Iranian Journal of Geophysics, Vol 17 No 6, 2024, Page 9 – 23 (Research Article)
DOI: 10.30499/IJG.2023.375989.1474

Artificial neural networks to model earthquake magnitude and the direction of its energy propagation: a case study of Indonesia

Hery Leo Sianturi [1*], Valentinus Galih Vidia Putra[2] and Redi Kristian Pingak[3]

[1] Assistant Professor, Department of Physics, Faculty of Sciences and Engineering, University of Nusa Cendana, Kupang, Nusa Tenggara Timur, Indonesia
[2] Assistant Professor, Polytechnic of Indonesian Technical School of Textiles, Bandung, Indonesia
[3] Associate Professor, Department of Physics, Faculty of Sciences and Engineering, University of Nusa Cendana, Kupang, Nusa Tenggara Timur, Indonesia

(Received: 11 December 2022, Accepted: 19 April 2023)

Abstract

Indonesia is a region that experiences frequent earthquakes, and therefore is highly prone to earthquake hazards. Elevated seismicity in Indonesia means that building models to understand and predict earthquake characteristics and their associated hazards is important. The goal of this research is to develop a supervised learning artificial neural network (ANN) application that can predict the magnitude of earthquake wave propagation (bar) and the direction of propagation of earthquake wave using some selected earthquake data (Mw 5-8) happened in Indonesia from 1996 to 2019. The data was taken from the United States Geological Survey (USGS) database and the Indonesian Agency for Meteorological, Climatological, and Geophysics (BMKG) database. The earthquake data used in the artificial neural network application consists of a hidden layer with four neurons and seven input neurons that contain earthquake parameters, including longitude, latitude, magnitude, depth, strike, dip, and rake, as well as one output neuron. The magnitude and direction of energy propagation of the earthquakes were successfully predicted using the ANN program. Excellent agreement between the results of ANN and those of Coulomb 3.3 software strongly indicates that the ANN program can be used as an alternative to the existing Coulomb 3.3 software. The ANN model can also be further applied to other earthquake data around the world. The results are expected to contribute to the development of earthquake detection software tools. With artificial intelligence, earthquake prediction software will be more effective and can reduce the risks of failure in predicting the magnitude and direction of earthquake wave propagation.

Keywords: Earthquake, magnitude, energy propagation, artificial neural network, Coulomb stress

[*]Corresponding author: hlsianturi@staf.undana.ac.id

1 Introduction

Earthquakes of varying magnitudes occur every day around the globe. Indonesia, in particular, is very prone to earthquake hazards due to the frequent earthquakes it experiences, ranging from small-magnitude earthquakes to the most devastating ones. Earthquakes in this region are generated by a variety of sources and are characterized by a very wide diversity of faulting, including extension, thrusting, subduction, and strike-slip faulting (Hutchings and Mooney, 2021).

Elevated seismicity in Indonesia, especially in the past two decades, has led to a large number of studies to better understand Indonesian seismicity and tectonics (Hutchings and Mooney, 2021; Socquet et al., 2019; Bradley et al., 2017; Supendi et al., 2018; Suhardja et al., 2020; Sahara et al., 2021; Patria and Putra, 2020; Nugraha and Hall, 2018; Liu and Harris, 2014; Widiyantoro et al., 2020). Some of these studies suggested that an earthquake with a small magnitude may not be that dangerous, but large magnitude earthquakes can have direct impacts due to the aftershocks that may occur afterwards. The impacts vary from place to place depending on the local conditions where the earthquake strikes.

The emergence of thousands of aftershocks is inseparable from the process of spreading and changing stress on the rock, which is then distributed in all directions so that it disturbs other rocks around it. When the elastic limit of the rock is exceeded, energy is released as a new earthquake because the rock can no longer withstand stress. One method to visualize the distribution of earthquake stress is the Coulomb stress change method (King et al., 1994). Research to study changes in Coulomb stress has been carried out by many researchers (Parsons et al., 2006; Sianturi et al., 2018; Miao and Shou-Biao, 2012).

One of the most commonly used programs to investigate changes in Coulomb stress is the Coulomb Geophysics 3.3 software, an open source provided by the USGS (United States Geological Survey) (Toda et al., 2011). Coulomb is aimed at both publication-directed research and academy classroom instruction. Static deflections (on any surface or at GPS stations), strains, and stresses caused by fault slip, magmatic intrusion, or dike expansion can be calculated. Coulomb is interested in how an earthquake helps promote or prevent failure on nearby faults and how fault slip or dike growth compresses a nearby magma chamber. Geologic deformation caused by strike-slip faults, normal faults, or fault-bend folds is another practical application (Toda et al., 2011). In addition, Artificial Neural Network (ANN) has been used recently to predict the magnitude of earthquakes happening around the world, such as in Southern California (Panakkat and Adeli, 2007), the Northern Red Sea (Alarifi et al., 2012), and Greece (Moustra et al., 2011). The ANN has also been widely implemented in many studies in various fields, including in material science (Nasouri et al., 2013; Brooks and Tucker, 2015; Naghibzadeh and Adabi, 2014; Faridi-Majidi et al., 2012).

Recently, a model based on ANN was used in studies related to earthquakes in Indonesia. Syifa et al. (2019) applied ANN to map post-earthquake damage for the Mw 7.4 earthquake in 2018 striking Donggala, Central Sulawesi, Indonesia, which triggered a tsunami and liquefaction in the surrounding area. Similarly, using 123 seismic datasets, ANN was also applied to predict earthquake causalities and damages in Indonesia (Oktarina et al., 2019). Moreover, Shodiq et al. (2018) developed an ANN model to study the aftershocks of some selected earthquakes in Indonesia. These studies, however, have not studied the prediction of magnitude and direction of energy propagation of earthquakes in Indonesia. Hence, this study aims to develop an ANN program to determine the earthquake magnitude and the direction of its energy propagation for some selected earthquakes occurred in Indonesia from 1996 to 2019 with a magnitude of Mw 5 to 8. The performance of the ANN program is also evaluated by comparing its results with Coulomb 3.3 results. Another novel aspect of this research involves using artificial intelligence to calculate and simulate the magnitude (bar) and direction of earthquake wave propagation for some selected earthquakes. This research will help experts, researchers, and technicians construct a software tool to detect earthquakes. With artificial intelligence, earthquake prediction software will be more productive and will reduce the risks of failure in predicting the magnitude and direction of earthquake wave propagation in the future.

