most successful result in each k layer and the Area Under the Curve (AUC) value of each class were calculated. ROC is a probability curve and AUC represent area under the curve. The AUC show degree or measure of separability and specifies how much the model is capable of distinguishing between classes. The higher the AUC, the better the models ability to distinguish between classes (Hoo, Candlish & Teare, 2017). Figure 4 shows the ROC curves and AUC values of the model trained with 500 epochs and two hidden layers that have eight neurons. When these values were examined, it was determined that the model had a high ability to distinguish between classes.

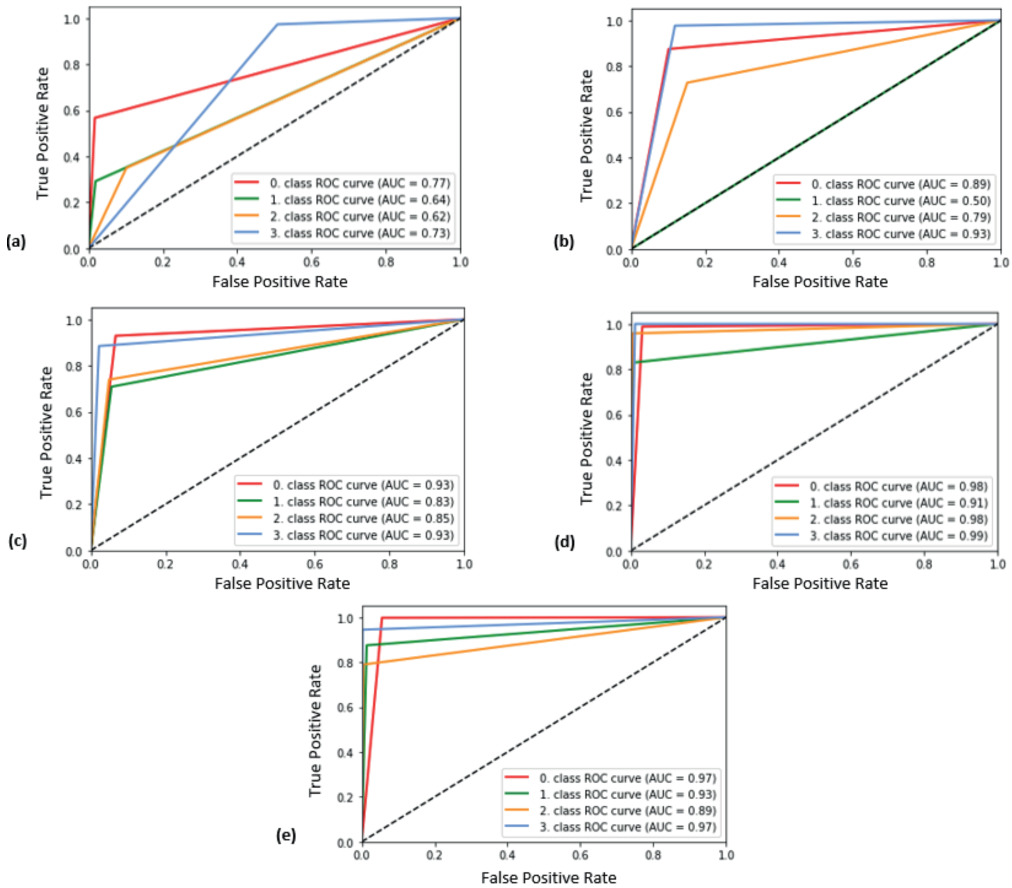

Figure 4. The ROC curve that the parameters produce the best result in each k and the AUC value for each class. (a) Trained and tested with K1 data. (b) Trained and tested with K2 data. (c) Trained and tested with K3 data. (d) Trained and tested with K4 data. (e) Trained and tested with K5 data

