International Journal of Informatics and Computation (IJICOM)
Vol.6, No.2, 2024
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Prediction of Peak Ground Acceleration (PGA) in Java Using Artificial Neural Network Method

Sofyan Hadi Rahmawan[1], Cahyo Crysdian[2], Sri Harini[3]

Abstract

Java is one of the islands in Indonesia that frequently experiences earthquakes. Earthquakes can cause significant ground motion that can damage buildings and threaten human life. Peak Ground Acceleration (PGA) is a measure of the maximum ground acceleration that occurs during an earthquake and is an important factor that must be considered at every construction site to assess the potential damage that can be caused by an earthquake. The parameters considered in determining PGA predictions are earthquake parameters, such as magnitude and hypocenter distance. In addition, the PGA value is also influenced by local site conditions. With advances in information technology and artificial intelligence, especially in the development of Artificial Neural Networks (ANN), research on PGA prediction needs to be conducted as one of the efforts to reduce the risk of earthquakes. The purpose of this research is to obtain the best network architecture for predicting PGA values. The criteria for selecting the best network architecture is done by comparing the error value of each possible architecture formed. The best prediction results are obtained in the model with 3-15-1 architecture with a correlation value of 0.67.

Keywords:
Earthquake, Prediction, Peak Ground Acceleration (PGA), Artificial Neural Network.

1. Introduction

Java Island is one of the islands in Indonesia that often experiences earthquakes. The seismo-tectonics of Java Island can be divided into two lanes, namely the southern Java subduction seismo-tectonic lane and the Java mainland active fault seismo-tectonic lane [1]. The regional tectonics of Java Island is mainly controlled by the perpendicular subduction zone between the Indian Ocean Plate and the Eurasian Continental Plate in the south of Java Island [2]. The results of earthquake activity monitoring conducted by BMKG during the period 2009-2024 show an increasing trend of earthquake activity significantly since 2013. The implications of this increase in earthquake activity will certainly have an impact on increasing the potential for earthquake disasters.

Earthquakes can cause significant ground motion. When an earthquake occurs, the resulting seismic waves cause ground motion that can result in landslides, liquefaction, or even deformation of the land surface. This is certainly very dangerous as it can damage buildings and threaten the safety of human life. One of the ground motions caused by earthquakes is Peak Ground Acceleration (PGA).

Peak Ground Acceleration (PGA) is a measure of the maximum ground acceleration that occurs during an earthquake. PGA is a very important factor that must be considered at every construction site to assess the potential damage that can be caused by an

Corresponding Author: Sofyan Hadi Rahmawan (sofyan1508@gmail.com)
1 Sofyan Hadi Rahmawan, Master's Program in Informatics, UIN Maulana Malik Ibrahim, sofyan1508@gmail.com
2 Cahyo Crysdian, Master's Program in Informatics, UIN Maulana Malik Ibrahim
3 Sri Harini, Master's Program in Informatics, UIN Maulana Malik Ibrahim

earthquake. Seismograph records from nearby stations can be used as a basis for calculating PGA values, but reliable estimation methods are also useful to provide more detailed information about the characteristics and motion of the earthquake [3]. The parameters considered in determining PGA predictions are earthquake parameters, such as magnitude and hypo-central distance. In addition, the PGA value is also influenced by local site conditions. Different earthquake types and local site conditions will produce different ground motions despite having the same magnitude and distance [4].

Most research on PGA prediction uses conventional attenuation relationships by applying the empirical equations of the Ground Motion Model [5]. Conventional methods of predicting PGA can have limitations, such as reliance on historical data that may not always be representative of current conditions, and possible errors in physical or statistical model assumptions. In addition, the process can require complex calculations and a long time [6], when sufficient data is available all machine learning algorithms provide more accurate estimates than conventional linear regression in predicting ground motion.

With advances in information technology and artificial intelligence (AI), especially in the development of Artificial Neural Networks (ANN), there are new opportunities to improve accuracy and efficiency in PGA prediction. ANNs are algorithms inspired by the workings of the human brain, which can learn complex patterns from big data and perform highly accurate predictions [6]–[9]. This research explores the use of Artificial Neural Network (ANN) algorithms to extract complex and nonlinear behavior from data. The ANN method has its appeal in geoscience studies and has been used in various ground motion prediction studies around the world. [3] demonstrated the application of ANN methods to develop ground motion prediction models (PGA and PGV) based on magnitude, distance from the earthquake source to the site, and shear wave velocity.

