➔ Regular Research Paper – NS

Evaluation of Clustering Algorithms in Earthquake Magnitude Prediction: An Adaptive Neuro-Fuzzy Inference System-Based Study

Burak Surucu
Graduate School of Natural and Applied Sciences, Department of Civil Engineering
Suleyman Demirel University, Turkey
d2240101001@ogr.sdu.edu.tr

Soner Uzundurukan
Faculty of Engineering and Natural Sciences, Department of Civil Engineering
Suleyman Demirel University, Turkey
soneruzundurukan@sdu.edu.tr

Abstract

Earthquakes are natural disasters that can cause significant loss of life and property, and accurate prediction of earthquake parameters is of great importance in earthquake engineering. This study investigates the effect of different clustering algorithms on Adaptive Neuro-Fuzzy Inference System (ANFIS)-based models in predicting earthquake magnitude. The study used 374 earthquake records obtained from the AFAD Turkey Acceleration Database and Analysis System (TADAS). The dataset was divided into 340 training and 34 test data. Model inputs were defined as year, earthquake depth, and peak ground acceleration (PGA), and the model output was moment magnitude (Mw). K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Model (GMM) algorithms were used to determine the data structure. The performance of the generated ANFIS models was evaluated using R^2, RMSE, and MAE metrics. The results show that model configurations created with GMM-based clustering provide higher prediction accuracy.

Keywords: *Earthquake magnitude prediction, ANFIS, Clustering algorithms, Machine learning*

1. INTRODUCTION

Earthquakes are among the destructive natural disasters that can cause critical loss of life and property over large areas in a short time (Kanamori, 1977). Therefore, interpreting earthquake behavior and accurately estimating earthquake parameters are crucial for earthquake engineering studies (Shearer, 2019). Earthquake magnitude is a fundamental parameter that expresses the magnitude of the energy released at the source of an earthquake and plays a significant role in assessing earthquake hazard and conducting damage analyses after an earthquake (Hanks and Kanamori, 1979). Turkey is significantly at risk of earthquakes due to its location on active fault lines (Bozkurt, 2001). Earthquakes occurring particularly in active tectonic zones such as Western Anatolia and the Aegean Region demonstrate the seismically dynamic nature of the region (Şengör et al., 1985). Therefore, analyzing earthquake data and modeling earthquake parameters using data-based methods is valuable for both academic studies and applied earthquake engineering (Akkar et al., 2010). Today, earthquake records are available through national and international databases such as AFAD, ISC, and these datasets are widely used in earthquake prediction studies (Nurlu et al., 2014).

Earthquakes involve complex and often nonlinear relationships between temporal, spatial, and physical parameters. This can sometimes limit the effectiveness of statistical methods (Riddell, 1979). Recently, advances in artificial intelligence have led to widespread data-driven approaches in areas such as earthquake magnitude estimation, ground motion analysis, and seismic hazard assessment (Kong et al., 2019; Mousavi et al., 2020). Machine learning methods are particularly important in uncovering hidden patterns and modeling nonlinear relationships, especially in large datasets (Bishop, 2006). Among artificial intelligence-based methods, fuzzy logic systems and artificial neural networks are frequently used techniques in solving complex problems involving uncertainty (Azar, 2010). Combining the strengths of these two approaches, the Adaptive Neural Network-Fuzzy Inference System (ANFIS) emerged as a hybrid model that combines the interpretable nature of fuzzy logic with the learning ability of artificial neural networks (Jang, 1993). The ANFIS model is widely used in the analysis of engineering problems due to its ability to learn nonlinear relationships between input variables (Cabalar et al., 2012).

Several studies using artificial intelligence and fuzzy logic-based methods in earthquake magnitude estimation are available in the literature. One such study was conducted by Mirrashid (2014). In this research, the Adaptive Neural Network-Fuzzy Inference System (ANFIS) and the fuzzy C-means clustering algorithm were used together with earthquake data from 1950–2013. The study specifically aimed to develop a model for predicting earthquakes with magnitudes of 5.5 and above. The results obtained showed that the ANFIS-based approach can provide successful results in earthquake magnitude estimation (Mirrashid, 2014).