2 Methods

2-1 Artificial Neural Network (ANN)

An earthquake can occur by a sudden slip on a fault. The existence of a fault results in a change in shear stress in the surrounding area. The increase or decrease in shear stress that occurs depends on the position, geometry, and slip of the fault source and the geometry of the receiver, such as the rake. By assuming a simple Coulomb friction model that uses a hemispherical elastic model in a rectangular plane, for earthquakes, the change in Coulomb failure stress is formulated as (Okada, 1992):

$$\sigma_f = \tau_\beta - \mu(\sigma_\beta + P) \qquad (1)$$

where the P value is the pore pressure (bar) which can change the normal stress along the fault plane and is related to the Skeptom coefficient (B) (friction coefficient) with values varying from 0 to 1. τ_β is the change in shear stress on a given fault plane (positive in the direction of fault slip), σ_β is the change in normal fault stress, and $\mu' = \mu(1 - B)$ is the effective friction coefficient. The coordinate system for calculating the Coulomb stress on the fault plane is shown in Figure 1-a. Thus, the change in static Coulomb stress (Coulomb Failure Function) caused by a primary vibration with simple assumptions for the effect of pore pressure (King et al., 1994) is:

$$\sigma_f = \tau_\beta + \mu'\sigma_\beta \qquad (2)$$

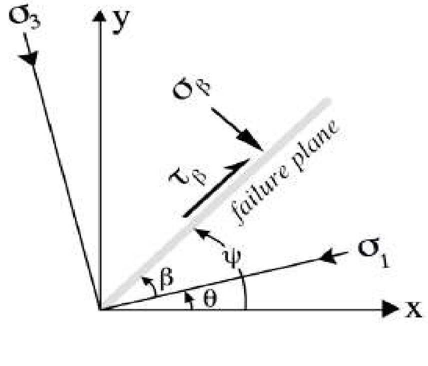
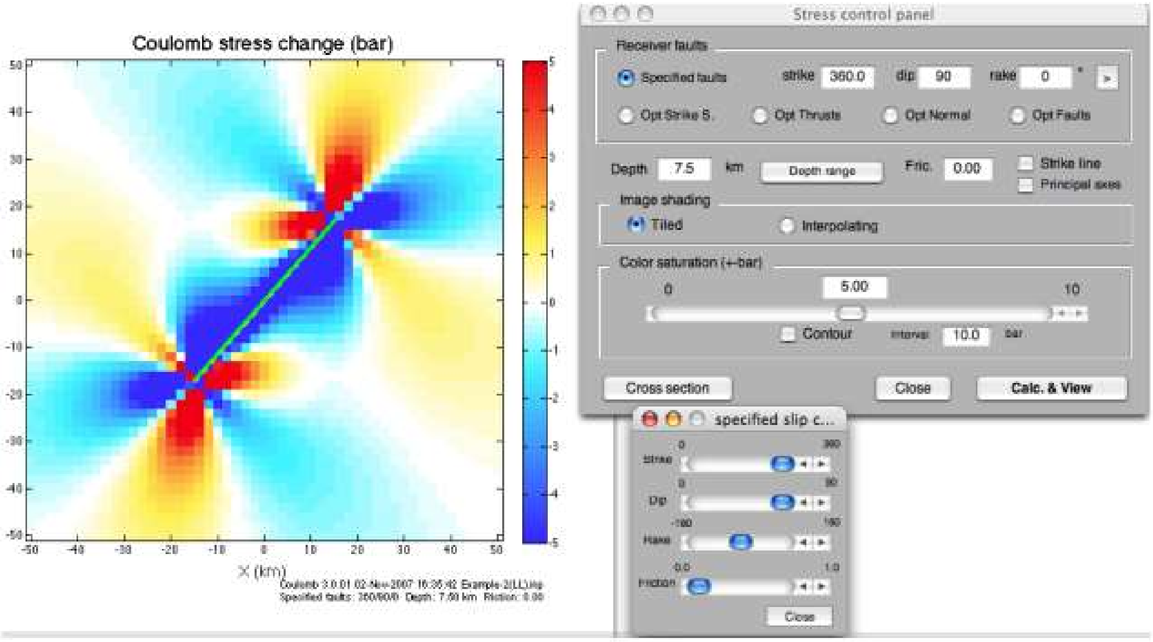


Figure 1. (a) The coordinate system for calculating Coulomb stress on fault planes (King et al., 1994).

The Coulomb Failure Function (Coulomb stress change) can be described in terms of the magnitude of the earthquake wave propagation and the direction of earthquake wave propagation. In Coulomb, there are four kinds of receiver faults: (1) "specified" receiver faults in which all faults have uniform receiver fault geometry, (2) faults optimally oriented for failure, (3) receiver faults with the geometry set in the input file, and (4) focal mechanism files, in which there are always two orthogonal fault planes. Illustration of data input and results of Coulomb stress changes in Coulomb software 3.3 on specified receiver fault planes, using the default "specified faults" is shown in Figure 1-b. Coulomb averages the information on all input fault patches in the input file and puts the values in the boxes, such as strike, dip, and rake. For practice, choose a strike/dip/rake of 360°/90°/0° (for example), change the friction coefficient to 0.0, set the stress change color saturation to ±5 bars (for example) to compute magnitude and direction of earthquake propagation, and hit "Calc. & View", resulting in optimum slip planes rotating near the fault. If the regional deviatoric stress is much larger than the earthquake stress drop (red color), the orientations of the optimum slip planes are more limited, and regions of increased Coulomb stress diminish in size and become more isolated from the source fault. In this and subsequent plots, the maximum and minimum stress changes exceed the plotted color bar range (in other words, the scale is saturated).