The novelty in this research is the use of local data related to the occurrence of earthquakes and geological conditions on the island of Java, namely earthquake magnitude and earthquake hypocenter distance. Geological data is related to local site conditions, namely using shear wave velocity data at a depth of 30 km from the ground surface commonly referred to as Vs30. Therefore, the input variables used are earthquake magnitude, hypocenter distance, and Vs30 with the selected output variable being the PGA value. In this case, the PGA value is obtained from the BMKG accelerograph sensor on the island of Java.

2. Related Works

Research by [6], who created a ground motion prediction equation based on ANN algorithms for shallow crustal earthquakes, is one of the earlier works that is connected to this one. The model was trained using a hybrid approach that combined the Levenberg-Marquardt technique with genetic algorithms. Peak Ground Acceleration (PGA), Peak Ground Velocity (PGV), and Spectral Acceleration (SA) are all predicted by the current model. Moment magnitude (Mw), focal mechanism (F), shear wave velocity in the area (Vs30), and closest distance to the fault plane (Rrup) are the input parameters for prediction. The findings demonstrate that the ANN algorithm may be employed as a PGA predicting algorithm with an accuracy of 94.9%. It was discovered in this investigation to be similar to linkages that already existed in the global database. The developed ANN model can capture the salient characteristics with GMPE and can estimate the site-specific response spectrum for Shimla City located in the Himalayan region.

A ground motion prediction model (PGA and PGV) based on magnitude, distance from the earthquake source to the location, and shear wave velocity in North, Central, and East America was also developed by [3] using artificial neural networks (ANN). Learning this model yielded a 90% PGA prediction accuracy and a 92% PGV prediction accuracy.

According to the study's findings, the ground motion prediction model created with the ANN model predicts ground motion intensity measurements at all distances and magnitudes with significantly greater accuracy. To forecast ground motion (PGA) based on source characteristics, source-to-site distance, and local site circumstances, [3] also contrasted machine learning techniques with traditional linear regression. The results show that when sufficient data is available all machine learning algorithms provide more accurate estimates than conventional linear regression. Among machine learning algorithms, Random Forest outperformed ANN and SVM with an accuracy of 95.8%.

Utilizing a gradient boosting framework and decision tree-based ensemble machine learning algorithm, the XGBoost method [7] showed a remarkable 99.4% accuracy in predicting PGA in India and Nepal using the following input parameters: Magnitude, focal depth, epicenter distance, and Vs30. An analysis of observed PGA data from the earthquakes in India and Nepal using XGBoost supervised machine learning modeling showed that ML modeling may greatly enhance PGA value prediction and provide much better GMPE for any region. Building more earthquake-resistant structures, lowering the number of fatalities from earthquakes, and creating earthquake-resilient communities in India and elsewhere will all begin with machine learning modeling. With the abundance of ground acceleration data accessible in tectonically active locations like California, machine learning techniques for creating Ground Motion Models (GMMs) are becoming more and more popular. Complex linear and/or nonlinear trends in high-dimensional data can be learned using these entirely data-driven machine-learning algorithms. A branch of artificial intelligence called machine learning enables computers to recognize patterns and relationships in data without the need for a predetermined functional form.

3. Proposed Method

This research uses secondary data obtained from the Meteorology Climatology and Geophysics Agency (BMKG) and the United States Geological Survey (USGS). The data used in this study are earthquake parameter data, local site conditions, and PGA. Earthquake parameter data consists of magnitude and hypocenter distance. Local site data uses shear wave velocity data (Vs30). The magnitude, hypocenter distance, and PGA data used are sourced from BMKG as much as 1828 data derived from 137 earthquake events felt on the island of Java during the period August 2020 to January 2024. Meanwhile, shear wave velocity data at a depth of 30 km or Vs30 is sourced from the USGS.