In another study by Pandit and Biswal (2019), an ANFIS-based modeling approach was used for estimating earthquake magnitude. The study evaluated different clustering methods for creating the ANFIS model and stated that the subtractive clustering method, in particular, yielded more successful results in estimating earthquake magnitude compared to other methods. The study shows that the ANFIS model is sensitive to the structure of the dataset and that the appropriate clustering approach can significantly affect model performance (Pandit & Biswal, 2019).

In a study conducted by Kamath and Kamat (2017), an ANFIS-based earthquake magnitude prediction model was developed using earthquake data from the Andaman-Nicobar Islands region. The study reported that successful results in earthquake magnitude prediction were obtained through a hybrid model created by combining artificial neural networks and fuzzy inference systems (Kamath and Kamat, 2017).

Clustering algorithms, which are part of data preprocessing, allow for a better understanding of the data structure by grouping similar examples in the dataset (Xu and Tian, 2015). Clustering techniques can reveal natural groups within the dataset and create more meaningful data subsets during modeling (Jain, 2010). In the literature, different clustering algorithms such as K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Model (GMM) are widely used in data analysis and pattern recognition studies (Bishop, 2006; Ester et al., 1996).

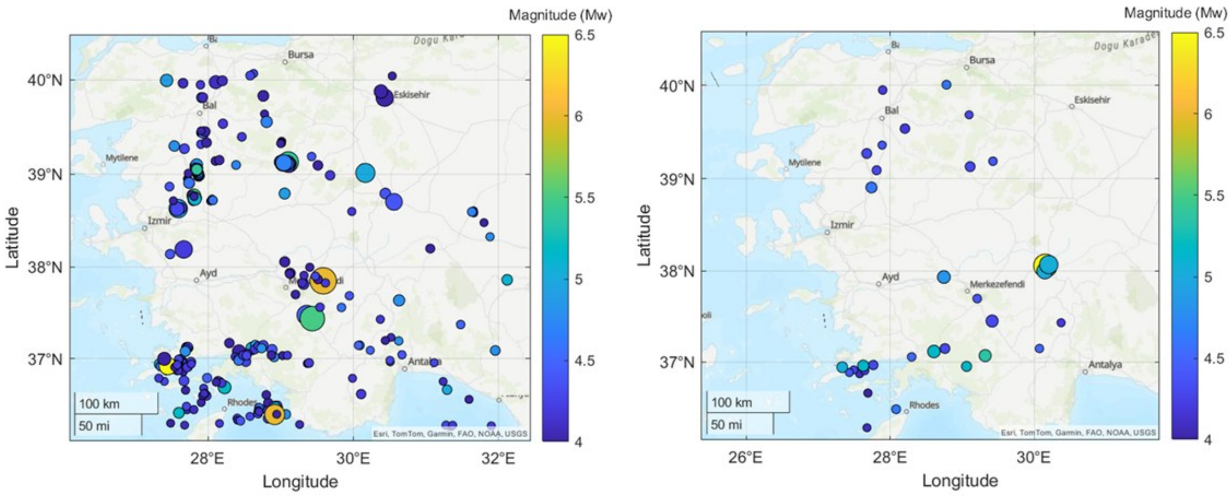
In this study, ANFIS-based models were created using data structures obtained with different clustering algorithms for estimating earthquake magnitude. Within the scope of the study, different clustering structures of the dataset were analyzed using K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Model algorithms. Subsequently, different ANFIS model configurations were prepared with the determined datasets, and the performance of these models was compared. The performance of the developed models was evaluated using common performance metrics such as coefficient of determination (R^2), root mean square error (RMSE) and mean absolute error (MAE) (Chicco et al., 2021). Furthermore, the success of the models on both training and test datasets was analyzed together to determine the most successful model structures. Thus, the aim was to determine the effects of different clustering algorithms on ANFIS-based earthquake magnitude prediction models.