5. CONCLUSION

Earthquake damage prediction is a challenging problem that has recently received a great deal of attention. In this study, we have developed an ANN-based model and pioneer support service for predicting the human casualties that may occur due to a possible earthquake in any country in the world. For this purpose, an open-source dataset containing information such as latitude, longitude, Ml size, depth, loss of life, and property of earthquakes occurring worldwide, obtained from the database of the NOAA (2021), was used. We compared the results of some traditional ML models, such as DT, NB, SVM, LR, and the proposed ANN according to various metrics on the preprocessed dataset. The experimental results showed that the proposed ANN outperformed the other algorithms. Then, to improve the performance of the ANN model, we have used various optimization algorithms, such as GD, MBSGD, RMSProp, ADAM optimizer, and heuristic algorithms, such as GA and PSO. PSO-ANN achieved the best results based on various performance evaluation metrics, such as accuracy, MSE, RMSE, precision, recall, and f1-score. In the proposed PSO-ANN model, the effect of the number of neurons in the hidden layers and the changes in the epoch values on the model were observed. Accordingly, it was determined that the values obtained as a result of training the model with two hidden layers that had eight neurons by using 500 epochs were more successful than the others.

The dataset used in the training and testing phase of the model proposed contained real earthquake data values. However, the dataset is unbalanced because it does not have equal or close numbers of data from each class. Despite this situation, it was shown that the proposed PSO-ANN model obtained successful and acceptable results based on performance metrics. Therefore, it can be concluded that this model can be used effectively for earthquake damage estimation.

In addition, it is believed that these predicted data will be an essential reference for government institutions and non-governmental organizations during emergency planning efforts. However, the proposed model can produce more successful results if the dataset contains more data and balanced classes. This problem can be overcome by creating realistic synthetic data. In future studies, it is planned to use the proposed model by including synthetic data in ready-made and open datasets or by creating our own dataset for a region with high earthquake risk.

Peer-review: Externally peer-reviewed.
Conflict of Interest: The authors have no conflict of interest to declare.
Grant Support: The authors declared that this study has received no financial support.
Author Contributions: Conception/Design of Study- A.B.V.M., Z.O., R.Ş.; Data Acquisition- A.B.V.M.;
Data Analysis/Interpretation- A.B.V.M.; Drafting Manuscript- A.B.V.M., Z.O., R.Ş.; Critical Revision of Manuscript- Z.O., R.Ş.; Final Approval and Accountability- A.B.V.M., Z.O., R.Ş.; Material and Technical Support- A.B.V.M.; Supervision- Z.O., R.Ş..

Hakem Değerlendirmesi: Dış bağımsız.
Çıkar Çatışması: Yazarlar çıkar çatışması beyan etmemişlerdir
Finansal Destek: Yazarlar finansal destek beyan etmemişlerdir.
Yazar Katkıları: Çalışma Konsepti/Tasarım- A.B.V.M., Z.O., R.Ş.; Veri Toplama- A.B.V.M.; Veri Analizi/Yorumlama- A.B.V.M.; Yazı Taslağı- A.B.V.M., Z.O., R.Ş.; İçeriğin Eleştirel İncelemesi- Z.O., R.Ş.; Son Onay ve Sorumluluk- A.B.V.M., Z.O., R.Ş.; Malzeme ve Teknik Destek- A.B.V.M.; Süpervizyon Z.O., R.Ş.

Kaynaklar/References

Abraham A., Rohini V. (2019). A Particle Swarm Optimization-Backpropagation (PSO-BP) Model for the Prediction of Earthquake in Japan. *In Emerging Research in Computing, Information, Communication and Applications*, 882:435-441.

Adjei C., Tian W., Onzo B., Kedjanyi E. A. G. & Darteh O. F., Chen S (2021). Rainfall Forecasting in Sub-Sahara Africa-Ghana using LSTM Deep Learning Approach. *International Journal of Engineering Research & Technology*, 10(3):464-470

Aghamohammadi H., Mesgari M. S., Mansourian A. & Molaei D. (2013). Seismic human loss estimation for an earthquake disaster using neural network. *International Journal of Environmental Science and Technology*, 10:931-939. https://doi.org/10.1007/s13762-013-0281-5

Ahmad N. (2019). Fragility Functions and Loss Curves for Deficient and Haunch-Strengthened RC Frames. *Journal of Earthquake Engineering, accepted*. https://doi.org/10.1080/13632469.2019.1698478