Inspired by how the human brain functions, artificial neural networks (ANN) are information processing systems that employ weight adjustments to learn and solve problems. An input layer, a hidden layer, and an output layer make up an ANN. Data, features, and signals from the outside world must be received by the input layer. Neurons in the buried layer perform the majority of internal processing and create patterns. Neurons at the output layer are also responsible for delivering the information or signals that are the outcome of the network's processing.

[3], [10] stated that 1 hidden layer is enough to produce output that meets the target. In addition, [11], [12], and [13] also used the I hidden layer in researching PGA prediction. Therefore, the architecture of the ANN model in this study is designed using a 3-layer Neural Network, which consists of 1 input layer, 1 hidden layer, and 1 output layer. This research uses three nodes in the input layer because there are three variables used as input data, namely magnitude, epicenter distance, and Vs30 velocity [2], [5], [10]. The output layer uses one node because the predicted target is only one variable, namely Peak Ground Acceleration (PGA). The architecture of the ANN model used in this study can be seen in Figure 1.

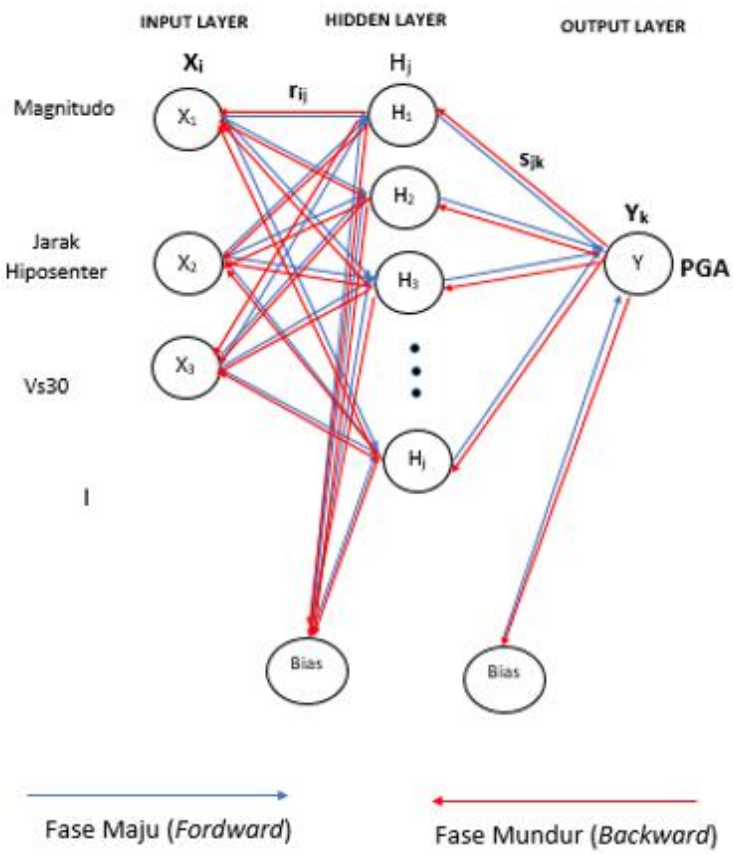

Figure 1. ANN Design

In theory, the ANN method has no standardized rules in determining the optimal architecture, so the determination of a network is done by trial and error to determine the maximum results. On that basis, this research tries to experiment by varying the composition of training data and test data 70%: 30 %, 80% : 20% and 90%: 10%. For the selection of the number of nodes in the hidden layer is done based on the references of previous researchers [11], [13], [14] related to the prediction of PGA using the ANN method and also varies the number of nodes. Details are shown in Table 1.