2. MATERIAL AND METHODS

2.1. Data Set and Earthquake Parameters

In this study, two separate datasets were prepared for estimating earthquake magnitude: one for training and one for testing. Care was taken to select the data from reliable sources and to ensure they covered different years. The training and test datasets consist of a total of 374 earthquake records compiled within the scope of the AFAD Turkey Acceleration Database and Analysis System (AFAD TADAS).

These records are sourced from AFAD, DDA, GDDA, and ISC and include information such as ID, recording institution, location, longitude, latitude, depth, date, peak ground acceleration (PGA), and moment magnitude (Mw). The geographical distribution and basic seismological characteristics of the records in the dataset are presented in Figure 1 and Table 1 (AFAD, 2025).



(a) (b)

Figure 1.(a) Geographic distribution of earthquake records for Train Data (AFAD, 2025)
Figure 1.(b) Geographic distribution of earthquake records for Test Data (AFAD, 2025)

The earthquakes included in the dataset mainly occurred in Western Anatolia, the Aegean Region, and its surrounding areas. It also includes some events around the Aegean Sea and the island of Rhodes. This allows the dataset to represent both terrestrial and marine seismic sources. The training dataset consists of a total of 340 earthquake records. The training data utilizes information on year, depth, peak ground acceleration (PGA), and earthquake magnitude. The training records cover events between 1994 and 2024 and consist of earthquakes with varying magnitudes and acceleration levels. In the training dataset, earthquake magnitudes range from approximately 4.0 to 6.5 Mw. However, in the test data presented in Table 1, earthquake magnitudes range from 4.0 to 6.4 Mw, and depth values range from approximately 2.3 km to 47.38 km. Similarly, the variation of PGA values between 1.02 cm/s^2 and 320.76 cm/s^2 indicates that the dataset includes both low and high ground motion levels.

During the modeling process, independent variables were considered as input parameters defining the earthquake, while earthquake magnitude was treated as the output variable to be predicted. In this context, date, depth, and PGA (Peak Ground Acceleration) variables constitute model inputs; moment magnitude (Mw) constitutes the model output. In earthquake magnitude estimation, the temporal and physical diversity of the dataset is a crucial factor affecting model success.

The training and test datasets used in this study provide significant representation by including events from different years, at different depths, and with varying ground motion magnitudes. Specifically, separating the test dataset as an independent validation group allows for the evaluation not only of the ANFIS-based models fit to the training data but also their generalization ability. In this way, model performance can be assessed as more reliable in terms of both learning success and predictive ability on new data.

Table 1. Earthquake Characteristics of Test Data (AFAD TADAS, 2025)