Coulomb 3.0.01 02-Nov-2007 16:36:42 Example-2(LL).inp
Specified faults: 360/90/0 Depth: 7.50 km Friction: 0.00

Stress control panel

Receiver faults

● Specified faults strike 360.0 dip 90 rake 0 °

○ Opt Strike S. ○ Opt Thrusts ○ Opt Normal ○ Opt Faults

Depth 7.5 km Fric. 0.00 □ Strike line □ Principal axes

Image shading: ● Tiled ○ Interpolating

Color saturation (+-bar) 5.00 (0 – 10)

□ Contour interval 10.0 bar

specified slip c...

Strike 0 – 360
Dip 0 – 90
Rake -180 – 180
Friction 0.0 – 1.0

Figure 1. (b) Illustrasion of data input and results of Coulomb stress changes in Coulomb software 3.3 on specified receiver fault planes.

The ANN method is a branch of artificial intelligence that uses mathematical or computational models where the way it works is a simplification of the biological network model of the human brain. This is because, in principle, ANN is a computer program created based on the workings of the biological network of the human brain. In terms of its function, ANN was created to design a computer that can be used to carry out the learning process from an example of an incident, while from the design structure. ANN is a calculating device that is intended to be able to do something similar to the workings of the biological network of the human brain. Like the human brain, a neural network also consists of several neurons that have connections with one another. These neurons will transform the information received through their exit connections to other neurons; this relationship is known as the weight. In a neural network, neurons are collected in layers called neuron layers.

The ANN structure generally consists of an input layer, a hidden layer, and an output layer. The input layer contains neurons that receive data directly from outside. The hidden layer receives signals from the input layer and forwards them to the output layer. The effect of the number of input and output layers on the results and accuracy of work is that the more layers, the higher the value of the results and accuracy, but the longer it takes the computer to analyze. The output layer contains neurons that represent the target and the output of the calculation model. The difference between the target and the output of the ANN calculation model is the ANN error rate.

In general, ANN has one or more hidden layers and an output layer. The ANN process is calculated from the input layer to the hidden layer. Furthermore, the results of the hidden layer are used to calculate the output layer. Errors in the output cells are corrected, and the latest error values are recalculated by adjusting the weight through the hidden layer, and then calculated back to the input layer. Training the neural network in this model means changing the weights. The weights are changed by minimizing the sum of the squares of the differences between the target and network output values. We use the back-propagation algorithm to train neural networks in this study. This algorithm updates the network weights when the performance function rapidly decreases. Therefore, the convergence of errors can take some time, depending on the errors allowed in the output layer.

An ANN consists of a hidden layer with multiple nodes (neurons), input neurons, and one output neuron. The hidden neuron activation function adopts a sigmoid function, and the output neuron adopts a linear function. The neuron in this study enters the weighted sum into the linear function (activation function) and produces its output. The output activation function considers the neuron's system, which is commonly a linear function. An artificial neural network consisting of these variables is shown in Figure 2.

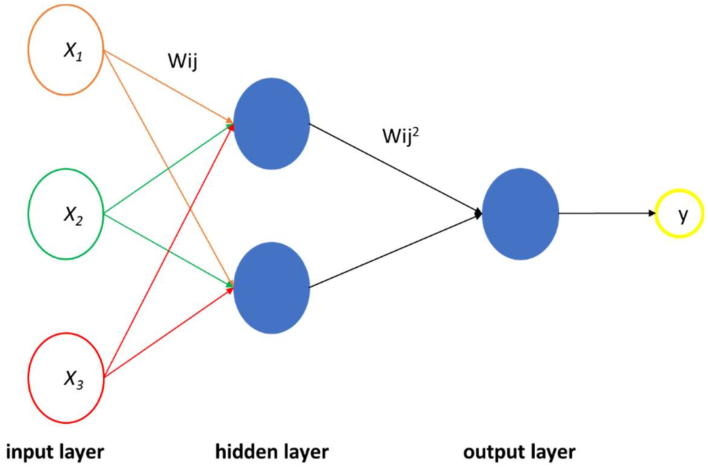


Figure 2. Schematic of an Artificial Neural Network (ANN).

Circles and arrows in Figure 2 indicate neurons and signal flow, respectively. The terms x_1, x_2, and x_3 are inputs. w_{ij} and w_{ij}^2 are the first and second layer weight matrices, respectively. The input can be multiplied by the weights before reaching the neurons. Once the neuron-weighted inputs have been collected, these values are added up as a weighted sum. In this study, we modified the equations of artificial neural networks used by several researchers (Alarifi et al., 2012; Alizadeh et al., 2018). The weighted amount can be calculated using Eq. (3):

$$v_i = \sum_i \sum_j w_{ij} x_j = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}_{2\times 3} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{3\times 1} \qquad (3)$$

where w_{ij} and x_j are weights and inputs, respectively.

The neuron enters the weighted amount into the activation function (sigmoid function), and produces its output as Eq. (4):

$$\begin{pmatrix} o_1 \\ o_2 \end{pmatrix} = \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \end{pmatrix} \qquad (4)$$

Eq. (4) is the input data for the second screen and one gets the total μ_i in the second screen as Eqs. (5) and (6):

(5)

$$\mu_i = \sum_i \sum_j w_{ij}{}^{(2)} o_j = \sum_i \sum_j w_{ij}{}^{(2)} \varphi(v_j)$$

$$\mu_i = \begin{pmatrix} w_{11}{}^{(2)} & w_{12}{}^{(2)} \end{pmatrix} \times \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \end{pmatrix} \qquad (6)$$

The neuron inputs the sum of the weights into the linear function and produces the output. The output activation function (linear function) defines the behaviour of the neuron, and can be expressed as:

$$y = \psi(\mu_i) = \psi\left(\sum_i \sum_j w_{ij}{}^{(2)} \varphi(v_j)\right) \quad (7)$$

where $\varphi(v)$ and $\psi(\mu)$ are sum function and a linear function, respectively.