Table 1. Neural Network Test Architecture Design

A 70% : 30%	**3-4-1**	**3 input – 4 *node hidden layer* 1 – 1 output**	NN-A1
	3-5-1	3 input – 5 *node hidden layer* 1 – 1 output	NN-A2
	3-10-1	3 input – 10 *node hidden layer* 1 – 1 output	NN-A3
	3-15-1	**3 input – 15 *node hidden layer* 1 – 1 output**	NN-A4
	3-20-1	3 input – 20 *node hidden layer* 1 – 1 output	NN-A5
B 80% : 20%	**3-4-1**	**3 input – 4 *node hidden layer* 1 – 1 output**	NN-B1
	3-5-1	3 input – 5 *node hidden layer* 1 – 1 output	NN-B2
	3-10-1	3 input – 10 *node hidden layer* 1 – 1 output	NN-B3
	3-15-1	**3 input – 15 *node hidden layer* 1 – 1 output**	NN-B4
	3-20-1	3 input – 20 *node hidden layer* 1 – 1 output	NN-B5
C 90% : 10%	**3-4-1**	**3 input – 4 *node hidden layer* 1 – 1 output**	NN-C1
	3-5-1	3 input – 5 *node hidden layer* 1 – 1 output	NN-C2
	3-10-1	3 input – 10 *node hidden layer* 1 – 1 output	NN-C3
	3-15-1	**3 input – 15 *node hidden layer* 1 – 1 output**	NN-C4
	3-20-1	3 input – 20 *node hidden layer* 1 – 1 output	NN-C5

The ANN design in this study consists of input data (Xi) in the form of magnitude data, hypocenter distance and velocity Vs30, the weighting value between the input layer to the first hidden layer (rij), the output result in the first hidden layer whose number of nodes has been determined (Hj), the weighting value between the first hidden layer to the output (sjk), and the output value in the form of PGA value (Yk) [10], [15]. The training process uses Mean Square Error (MSE) as a process termination [1], [9], [16], [17] and final weight to get the optimal weight. This means that the training process will stop if the convergence time has been reached. The following are the stages in predicting PGA with the ANN method [4], [9], [11], [13], [16].

4. Experimental Setup

2.1 The Forward Propagation Phase

a. Calculate the input signal at each hidden layer (Hj). In this phase the first information is given to the input variable (Xi) which will forward the input signal to all hidden layers.

$$H_{net\,j} = r_{0j} + \sum_{i=1}^{n} X_i r_{ij} \qquad (1)$$

Where i is the i-th node (i = 1, 2, ..., n) in the input layer and j is the j-th node (j = 1, 2, ..., p) in the hidden layer. X_i is the input value in the i-th input node. Meanwhile, r_0j is the bias in the input layer and r_ij is the weight at the i-th input node that goes to the

j-th hidden node.

b. All outputs in the first hidden layer are calculated using the Rectified Linear Units (ReLU) activation function. The results will be used by each node in the first hidden layer for the next process.

$$H_j = f(H_{net\,j}) = \max(0, H_{net\,j}) \quad (2)$$

The result of the calculation f(〖H_net〗 _j) is the activation value at the jth hidden node to be sent to all output nodes.

c. After obtaining the value of each node in the hidden layer, then add up all the signals that enter the output (H_j ...Hn) with their weights.

$$y_{net\ k} = s_{0k} + \sum_{j=1}^{k} H_j s_{jk} \quad (3)$$

s_{0k} is the bias value in the hidden layer and H_j is the result of the activation function coming out of the hidden layer and s_{jk} is the weight at the j[th] hidden node to the k[th] output node *k* (*k* = 1, 2, …, *m*).

d. By using the Rectified Linear Units (ReLU) activation function to calculate the output signal, the output value will be obtained, namely:

$$y_k = f(y_{net\,k}) = \max(0, y_{net\,k}) \quad (4)$$

Ynet k is the result of the activation function, which is the sum of the weights between the hidden nodes and the output nodes.

e. After the output value is generated, the difference with the output target value is calculated using the MSE formula.

$$mse = \frac{1}{n}\sum_{k}^{n}(t_k - y_k)^2 \quad (5)$$

If the convergence time has been reached, the process will stop. Conversely, if the convergence time has not been reached, the backpropagation process is carried out by updating the weights.