Test No	Earthquake ID	Recording Institute	Location	Long. (°)	Lat. (°)	Depth (km)	Year	PGA (cm/s^2)	Mag. (Mw)
1	609812	AFAD	Ege Sea (Mugla)	27.676	36.268	6.22	2023	1.02	4.1
2	534739	AFAD	Bigadic (Balikesir)	28.203	39.537	7.01	2022	19.08	4.2
3	528829	AFAD	Soma (Manisa)	27.6728	39.2713	9.95	2022	23.47	4.3
4	513530	AFAD	Bucak (Burdur)	30.3698	37.432	11.74	2021	3.64	4.2
5	501844	AFAD	Ege Sea (Mugla)	27.6886	36.654	47.38	2021	3.53	4
6	471616	DDA	Kirkagac (Manisa)	27.8096	39.0911	6.99	2020	13.66	4.3
7	435489	DDA	Acıpayam (Denizli)	29.4123	37.4505	7	2019	57.29	4.3
8	420303	DDA	Gokova Gulf	27.7633	36.9653	6.93	2018	17.49	4.3
9	413846	DDA	Gokova Gulf	27.5743	36.8686	13.55	2018	2.57	4.1
10	387643	AFAD	Gokova Gulf	27.645	36.9051	9.12	2017	5.75	4.2
11	348625	AFAD	Akhisar (Manisa)	27.7448	38.9023	17.24	2016	24.7	4.5
12	348611	AFAD	Akhisar (Manisa)	27.737	38.906	24.27	2016	38.67	4.6
13	312185	AFAD	Simav (Kutahya)	29.1071	39.1295	6.1	2015	24.4	4.3
14	289370	AFAD	Korkuteli (Antalya)	30.069	37.1475	11.86	2014	6.02	4.5
15	279316	AFAD	Koycegiz (Mugla)	28.758	37.1473	10.34	2014	23.75	4.3
16	201698	AFAD	Emet (Kutahya)	29.4238	39.188	20.08	2013	8.19	4.4
17	169361	AFAD	Manyas (Balikesir)	27.8937	39.9482	19.79	2012	7.96	4.2
18	261646	GDDA	Buharkent (Aydin)	28.743	37.933	11.1	2006	66.79	4.8
19	264336	GDDA	Ula (Mugla)	28.2954	37.055	24.4	2006	7.74	4.5
20	260536	GDDA	Rodos Island	28.08	36.4741	26	2005	12.82	4.6
21	263512	GDDA	Harmancik (Bursa)	29.0928	39.6844	4.9	2005	1.37	4.3
22	259659	ISC	Honaz (Denizli)	29.2057	37.6969	11.7	2004	8.33	4.4
23	237386	ISC	Kemalpasa (Bursa)	28.7787	40.005	13.6	2003	11.61	4.6
24	237179	ISC	Ege Sea	27.4929	36.9116	6.3	2002	1.37	4.3
25	237178	ISC	Ege Sea	27.4317	36.8821	2.3	2002	1.89	4.4
26	243276	ISC	Kirkagac (Manisa)	27.886	39.361	6	2001	4.24	4.4
27	-	ISC	Dalaman (Mugla)	29.0582	36.9499	25	1994	25.54	5.1
28	385714	AFAD	Gokova Gulf	27.6236	36.9576	11.03	2017	43.69	5.1
29	240861	ISC	Civril (Denizli)	30.1515	38.0561	5	1995	320.76	6.4
30	396950	AFAD	Ula (Mugla)	28.6045	37.1146	24.46	2017	64.19	5.1
31	381868	AFAD	Ege Sea (Mugla)	27.332	36.941	37.5	2017	36.01	5
32	240904	ISC	Dinar (Afyon)	30.1438	38.0002	11.8	1995	128.84	5

33	240849	ISC	Kiziloren (Afyon)	30.1989	38.0685	11.3	1995	180.38	5
34	59098	GDDA	Denizli	29.3165	37.069	28.9	2007	56.6	5.3

2.2. Clustering Algorithms

2.2.1. K-Means Clustering Algorithm

The K-Means algorithm was first introduced by Stuart Lloyd in 1957. This algorithm aims to minimize the distances between the data points and the centroid of each cluster by dividing the data into a certain number of clusters (k) (Maheswaran, 2013). Lloyd's recommended algorithm has been frequently used in machine learning and data mining over time. In the first step of the algorithm, K centroids are determined (Noor and Mustafa, 2024). These centroids form the starting points of the clusters. In the second step, each data point is assigned to the nearest centroid. Here; x is the data point, c_i is the centroid, n is the number of features of the data point, and j indicates the difference made for each feature.

$$d(x, c_i) = \sqrt{\sum_{j=1}^{n} (x_j - c_{i,j})^2}$$

A new centroid is calculated from the average of the data points assigned to each set. This value represents the average of all data points in that set. Here, c_i^{new} is the new centroid, $[S_i]$ is the set, S_i is the number of elements, and $x \in S_i$ is each data point in the set.