A systematic technique for adjusting weights according to the information provided is called a learning rule. Since training is a neural network's way of storing data, learning rules are an important element of neural network research. Assuming that the neural network has the correct output d_i (where d_i is the correct output of the neuron-i output), then the error of the output node-i can be written as:

$$e_i = d_i - \psi(\mu_i) \qquad (8)$$

The delta rule can be formulated as:

$$w_{ij} \leftarrow w_{ij} + \alpha \delta_i x_j \qquad (9)$$

$$w_{ij} \leftarrow w_{ij} + \alpha \dot{\psi}(v) e_i x_j \qquad (10)$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \qquad (11)$$

where α and δ_i are the rates ($0<\alpha<1$) and the delta function is output i. e_i is an error from output i and $\dot{\psi}(v)$ is the derivative of the linear function $\psi(v)$ at the result of node i. In the hidden screen, the sigmoid function, $\varphi(v)$, and its derivative, $\dot{\varphi}(v)$ can be written as (Alarifi et al., 2012; Alizadeh et al., 2018):

$$\varphi(v) = \frac{1}{1+e^{-v}} = (1+e^{-v})^{-1} \qquad (12)$$

$$\dot{\varphi}(v) = \frac{d\varphi(v)}{dv} = \varphi(v)\big(1-\varphi(v)\big) \qquad (13)$$

This study uses a back-propagation algorithm to train a neural network algorithm. This algorithm updates the network weights where the performance function degrades rapidly.

ANN Backpropagation is an algorithm that is often used in solving complex problems. This method is an excellent method for dealing with complex pattern recognition problems. The characteristics of the activation function of ANN Backpropagation are that it must be continuous, differentiable and monotonic (non-decreasing). The function is written in the form of Eqs. (14) and (15):

$$f(x) = \frac{1}{1+e^{-x}} \qquad (14)$$

$$f(x)' = f(x)[1 - f(x)] \qquad (15)$$

ANN Backpropagation calculations consist of three processes, namely forward calculations, back-propagation calculations, and changes in weights and biases. The data is entered for each neuron in the input layer and the calculations are continued up to the output layer (forward calculations). The difference between the output value predicted by ANN and the actual output value (target value) is called the error rate. The error value is then used to modify the ANN weight factor in the backward calculation process from the output layer to the input layer.

The prediction accuracy of ANN can be determined by calculating the coefficient of determination (R^2). The value of R^2 is determined by Eq. (16), which is a function of the difference between the predicted output of ANN (Y_k) and the target value (T_k):

$$R^2 = 1 - \left[\frac{\sum_k^n (T_k - Y_k)^2}{\sum_k^n (T_k - Y_{avg})^2}\right] \qquad (16)$$

where the magnitude of Y_{avg} is the average value of all the target values of *n* pieces of data. The closer the R^2 value to 1, the better the ANN prediction of the actual (target) value.

The input and target parameters in ANN generally have different dimensions, units, and ranges of values. The raw data used to train the ANN must be normalized first to reduce the gap between input and output parameter values. Normalization of ANN raw data is done by Eq. (17) (Alarifi et al., 2012; Alizadeh et al., 2018):

$$V_{new} = \frac{V_{old} - V_{min}}{V_{max} - V_{min}} x(D_{max} - D_{min}) + D_{min} \qquad (17)$$

The cells represent the function (γ) which sums the products of the weights (w_i) and input (u_i) and also adds the bias (w_0) as shown in Eq. (18):

$$\gamma = w_0 + \sum_{i=1}^{3} u_i\, w_i \qquad (18)$$

In general, an Artificial Neural Network (ANN) has one or more hidden layers and an output layer. The ANN process is calculated from the input layer to the hidden layer. Furthermore, the results of the hidden layer are used to calculate the output layer. Errors in the output cells are corrected and the latest error values are recalculated by adjusting the weight through the hidden layer, and then calculated back to the input layer. Therefore, the convergence of errors can take some time, depending on the errors allowed in the output layer. The total error in the ANN output is defined as:

$$E = \frac{1}{2}\sum_{j \in J}(T_j - O_j)^2 \qquad (19)$$

where T_j denotes the target output, O_j denotes the activation value of the output layer, and J is the number of iterations for the learning process.

2-2 Implementation of ANN

In the present study, an artificial neural network was created using earthquake data consisting of a hidden layer with 4 neurons, 7 input neurons comprising 7 earthquake parameters, i.e., longitude, latitude, magnitude, depth, strike, dip, and rake, and one output neuron. The hidden neuron activation function adopts a sigmoid function whereas the output neuron adopts a linear function (Figure 3). Earthquake data used in this study was adapted from USGS database and the Indonesian Agency for Meteorological, Climatological, and Geophysics (BMKG) database. The data included earthquakes occurring between 1996 and 2019 with a magnitude of Mw 5-8.

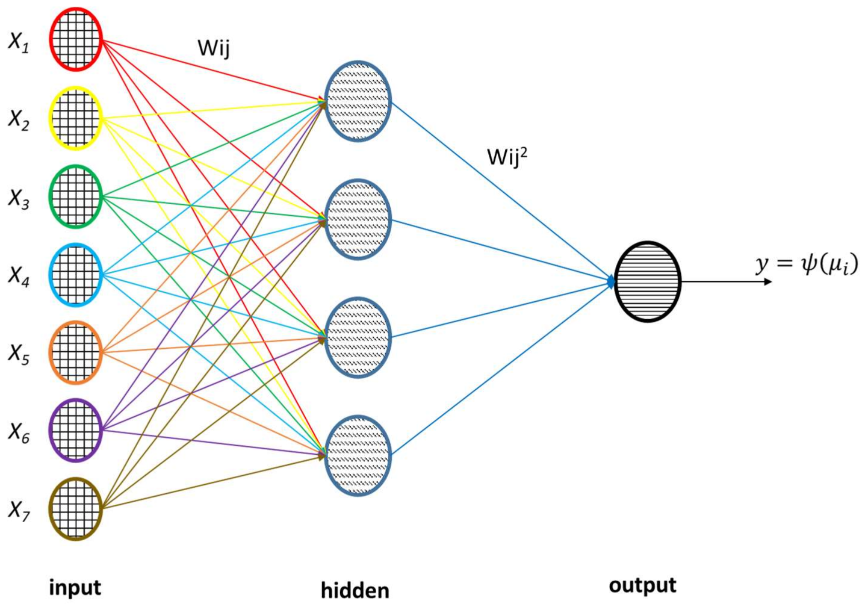


Figure 3. Artificial neural network model.