2.2 Backward Propagation Fase

a. Calculate the error factor (δ_k) at the output layer (y_k) that receives the target pattern t_k which is then sent to the next layer to be used to calculate the weight and bias correction between the input layer and hidden layer.

$$\delta_k = (t_k - y_k) f'^{(y_{net_k})} \quad (6)$$

where,

$$f'^{(y_{net_k})} = \left(1 - f\left(y_{net_k}\right)\right) f\left(y_{net_k}\right) \quad (7)$$

b. After the error value in the hidden layer and output layer is obtained, then calculate

the error value in the input layer and hidden layer.

$$\delta_j = \delta_{netj} f'^{(H_{netj})} \quad (8)$$

Where,

$$\delta_{netj} = \sum_{k=1}^{m} \delta_k s_{jk} \quad (9)$$

$$f'^{(H_{netj})} = \left(1 - f(H_{netj})\right) f(H_{netj}) \quad (10)$$

c. After the error value of each layer is known, the amount of change or correction that will be added to the old weight can be calculated.

$$\Delta r_{ij} = \alpha \delta_j X_i \quad (11)$$

$$\Delta s_{jk} = \alpha \delta_k H_j \quad (12)$$

d. The original weights will be modified with the addition of the modifications ∆rij and ∆sjk. The learning rate is represented by α, the error value between the input and hidden layers is represented by δj, and the error value between the hidden and output layers is represented by δ_k. Hj is the outcome of the hidden layer's activation function, whereas Xi is the input that travels from node I to node J.

2.3 Weight Change Phase

After ∆rij and ∆sjk are obtained the next step is to fix the old weights into new weights.

$$r_{ij}(new) = r_{ij}(old) + \Delta r_{ij} \quad (13)$$

$$s_{jk}(new) = s_{jk}(old) + \Delta s_{jk} \quad (14)$$

5. Result and Analysis

Input and output data have different units, so normalization needs to be done first [4], [5], [18]. The normalization method used is the min-max method. This process is done using python software on Google Colab. After normalizing the data, the next step is to implement the ANN model, which is also done using python software on Google Colab. This ANN model uses three layers, consisting of one input layer, one hidden layer, and one output layer. The hidden layer uses five variations in the number of nodes. In the evaluation process, network optimization is processed based on the smallest MSE value during the learning process. The training parameters used can be seen in Table 2.

Table 2. ANN Model Parameters

Network type	BP-ANN
Activation function	*Rectified Linear Units* (ReLU)
Optimizer function	Adam
Performance	MSE (Mean Square Error)
Number of layer	3

Number of nodes input layer	3
Number of hidden layer nodes	4, 5, 10, 15, 20
Output layer nodes	1
Data composition	70:30, 80:20, 90:10
Learning rate	0.001

The training results of each neural network architecture model are presented in Table 3.

Table 3. Training Outcomes

No	Trainer Name	Number of Iteration	Iteration Time (secon)	MSE Train
1	NN-A1	3720	8.92	5.2757
2	NN-A2	3059	7.88	5.2487
3	NN-A3	2967	6.64	5.2454
4	**NN-A4**	**2746**	**7.62**	**5.1978**
5	NN-A5	2358	6.25	5.2086
6	NN-B1	2985	7.54	5.4449
7	NN-B2	3049	7.08	5.4871
8	NN-B3	2883	8.32	5.6295
9	NN-B4	2184	7.1	5.2556
10	NN-B5	2104	5.42	5.3378
11	NN-C1	3837	10.91	5.2122
12	NN-C2	1608	5.72	5.2315
13	NN-C3	2983	9.34	5.2523
14	NN-C4	2835	8.24	5.2695
15	NN-C5	1722	6.38	5.2202

Based on Table 3, the best network architecture is NN-A4, which has 15 nodes in the hidden layer. The smallest MSE value in the training process is 5.1978. The comparison graph between the error value and iterations in the training process of the best network architecture data with the training name NN-A4, is shown in Figure 4.