Ahmad N., Ali Q., Ashraf M., Alam B. & Naem A. (2012). Seismic vulnerability of the Himalayan half-dressed rubble stone masonry structures, experimental and analytical studies, *Natural Hazards and Earth System Sciences*, 12:3441-3454. https://doi.org/10.5194/nhess-12-3441-2012

Ahmad N., Ali Q., Crowley H. & Pinho R. (2014). Earthquake loss estimation of residential buildings in Pakistan, *Natural Hazards*, 73: 1889-1955. https://doi.org/10.1007/s11069-014-1174-8

Alizadeh M., Alizadeh E., Kotenaee S. A., Shahabi H., Pour A. B., Panahi M., Ahmad B. B. & Saro L. (2018). Social Vulnerability Assessment Using Artificial Neural Network (ANN) Model for Earthquake Hazard in Tabriz City, Iran, *Sustainability*, 10, ID: 3376. https://doi.org/10.3390/su10103376

Alizadeh M., Ngah I., Hashim M., Pradhan B. & Pour A. B. (2018). A Hybrid Analytic Network Process and Artificial Neural Network (ANP-ANN) Model for Urban Earthquake Vulnerability Assessment, *Remote Sensing*, 10, ID: 975. https://doi.org/10.3390/rs10060975

Anupam S. and Pani P. (2020). Flood forecasting using a hybrid extreme learning machine-particle swarm optimization algorithm (ELM-PSO) model, *Modeling Earth Systems and Environment*, 6:341-347. https://doi.org/10.1007/s40808-019-00682-z

Asim K. M., Idris A., Iqbal. T & Martínez-Álvarez F. (2018). Seismic indicators-based earthquake predictor system using Genetic Programming and AdaBoost classification. *Soil Dynamics and Earthquake Engineering*, 111:1-7. https://doi.org/10.1016/j.soildyn.2018.04.020

Bano P., Singh R. &Aggarwal G. (2021). Forecasting of Flood in Upper Yamuna Basin by Using Artificial Neural Network and Geoinformatics Techniques & Learning, *Elementary Education Online*, 20(5):3008-3021. 10.17051/ilkonline.2021.05.326

Bath, M., (1979). Seismic risk in Fennoscandia, *Tectonophysics,* 57, 285–295.

Berry D. (1996). *Statistics-A Bayesian Perspective*, Duxbury Press.

Botton L. (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. *Proceedings of 19th International Conference on Computational Statistics.* 177-186

Boutkhamouine B., Roux H. & Pérés F. (2020). Data-driven model for river flood forecasting based on a Bayesian network approach, J*ournal of Contingencies and Crisis Management*, 29(3):215-227. https://doi.org/10.1111/1468-5973.12316

Chawla M. & Singh A. (2021). A data efficient machine learning model for autonomous operational avalanche forecasting, *Natural Hazards and Earth System Sciences*, 106. https://doi.org/10.5194/nhess-2021-106

Choubin B., Borji M., Hosseini F. S., Mosavi A. & Dineva A. A. (2020). Mass wasting susceptibility assessment of snow avalanches using machine learning models, *Nature Research*, 10, ID: 18363. https://doi.org/10.1038/s41598-020-75476-w

Cui S., Yin Y., Wang D., Li Z. & Wang Y. (2021). A stacking-based ensemble learning method for earthquake casualty prediction. *Applied Soft Computing*, 101. https://doi.org/10.1016/j.asoc.2020.107038

Dazzi S., Vacondio R. & Mignosa P. (2021). Flood Stage Forecasting Using Machine-Learning Methods: A Case Study on the Parma River (Italy), *Water*, 13:1612-1633. https://doi.org/10.3390/w13121612

Dhunny A. Z., Seebocus R. H., Allam Z., Chuttur M. Y., Eltahan M. & Mehta H. (2020). Flood Prediction using Artificial Neural Networks: Empirical Evidence from Mauritius as a Case Study. *Knowledge Engineering and Data Science*, 3(1):1-10. http://dx.doi.org/10.17977/um018v3i12020p1-10

Eberhart R. C & Kennedy J. (1995). A New Optimizer Using Particle Swarm Theory, in Proceeding of Symposium on Micro Machine and Human Science. Japan: Nagoya, Piscataway, NJswarm optimization and neural network. *Engineering with Computers*, 32:85-97. 10.1109/MHS.1995.494215

Epstein, L. and Lomnitz, C. (1966). A model for the occurrence of large earthquakes, *Nature,* 211, 954–956.