The circles and arrows in Figure 3 represent neurons and signal flows of the form $x_1, x_2, x_3,$ and x_4 are the earthquake input parameters. $\mathrm{w_{ij}}$ and $\mathrm{w_{ij}}^2$ are the matrix weights of the first and second screens. The total weight can be calculated using Eqs. (20) and (21):

$$v_i = \sum_i \sum_j w_{ij} x_j \tag{20}$$

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix}_{7\times1} = \begin{pmatrix} w_{11}\, w_{12} w_{13} w_{14} w_{15} w_{16} w_{17} \\ w_{21} w_{22} w_{23} w_{24} w_{25} w_{26} w_{27} \\ w_{31} w_{32} w_{33} w_{34} w_{35} w_{36} w_{37} \\ w_{41} w_{42} w_{43} w_{44} w_{45} w_{46} w_{47} \end{pmatrix}_{7\times4} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}_{4\times1} \tag{21}$$

Once the number of neurons are substituted into the activation function (sigmoid function), the output is obtained:

$$\begin{pmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \end{pmatrix}_{4x1} = \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \\ \varphi(v_3) \\ \varphi(v_4) \end{pmatrix}_{4x1} \tag{22}$$

Eq. (22) will be the second screen input, the results of which are shown in Eqs. (23) and (24):

$$\mu_i = \sum_i \sum_j w_{ij}{}^{(2)} o_j = \sum_i \sum_j w_{ij}{}^{(2)} \varphi\left(v_j\right) \tag{23}$$

$$\mu_i = \left(w_{11}{}^{(2)} \quad w_{12}{}^{(2)} \quad w_{13}{}^{(2)} \quad w_{14}{}^{(2)}\right)_{1\times4} \times \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \\ \varphi(v_3) \\ \varphi(v_4) \end{pmatrix}_{4\times1} \tag{24}$$

The result of the activation function (linear function) is the behaviour of the neuron and can be expressed as:

$$y = \psi(\mu_i) = \psi\left(\sum_i \sum_j w_{ij}{}^{(2)} \varphi\left(v_j\right)\right) \tag{25}$$

In this model, the goal of creating an artificial neural network is to change the weights and minimize the sum of the squares of the differences between the target and the output values.

2-3 Validation of ANN results based on Coulomb 3.3 software

After creating the artificial neural network program, it is validated using the same data by Coulomb 3.3 software. Flowchart for calculating Coulomb stress and strain based on Coulomb 3.3 software with a friction coefficient of 0.4 is presented in Figure 4.

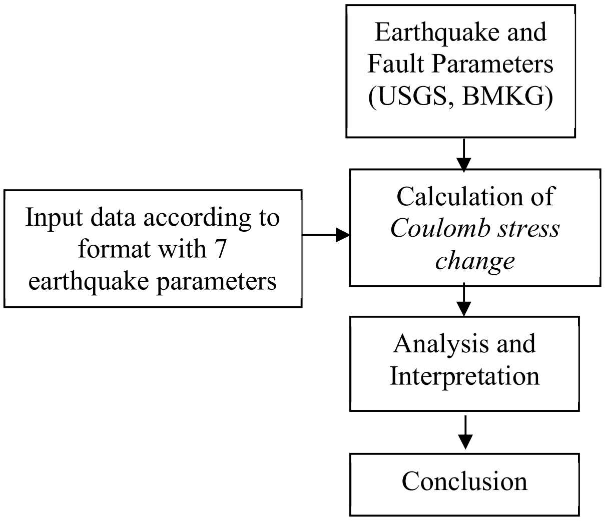
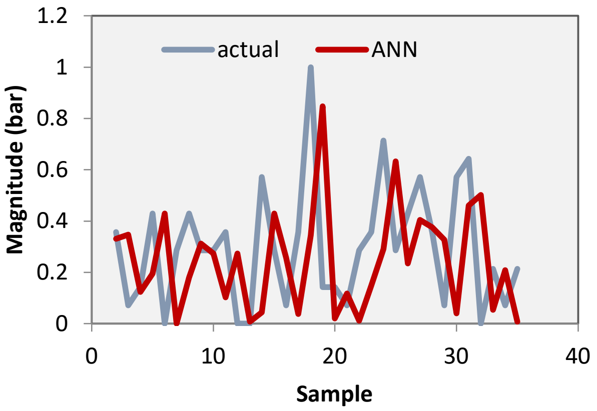

Earthquake and Fault Parameters (USGS, BMKG)
Input data according to format with 7 earthquake parameters
Calculation of *Coulomb stress change*
Analysis and Interpretation
Conclusion

Figure 4. Flowchart of calculations with Coulomb 3.3 software.