$$c_i^{new} = \frac{1}{[S_i]} \sum_{x \in S_i} x$$

The second and third steps are repeated until the centroids remain unchanged or a certain number of iterations are reached. When the algorithm is complete, each data point is assigned to the nearest centroid, although the clusters are as different from each other as possible. The ease of understanding and implementation of the K-means algorithm, the rapid availability of results, and its ability to work efficiently and with diverse datasets, even in large datasets, constitute its positive aspects (Noor and Mustafa, 2024). However, the necessity of estimating the number of clusters beforehand, the ability to reach a local optimum solution based on initial conditions, and the possibility of outliers causing changes in the position of the cluster center, thus leading to incorrect results, can occur in the K-means algorithm (Bishop, 2006).

2.2.2. Hierarchical Clustering Algorithm

Hierarchical clustering algorithms are unsupervised learning methods used to classify data points in a hierarchical structure. This algorithm initially treats each data point as its own cluster (Murtagh and Contreras, 2012). Then, it merges clusters to create larger clusters. The clustering process generally follows an agglomerative (bottom-up) approach, merging the two closest clusters at each step. The distance between data points can be calculated using different criteria (Mimmack et al., 2001). Euclidean distance is a commonly used technique and is calculated using the formula defined below.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{i,k} - x_{j,k})^2}$$

Here, x_i and x_j represent data points, and m represents the number of features for any given data point. Cluster joining criteria are generally determined using techniques such as single linkage, complete linkage, or average linkage. The single linkage is defined below.

$$d(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$$

Another method used to specify the distance between clusters is complete linkage, which can be expressed as follows.

$$d(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$$

The merging process is visualized using a tree structure called a dendrogram. Dendrograms allow us to determine which cluster merge is stronger based on the order of cluster merging. In this way, the number of clusters in a dataset can be obtained by selecting groups by cutting the heights in the dendrogram (Nielsen, 2016).

2.2.3. DBSCAN Clustering Algorithm

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithm is used to group data based on density. This algorithm is particularly effective in detecting noise or outliers. DBSCAN determines clusters by evaluating the number of neighbors and density of each data point. The basic parameters are ε and MinPts. ε represents the distance between a point and its neighbors, while MinPts represents the minimum number of neighbors required for a point to form a cluster (Deng, 2020).

A point is considered a "kernel point" if it has at least MinPts of neighbors within a distance ε. Relationships between kernel points form sets. If a point does not have enough neighbors within a distance ε but is a neighbor of a kernel point, it is considered a "boundary point". Outlier points are points that have no kernel neighbors and are not included in a set. Euclidean distance is generally used to determine the neighborhood of a point (Harchaoui et al., 2008).

$$dist(p, q) \leq \varepsilon$$

Here, dist(p,q) represents the distance between points p and q. For a point p to form a cluster, it must satisfy two conditions. First, if point p has at least MinPts of neighbors at a distance ε, then p is a kernel point. Second, if point p is a neighbor of a kernel point and p is considered a boundary point, then p is evaluated within this cluster. DBSCAN detects natural clusters based on the density structure of the data and is robust against noise points (Ienco and Bordogna, 2018).

2.2.4. GMM Clustering Algorithm

The Gaussian Mixture Model (GMM) is a probabilistic clustering method based on modeling data points as a combination of multiple Gaussian distributions. In this approach, the dataset is expressed as a linear combination of K Gaussian components with different mean and covariance parameters (Zhang et al., 2021). Instead of assigning data points to a specific cluster, GMM uses a soft clustering approach that calculates the probabilities of each data point belonging to different clusters (Ferraro and Giordani, 2020). The Gaussian mixture model is expressed by the following probability density function:

$$p(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)$$

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_j | \mu_j, \Sigma_j)}$$

Here, π_k represents the mixing coefficients, μ_k the mean vector, and $\sum_k$ the covariance matrix. Model parameter estimation is typically performed using the Expectation–Maximization (EM) algorithm (Dempster et al., 1977). This algorithm iteratively updates the model parameters by calculating the probabilities of data points belonging to clusters.