Gessang O. M. & Lasminto U. (2020). The flood prediction model using Artificial Neural Network (ANN) and weather Application Programming Interface (API) as an alternative effort to flood mitigation in the Jenelata Subwatershed. *In Proceedings of 4th International Conference on Civil Engineering Research*. 10.1088/1757-899X/930/1/012080

Goodfellow I., Bengio Y. & Courville A. (2016). *Deep Learning.* MIT Press, Cambridge, 96-161.

Gordan B., Armaghani D. J., Hajihassani M. & Monjezi M. (2016). Prediction of seismic slope stability through combination of particle swarm optimization and neural network. *Engineering with Computers* 32:85-97. https://doi.org/10.1007/s00366-015-0400-7

Gul M. & Guneri A. F. (2016). An artificial neural network-based earthquake casualty estimation model for Istanbul city. Natural Hazards, 84:2163-2178. https://doi.org/10.1007/s11069-016-2541-4

Hadid B., Duviella E. & Lecoeuche S. (2020). Data-driven modeling for river flood forecasting based on a piecewise linear ARX system identification, Journal of Process Control, 86:44-56. https://doi.org/10.1016/j.jprocont.2019.12.007

Hardin D., Guyon I. & Aliferis C. F. (2011). *A Gentle Introduction to Support Vector Machines in Biomedicine*. World Scientific, 2011

Hoo, Z. H., Candlish, J., & Teare, D. (2017). What is an ROC curve?. *Emergency Medicine Journal*, *34*(6), 357-359.

Hssina B., Merbouha A., Ezzikouri H. & Erritali M. (2014). A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, *4*(2), 13-19.

Jena R.& Pradhan B. (2020). Integrated ANN-cross-validation and AHP-TOPSIS model to improve earthquake risk assessment, *International Journal of Disaster Risk Reduction*, 50, ID: 101723. https://doi.org/10.1016/j.ijdrr.2020.101723

Joshi JC, Kaur P, Kumar B, Singh A, Satyawali PK (2021). HIM-STRAT: a neural network-based model for snow cover simulation and avalanche hazard prediction over North-West Himalaya, *Natural Hazards*, 103:1239-1260. https://doi.org/10.1007/s11069-020-04032-6

Kandilli Observatory and Earthquake Research Institute, http://www.koeri.boun.edu.tr/bilgi/buyukluk.htm (16.06.2021)

Kaur, P., Joshi, J. C. & Aggarwal, P. (2022). A multi-model decision support system (MM-DSS) for avalanche hazard prediction over North-West Himalaya. *Nat Hazards* **110,** 563–585. https://doi.org/10.1007/s11069-021-04958-5

Kingma D. P. and Ba J. L. (2015). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations, 1-13. https://doi.org/10.48550/arXiv.1412.6980

Kleinbaum D. G. & Klein M. (2010). *Logistic Regression A Self-Learning Text*, Third Edition, Springer

Li C. & Liu X. (2016). An improved PSO-BP neural network and its application to earthquake prediction. *In Proceeding of Chinese Control and Decision Conference,* 3434-3438. 10.1109/CCDC.2016.7531576

Mcmahan H. B.& Streeter M. (2014). Delay-Tolerant Algorithms for Asynchronous Distributed Online Learning. *Advances in Neural Information Processing Systems*, 1-9.

Mirjalili S. (2019). *Evolutionary Algorithms and Neural Networks Theory and Applications*, Springer. 43-53

Moayedi H., Mehrabi M., Mosallanezhad M., Rashid A. S. A. & Pradhan B. (2019). Modification of landslide susceptibility mapping using optimized PSO-ANN technique. *Engineering with Computers*, 35:967-984. https://doi.org/10.1007/s00366-018-0644-0

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

National Geophysical Data Center / World Data Service: NCEI/WDS Global Significant Earthquake Database. NOAA National Centers for Environmental Information. https://www.ngdc.noaa.gov/hazel/view/hazards/earthquake/search (19.06.2021)