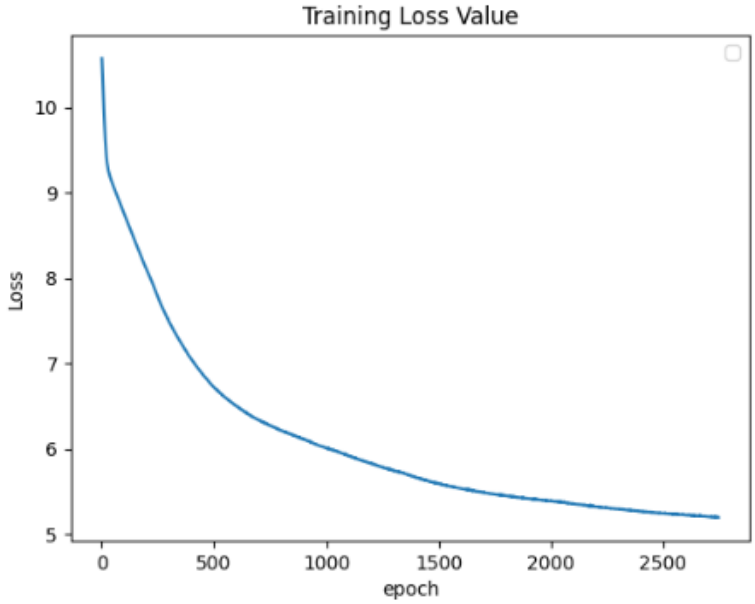

Figure 4. Error Value of NN-A4 Training

Figure 4 is a graph showing the change in error rate during the training process of the NN-A4 network architecture. Where the x-axis represents iterations or steps during the training process, while the y-axis represents the error or loss value. At the beginning of the training the error value is quite high around 10.8. But as the training iteration process increases, the error value continues to decrease with a fairly fast pattern at the beginning of training or until around the 500th iteration and begins to slow down in the middle to the end of the iteration. The training process stops at iteration 2746 because the convergence time has been reached. This process took 7.62 seconds.

The graph of the calculation of the predicted value and its comparison with the actual value on the best network architecture with the training name NN-A4 can be seen in Figure 5.

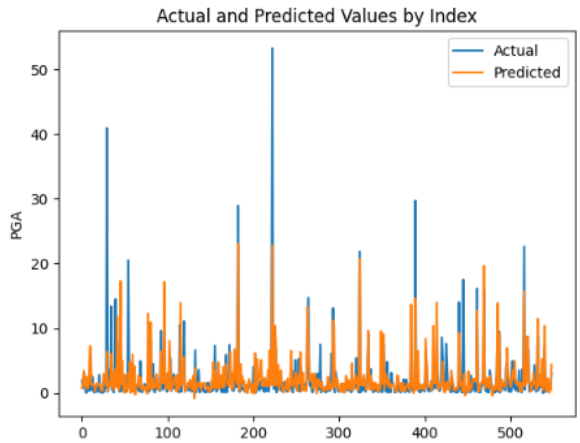

Figure 5. Comparison of predicted and actual values on NN-A4 architecture

From Figure 5 above, it can be seen that the network architecture with the training name NN-A4 predicted values follow the actual data pattern, so this architecture is quite good at predicting Peak Ground Acceleration (PGA) values, especially for predicting PGA

values of less than 20 gal. For predicting PGA values above 20 gal, it is still not very accurate. This is because the PGA data with values above 20 gal are still few, most of the PGA data used are below 20 gal. The accuracy of this architecture has a correlation value of 0.67.

6. Conclusion

Research on PGA prediction is very important. Until now, no technology can predict the occurrence of earthquakes, but what can be done as an effort to minimize the risk of earthquakes is to research PGA prediction. The hope is that with research on PGA prediction, relevant stakeholders can map areas that have the potential for major damage when an earthquake occurs and can reconstruct buildings that are standard and safe against earthquakes. The best ANN network architecture resulting from this research is 3-15-1 with a correlation value of 0.67.

Acknowledgment

The authors would like to thank BMKG for providing the data used in this study.

References

[1] S. Rohadi, R. M. Taruna, A. Rudyanto, and D. T. Heryanto, “Determination of Ground Motion Prediction Equations (Gmpes) Using Euclidean and Likelihood Methods for East Java Region,” *J. Meteorol. dan Geofis.*, vol. 17, no. 3, pp. 177–190, 2018, doi: 10.31172/jmg.v17i3.357.

[2] Y. Bock, “Crustal motion in Indonesia from Global Positioning System measurements,” *J. Geophys. Res.*, vol. 108, no. B8, 2003, doi: 10.1029/2001jb000324.