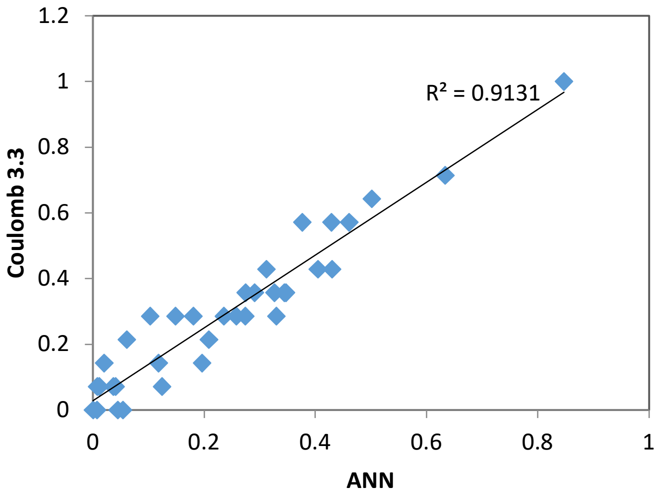
3 Results and discussion

3-1 Determination of the earthquake magnitude

In this model, five variables are used as free variables x_i, where x_1, x_2, x_3, x_4, and x_5 are depth, magnitude, strike, dip, and rake, respectively. The artificial neural network operates four nodes in the hidden layer to receive five inputs x_1, x_2, x_3, x_4, and x_5 and uses 100,000 iterations. According to empirical evidence in the computational physics and artificial intelligence laboratory, the computer can process data quickly while achieving precise modeling results after 100,000 iterations. The magnitude of earthquake wave propagations of some earthquake samples based on this model is presented in Figure 5-a, where the actual data is also shown. It is obvious from the figure that the results of the ANN model agree very well with the actual data. In this study, we used the R-squared (R^2) coefficient to demonstrate a satisfactory adjustment of the proposed model to the experimental data. Based on this study, the model is considered effective in predicting the magnitude of the earthquakes wave propogation (in bars) with R^2 value of 91.31% (Figure 5-b). The result of the model, $y_i = \psi(\mu_i)$, can be formulated as follows:



(a)



(b)

Figure 5. (a) Magnitude of earthquake wave propagation from ANN model compared with actual data in the unit (bar). (b) Comparison between ANN and Coulomb 3.3 results.

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}_{5x1} = \begin{pmatrix} 7.4751 & -8.7586 & -0.9049 & 0.2632 & 4.5147 \\ -0.3815 & 0.3329 & 5.2823 & -3.7359 & -3.5981 \\ 7.0753 & 6.7997 & 9.7022 & -14.5785 & -6.8419 \\ -0.0677 & -1.4207 & 3.3861 & -2.2888 & -1.6414 \end{pmatrix}_{5\times4} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}_{4\times1}$$

$$\mu_i = (-8.5828\ ^{(2)} \quad -12.6258^{(2)} \quad 8.0666\ ^{(2)} \quad -14.0501^{(2)})_{1x4} \times \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \\ \varphi(v_3) \\ \varphi(v_4) \end{pmatrix}_{4\times1}$$

$$y_i = \psi(\mu_i) = -600{,}798\ \varphi(v_1) - 883{,}806\varphi(v_2) + 564{,}662\varphi(v_3) - 983{,}507\varphi(v_4)$$

3-2 Determination of the direction of earthquake wave propagation

In this section, we give more details about the Coulomb 3.3 software and provide an example output of our method, both ANN and Coulomb 3.3. In this model, five earthquake source parameter variables are used as independent variables x_i, where x_1, x_2, x_3, x_4, and x_5 are depth, magnitude, strike, dip, and rake, respectively. An artificial neural network operates four nodes in the hidden layer to receive five inputs x_1, x_2, x_3, x_4, and x_5 using 100,000 iterations. Prediction of the direction of earthquake wave propagation from the ANN model is shown in Figure 6-a. Comparison with the actual data in the same figure shows an excellent agreement. Figure 6-b presents the comparison between results obtained from the ANN model and those from the Coulomb 3.3 program. As seen from the figure, good agreement is observed between the ANN model and the Coulomb

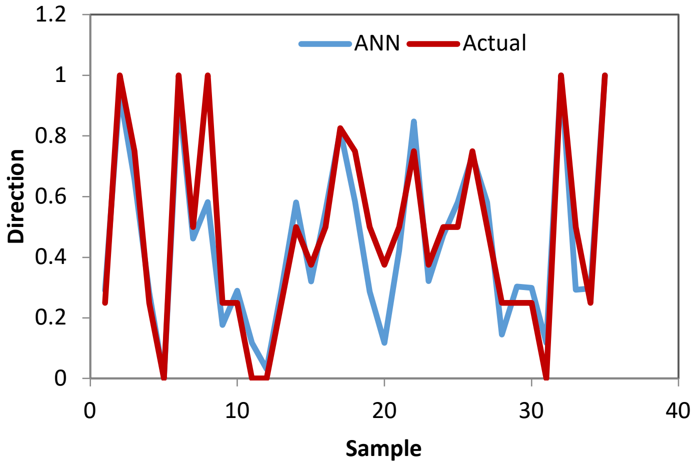
3.3 (R^2=85.88%). This implies that the ANN model is an effective model that can be used to predict the direction of earthquake wave propagation.

The result of the model , $y_i = \psi(\mu_i)$. can be formulated as follows:

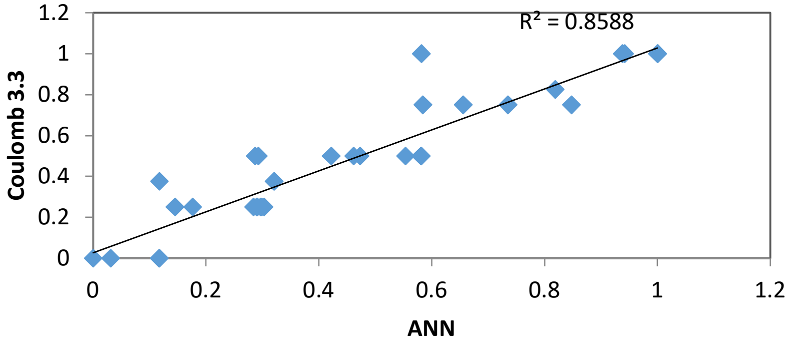
$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}_{5x1} = \begin{pmatrix} 38.9951 & -15.3374 & 10.9009 & 5.3144 & 18.7751 \\ -7.9803 & -16.2337 & 27.3341 & 12.6968 & -7.7837 \\ -43.4927 & 8.4118 & -12.3550 & -24.4871 & 24.0801 \\ 38.0894 & -20.1195 & -15.5644 & -16.1583 & 5.3950 \end{pmatrix}_{5\times 4} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}_{4\times 1}$$

$$\mu_i = (33.9902\ ^{(2)} \quad -36.0066^{(2)} \quad -33.6482\ ^{(2)} \quad -34.8866^{(2)})_{1\times 4} \times \begin{pmatrix} \varphi(v_1) \\ \varphi(v_2) \\ \varphi(v_3) \\ \varphi(v_4) \end{pmatrix}_{4\times 1}$$

$$y_i = \psi(\mu_i) = 33.9902\ \varphi(v_1) - 36.0066(v_2) - 33.6482(v_3) - 34.8866\varphi(v_4)$$



(a)



(b)

Figure 6. (a) Direction of earthquake wave propagation from ANN model compared with actual data. (b) Comparison between ANN and Coulomb 3.3 results.