$$\pi_k = \frac{N_k}{N}, \qquad \mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{N_k}, \qquad \sum_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{N_k}$$

2.3. ANFIS-Based Modeling

Adaptive Neuro-Fuzzy Inference System (ANFIS) is a hybrid modeling approach that combines the strengths of artificial neural networks and fuzzy logic systems (Salleh et al., 2017). This method leverages both the ability of fuzzy inference systems to model uncertainties and the learning capabilities of artificial neural networks to model complex nonlinear relationships (Jang, 1993). It is widely used, particularly in engineering and earth sciences, for data-driven prediction problems (Saplıoğlu and Uzundurukan, 2019).

The ANFIS architecture is generally based on a Takagi–Sugeno type fuzzy inference system. In this model, fuzzy rules are prepared through membership functions defined for the input variables, and the system output is analyzed as a weighted average of these rules (Walia et al., 2015). The ANFIS architecture consists of five layers: fuzzification layer, rule layer, normalization layer, defuzzification layer, and output layer. These layers work together to learn the nonlinear relationship between the input and output variables (Saplıoğlu and Acar, 2020).

The ANFIS model is typically trained using a hybrid learning algorithm. In this approach, the output parameters are calculated using the least squares method during the feedforward phase, while the parameters of the membership functions are updated using the gradient descent method during the backpropagation phase. This hybrid learning technique allows for faster and more stable learning of the model parameters (Kose and Arslan 2017).

In this study, a Sugeno-type ANFIS model was used to estimate earthquake magnitude. Sugeno-based fuzzy inference systems are widely preferred in data-driven estimation problems (Jang, 1993). In the model structure, membership functions consisting of three subsets are defined for each input variable. The shape of the membership functions is determined using triangular membership functions (trimf). This approach represents data distribution in a simple and interpretable way during the fuzzification process of the input variables. In the output part of the model, Sugeno-type constant result functions are used.

2.4. Data Set Preparation

The earthquake dataset used in this study contains earthquake records from different years, and data cleaning and organization processes have been applied. Homogenization of earthquake magnitudes was performed using moment magnitude (Mw). The dataset was then prepared for the modeling process. Care was taken to ensure that the dataset did not contain missing or inconsistent records and that all parameters could be transferred to the model in numerical form. In the modeling, the dataset was divided into two separate parts: training and test datasets. The training dataset consisted of a total of 340 earthquake records and was used in the learning phase of the model. The test dataset consisted of 34 earthquake records and was used to evaluate the model's performance on independent data. The dataset used in this study consists of seismological parameters that describe earthquake events. These variables were used as input parameters for the model. Earthquake magnitude was considered as the output variable that the model aimed to predict. The dataset includes earthquakes from different geographical regions. This ensures that the dataset used in the study contains earthquake events with different seismic characteristics.

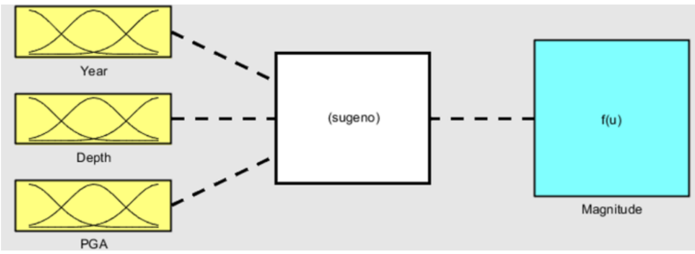


Figure 2. Input/output and definition of the ANFIS model (MATLAB, 2026).

Following this preparation phase, the dataset was formatted appropriately in MATLAB for use in the ANFIS-based modeling process and fed into the model to create the initial rule structure using clustering algorithms. Thus, the necessary data infrastructure was created for both the model's learning process and performance evaluation.