[3] F. Khosravikia, Y. Zeinali, Z. Nagy, P. Clayton, and E. M. Rathje, “Neural Network-Based Equations for Predicting PGA and PGV in Texas, Oklahoma, and Kansas,” no. August, pp. 538–549, 2018, doi: 10.1061/9780784481462.052.

[4] R. Ulusay, E. Tuncay, H. Sonmez, and C. Gokceoglu, “An attenuation relationship based on Turkish strong motion data and iso-acceleration map of Turkey,” *Eng. Geol.*, vol. 74, no. 3–4, pp. 265–291, 2004, doi: 10.1016/j.enggeo.2004.04.002.

[5] R. Sun, W. Qi, T. Zheng, and J. Qi, “Explainable Machine-Learning Predictions for Peak Ground Acceleration,” *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074530.

[6] A. A. ABIDIN, H. Hamzah, and M. Endah, “Efficient Fruits Classification Using Convolutional Neural Network,” *Int. J. Informatics Comput.*, vol. 3, no. 1, p. 1, 2021, doi: 10.35842/ijicom.v3i1.31.

[7] M. E. H., “DeepSkin: Robust Skin Cancer Classification Using Convolutional Neural Network Algorithm,” *Int. J. Informatics Comput.*, vol. 3, no. 2, p. 41, 2022, doi: 10.35842/ijicom.v3i2.40.

[8] M. H. Wathan and M. Aziz, “Establising CNN for Network Intrusion Detection : A Comparative Approach,” vol. 6, no. 1, 2024, doi: 10.35842/ijicom.

[9] J. Pathak, D. K. Paul, and P. N. Godbole, “Ann Based Model for Simulation of Design Earthquake in the Himalayan Region”.

[10] T. Kerh and D. Chu, “Neural networks approach and microtremor measurements in estimating peak ground acceleration due to strong motion,” *Adv. Eng. Softw.*, vol. 33,

no. 11–12, pp. 733–742, 2002, doi: 10.1016/S0965-9978(02)00081-9.
[11] J. Dhanya and S. T. G. Raghukanth, “Ground Motion Prediction Model Using Artificial Neural Network,” *Pure Appl. Geophys.*, vol. 175, no. 3, pp. 1035–1064, 2018, doi: 10.1007/s00024-017-1751-3.
[12] R. S. Ahluwalia and S. Chidambaram, “Proteomic pattern analysis using neural networks,” *Int. J. Ind. Eng. Theory Appl. Pract.*, vol. 15, no. 1, pp. 45–52, 2008.
[13] P. Mandal, “Peak Ground Acceleration Prediction using supervised Machine Learning algorithm for earthquakes of Mw5.6-7.9 occurring in India and Nepal,” 2022, [Online]. Available: https://doi.org/10.21203/rs.3.rs-1806354/v1
[14] H. Hong, T. Liu, and C. S. Lee, “Observations on the application of artificial neural network to predicting ground motion measures,” *Earthq. Sci.*, vol. 25, no. 2, pp. 161–175, 2012, doi: 10.1007/s11589-012-0843-5.
[15] S. Biswas, D. Kumar, M. Nas, M. Softa, E. Akgün, and U. K. Bera, “Performance of a Five-Layer ANN Model for Earthquake Magnitude Prediction and Spatial Risk Mapping in Turkey,” *Decis. Mak. Adv.*, vol. 3, no. 1, pp. 40–49, 2025, doi: 10.31181/dma31202553.
[16] M. Alizadeh, I. Ngah, M. Hashim, B. Pradhan, and A. B. Pour, “A Hybrid Analytic Network Process and Artificial Neural Network (ANP-ANN) model for urban Earthquake vulnerability assessment,” *Remote Sens.*, vol. 10, no. 6, 2018, doi: 10.3390/rs10060975.
[17] R. Oktarina, S. N. Bahagia, L. Diawati, and K. S. Pribadi, “Artificial neural network for predicting earthquake casualties and damages in Indonesia,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 426, no. 1, 2020, doi: 10.1088/1755-1315/426/1/012156.
[18] F. Khosravikia and P. Clayton, “Machine learning in ground motion prediction,” *Comput. Geosci.*, vol. 148, no. January, p. 104700, 2021, doi: 10.1016/j.cageo.2021.104700.