3-3 Comparison of results from ANN and Coulomb software 3.3

In this section, we give more details about the Coulomb 3.3 software and provide an example output of our method, both ANN and Coulomb 3.3. Based on our findings, Table 1 compares earthquake magnitude from ANN and Coulomb 3.3 software, while Table 2 shows the direction of earthquake wave propagation obtained from the ANN method and Coulomb 3.3 program. The magnitude of earthquake wave propagations of some earthquake samples based on this model is presented in Table 1 and Figure 5, where the actual data is also shown. It is evident from the Table 1 that the results of the ANN model agree very well with the actual data. The proposed model was adjusted satisfactorily to the experimental data using this R-squared (R^2) coefficient. In this study, the model effectively predicted earthquake wave propagation (in bars) with an R^2 value of 91.31 percent, but the accuracy is less than 95 percent, so the model results do not precisely match the Coulomb 3.3 results. Table 2 compares results obtained from the ANN model and those from the Coulomb 3.3 program. Table 2 and Figure 6 show good agreement between the ANN model and the Coulomb 3.3 (R^2=85.88%). Since the modeling accuracy of software data is only above 80% but less than 95%, the numbers in each volume in the Table 2 are not the same, but these results are considered good because R^2 values above 80% produce quite good results, while those below 80% are quite poor.

Table 1 presents the comparison between earthquake magnitude from ANN and Coulomb 3.3 software, while Table 2 shows the direction of earthquake wave propagation obtained from the ANN method and Coulomb 3.3 program. The two tables clearly indicate that the magnitude and direction of wave propagation of 35 earthquake samples calculated from the two models show very good agreement. Hence, the ANN program used in the present study can be used as an alternative to the Coulomb 3.3 software in predicting both the earthquake magnitude and the direction of its wave propagation.

3-4 Performance of the ANN model

Good results generated from an ANN model in this article is consistent with other reports in the literature. For instance, Huang et al. (2022) recently employed an ANN model to develop a probabilistic seismic demand model, the results of which were also compared to the conventional linear regression models. Their results from the ANN model are closer to the 1:1 line, indicating a good correlation between the predicted and measured data. They also showed that reliable fragility models can be generated using the employed ANN model. Furthermore, it was reported that A BP ANN model was successfully built for earthquake magnitude prediction in Himalayas (Narayanakumar and Raja, 2016) using ANN simulation with nine input nodes, 12 hidden layers of 4 nodes each, a Tan-sigmoid activation function, and an output. The results of this modeling have flaws, such as the computer's processing time being quite long and the input data being quite large, but having good prediction results similar to our neural networks. This study showed that the ANN model yielded good predictions for earthquakes of magnitude between 4.0 and 6.0. Similar results were also reported in some studies such as Moustra et al. (2011), Pozos-Estrada and Gómez (2014), Oktarina et al. (2019), and Mignan and Broccardo (2020).

ANN models can also be combined with other models to obtain more accurate prediction of earthquake characteristics.

For example, Alizadeh et. al. (2018) developed a new hybrid framework utilizing ANP (Analytic Network Process) and ANN models to construct a composite economic, environmental, social, and physical vulnerability index, applied to Tabriz city, a seismic-prone province in Iran. The model, they claimed, can be replicated and applied to other sismic-prone regions around the world. Additionally, a model based on support vector regressor and hybrid neural networks was also successfully developed and applied to Hindukush, Chile and Southern California regions (Asim et al., 2018). They showed that using this hybrid model, better prediction can be obtained for all considered regions compared to all previous studies.

All these studies indicate that the use of ANN in earthquake forcasting is promising and that the accuracy of the forecast can further be improved using hybrid models involving ANN models. In general, such models are not only applicable to specific regions, but can also be replicated and applied to other regions prone to seismic hazards.

4 Conclusions

An Artificial Neural Network (ANN) program has been created to determine the magnitude (bar) and direction of earthquake wave propagation of some selected earthquakes in Indonesia. The obtained results have a determinant coefficient R^2 of 0.9131 for the magnitude of earthquake wave propagation and R^2 of 0.8588 for the direction of earthquake wave propagation. These results prove that the ANN program developed can be used as an alternative to the existing Coulomb stress 3.3 software. The results of the present study are expected to make a contribution to the development of a software tool to detect earthquakes. Artificial intelligence also leads to more productive earthquake prediction software in the future.

References

Alarifi, A. S. N., Alarifi, N. S. N., and Al-Humidan, S., 2012, Earthquakes magnitude predication using artificial neural network in northern Red Sea area: Journal of King Saud University- Science, **24**, 301-313.

Alizadeh, M., Ngah, I., Hashim, M., Pradhan, B., and Pour, A. B., 2018, A hybrid analytic network process and artificial neural network (ANP-ANN) model for urban earthquake vulnerability assessment: Remote Sensing, **10**(6), 975.

Asim, K. M., Idris, A., Iqbal, T., and Martínez-Álvarez, F., 2018, Earthquake prediction model using support vector regressor and hybrid neural networks: PLoS ONE, **13**(7), e0199004.

Bradley, K. E., Feng, L., Hill, E. M., Natawidjaja, D. H., and Sieh, K., 2017, Implications of the diffuse deformation of the Indian Ocean lithosphere for slip partitioning of oblique plate convergence in Sumatra: Journal of Geophysical Research: Solid Earth, **122**, 572–591.

Brooks, H., and Tucker, N., 2015, Electrospinning predictions using artificial neural networks: Polymer, **58**(1), 22–29.

Faridi-Majidi, R., Ziyadi, H., Naderi, N., and Amani, A., 2012, Use of artificial neural networks to determine parameters controlling the nanofibers diameter in electrospinning of nylon-6, 6: Journal of Applied Polymer Science, **124**(2), 1589–1597.