Obasi A. A., Ogbu K. N., Orakwe C. L. & Ahaneku I. E. (2020). Rainfall-river discharge modelling for flood forecasting using Artificial Neural Network (ANN). *Journal Of Water And Land Development*, 44(I-II):98-105. 10.24425/jwld.2019.127050

Rani D. S., Jayalakshmi G. N. & Baligar V.P. (2020). Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks. *In Proceeding of 2nd International Conference on Innovative Mechanisms for Industry Applications*. 10.1109/ICIMIA48430.2020.9074928

Ranit A. B. & Durge P. V. (2019). Flood Forecast Development using Machine Learning, *Research and Innovations in Science and Engineering*, 5(12):155-158.

Rathee A. & Mathur R. P. (2013). Survey on Decision Tree Classification algorithms for the evaluation of Student Performance. *International Journal of Computers & Technology*, 4(2): 244-247

Reyes, J., Morales-Esteban, A., & Martínez-Álvarez, F. (2013). Neural networks to predict earthquakes in Chile. *Applied Soft Computing*, 13(2), 1314-1328.

Rumelhart D. E, Hinton G. E. & Williams R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533-536. https://doi.org/10.1038/323533a0

Saba S., Ahsan F. & Mohsin S. (2017). BAT-ANN based earthquake prediction for Pakistan region. *Soft Computing*, 21:5805-5813. https://doi.org/10.1007/s00500-016-2158-2

Sahoo A., Samantaray S. &Ghose D. K. (2021), Prediction of Flood in Barak River using Hybrid Machine Learning Approaches: A Case Study, *Journal Geological Society of India*, 97:186-198. https://doi.org/10.1007/s12594-021-1650-1

Shan K., Zhang Y., Zheng Y., Li L. & Deng H. (2020). Risk Assessment of Fracturing Induced Earthquake in the Qiabuqia Geothermal Field, China. *Energies*, 13, ID: 5977. https://doi.org/10.3390/en13225977

Shiuly A., Roy N. & Sahu R. B. (2020). Prediction of peak ground acceleration for Himalayan region using artificial neural network and genetic algorithm. *Arabian Journal of Geosciences*, 13:1-10. https://doi.org/10.1007/s12517-020-5211-5

Tao Z. (2015). Artificial Neural Network attempts for long-term evaluation of great earthquakes. *In Proceeding of 11th International Conference on Natural Computation*, 1128-1132. 10.1109/ICNC.2015.7378150

Turkan S. & Ozel G. (2014). Modeling destructive earthquake casualties based on a comparative study for Turkey. *Natural Hazards*, 72:1093-1110. https://doi.org/10.1007/s11069-014-1059-x

Wang H. X., Niu J. X. & Wu J. F. (2011). ANN model for the estimation of life casualties in earthquake engineering. *Systems Engineering Procedia*, 1:55-60. https://doi.org/10.1016/j.sepro.2011.08.010

Xi W. & Li G., Moayedi H, Nguyen H (2019). A particle-based optimization of artificial neural network for earthquake-induced landslide assessment in Ludian county, China. Geomatics. *Natural Hazards and Risk*, 10:1750-1771. https://doi.org/10.1080/19475705.2019.1615005

Xing H., Zhonglin Z., Shaoyu W. (2015). The prediction model of earthquake causality based on robust wavelet v-SVM. *Natural Hazards*, 77:717-732. https://doi.org/10.1007/s11069-015-1620-2

Yuan R. (2021). An improved K-means clustering algorithm for global earthquake catalogs and earthquake magnitude prediction. *Journal of Seismology*, 25:1005-1020. https://doi.org/10.1007/s10950-021-09999-8

Zhan X., Qin H., Liu Y., Yao L., Xie W., Liu G. & Zhou J. (2020). Variational Bayesian Neural Network for Ensemble Flood Forecasting, *Water*, 12:2740-2754. https://doi.org/10.3390/w12102740

Zhou Y., Guo S., Xu C., Chang F. & Yin J. (2020). Improving the Reliability of Probabilistic Multi-Step-Ahead Flood Forecasting by Fusing Unscented Kalman Filter with Recurrent Neural Network, *Water*, 12:578-592. https://doi.org/10.3390/w12020578