Huang, Z., Argyroudis, S. A., Pitilakis, K., Zhang, D., and Tsinidis, G., 2022, Fragility assessment of tunnels in soft soils using artificial neural networks: Underground Space, **7**(2), 242-253.

Hutchings, S. J., and Mooney, W. D., 2021, The seismicity of Indonesia and tectonic implications: Geochemistry, Geophysics, Geosystems, **22**, e2021GC009812.

King, G. C. P., Stein, R. S., and Lin, J., 1994, Static stress changes and the triggering of earthquakes: Bulletin of the Seismological Society of America, **84**(3), 935-953.

Liu, Z. Y. C., and Harris, R. A., 2014, Discovery of possible mega-thrust earthquake along the Seram trough from records of 1629 tsunami in eastern Indonesian region: Natural Hazards, **72**, 1311–1328.

Miao, M., and Shou-Biao, Z., 2012, Study of the impact of static Coulomb stress changes of megathrust earthquakes along subduction zone on the following aftershocks: Chinese Journal of Geophysics, **55**(5), 539-551.

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

Moustra, M., Avraamides, M., and Christodoulou, C., 2011, Artificial neural networks for earthquake prediction using time series magnitude data or Seismic Electric Signals: Experts Systems with Applications, **38**(12), 15032-15039.

Naghibzadeh, M., and Adabi, M., 2014, Evaluation of effective electrospinning parameters controlling gelatin nanofibers diameter via modelling artificial neural networks: Fibers and Polymers, **15**(4), 767–777.

Narayanakumar, S., and Raja, K., 2016, A BP artificial neural network model for earthquake magnitude prediction in Himalayas, India: Circuits and Systems, **7**, 3456-3468.

Nasouri, K., Shoushtari, A. M., and Khamforoush, M., 2013, Comparison between artificial neural network and response surface methodology in the prediction of the production rate of polyacrylonitrile electrospun nanofibers: Fibers and Polymers, **14**(11), 1849–1856.

Nugraha, A. M. S., and Hall, R., 2018, Late Cenozoic paleogeography of Sulawesi, Indonesia: Palaeogeography, Palaeoclimatology, Palaeoecology, **490**, 191–209.

Okada, Y., 1992, Internal deformation due to shear and tensile faults in a half-space: Bulletin of the Seismology Society of America, **82**, 1018–1040.

Oktarina, R., Bahagia, S. N., Diawati, L., and Pribadi, K. S., 2019, Artificial neural network for predicting earthquake casualties and damages in Indonesia: IOP Conference Series: Earth and Environmental Science, **426**, 012156.

Panakkat, A., and Adeli, H., 2007, Neural network models for earthquake magnitude prediction using multiple seismicity indicators: International Journal of Neural Systems, **17**(1), 13-33.

Parsons, T., Robert, S. Y., Yuji, Y., and Ahmad, H., 2006, Static stress change from the 8 October, 2005 M = 7.6 Kashmir earthquake: Geophysical Research Letters, **33**(6).

Patria, A., and Putra, P. S., 2020, Development of the Palu-Koro fault in NW Palu Valley, Indonesia: Geoscience Letters, **7**, 1–11.

Pozos-Estrada, A., and Gómez, R., 2014, Use of neural network to predict the peak ground accelerations and pseudo spectral accelerations for Mexican Inslab and Interplate Earthquakes: Geofísica Internacional, **53**(1), 39-57.

Sahara, D. P., Nugraha, A. D., Muhari, A., et al., 2021, Source mechanism and triggered large aftershocks of the Mw 6.5 Ambon, Indonesia earthquake: Tectonophysics, **799**, 228709.

Shodiq, M. N., Kusuma, D. H., Rifqi, M. G., Barakbah, A. R., and Harsono, T., 2018, Neural network for earthquake prediction based on automatic clustering in Indonesia: International Journal on Informatics Visualization, **2**(1), 37-43.

Sianturi, H. L., Mohamad, J. N., Mala, H.

U., and Tanesib, J. L., 2018, The identification of stress Coulomb changes in Lombok earthquakes on August 5 and 9, 2018: Proceeding the first international conference and exhibition on sciences technology, ICEST, University of Nusa Cendana, Labuan Bajo, Indonesia, 174-183.

Socquet, A., Hollingsworth, J., Pathier, E., and Bouchon, M., 2019, Evidence of supershear during the 2018 magnitude 7.5 Palu earthquake from space geodesy: Nature Geoscience, **12**, 192–199.

Suhardja, S. K., Widiyantoro, S., Métaxian, J. P., Rawlinson, N., Ramdhan, M., and Budi-Santoso, A., 2020, Crustal thickness beneath Mt. Merapi and Mt. Merbabu, Central Java, Indonesia, inferred from receiver function analysis: Physics of the Earth and Planetary Interiors, **302**, 106455.

Supendi, P., Nugraha, A. D., Puspito, N. T., Widiyantoro, S., and Daryono, D., 2018, Identification of active faults in West Java, Indonesia, based on earthquake hypocenter determination, relocation, and focal mechanism analysis: Geoscience Letters, **5**, 31.

Syifa, M., Ryoo, S., and Lee, C. W., 2019, Post-earthquake damage mapping using artificial neural network and support vector machine classifiers at Palu, Indonesia: IGARSS 2019- 2019 IEEE International Geoscience and Remote Sensing Symposium, 9577-9580.

Toda, S., Stein, R. S., Sevilgen, V., and Lin, J., 2011, Coulomb 3.3 graphic-rich deformation and stress-change software for earthquake, tectonic, and volcano research and teaching—user guide: U.S. Geological Survey Open-File Report 2011-1060, 63 p., available at http://pubs.usgs.gov/of/2011/1060/.

Widiyantoro, S., Gunawan, E., Muhari, A., Rawlinson, N., Mori, J., and Hanifa, N. R., 2020, Implications for megathrust earthquakes and tsunamis from seismic gaps south of Java Indonesia: Nature: Scientific Reports, **10**, 15274.