2.5 Performance Metrics

To evaluate the prediction performance of the prepared ANFIS-based models, three different performance metrics were used. These metrics were defined as the coefficient of determination (R^2), root mean square error (RMSE) and mean absolute error (MAE). These metrics are commonly used in prediction model studies to examine the agreement between predicted and actual values (Willmott and Matsuura, 2005; Chai & Draxler, 2014). In this study, the specified performance metrics were calculated separately for both the training and test datasets. Thus, the learning success and test performance of the developed ANFIS models on independent data were investigated.

The coefficient of determination (R^2) measures the strength of the relationship between the values predicted by the model and the actual values. The R^2 value ranges from 0 to 1, with a value closer to 1 indicating a higher predictive performance of the model. R^2 is calculated as follows (Ozer, 1985). Here, y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n represents the total number of data points.

$$R^2 = \left(\frac{\sum_{i=1}^{n} (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2}} \right)^2$$

RMSE is calculated as the root mean square of the errors between the predicted and actual values. This metric is frequently used in evaluating model performance, particularly because of its greater sensitivity to large errors (Chai and Draxler, 2014). RMSE is expressed as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

MAE represents the average of the absolute differences between the predicted and actual values. This metric is a simple and interpretable performance measure used to assess the average magnitude of model prediction errors (Willmott and Matsuura, 2005). MAE is calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |$$

3.FINDINGS

3.1. Examining the Results of Clustering Algorithms

3.1.1. Determining the number of clusters for the K-Means Algorithm

The study initially involved coding the K-means algorithm using MATLAB R2023b. The MATLAB code aims to determine the optimal number of clusters in a dataset by utilizing the Elbow method. First, the K-means algorithm was applied to each column of the data for different cluster numbers between 2 and 7. Accordingly, the total squared error (inertia) was calculated for each cluster number. The resulting error value is equal to the sum of the squares of the distances of each data point from the cluster center, and generally decreases with increasing cluster numbers. The Elbow method was used to find the "elbow" point in this decreasing curve. In other words, the aim was to find the point where the effect of increasing cluster numbers on the error becomes apparent. The inertia values were calculated for each cluster number and displayed on a graph. Then, second-order differences (sharp changes in the slope) were calculated, and the elbow point was found. The cluster number where the Elbow point is found represents the number that best suits the natural structure of the dataset. Finally, K-means clustering was performed with optimal cluster numbers, and the cluster to which each data point belonged was determined.

In the calculations for the K-means algorithm, the optimal cluster number was found to be 6 for all three input data points. Using the Elbow method, it was determined that the decrease in error values slows down as the cluster number increases, and this occurs at a turning point with 6 clusters.

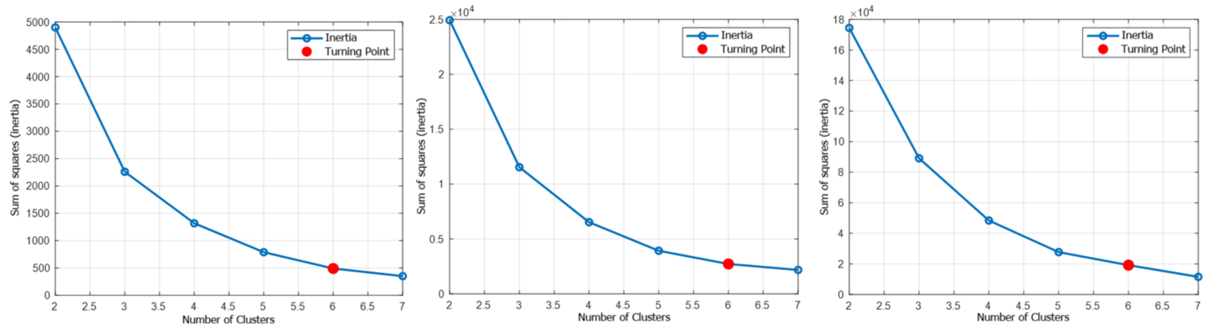


Figure 3. Determining the number of input clusters according to the Elbow method.

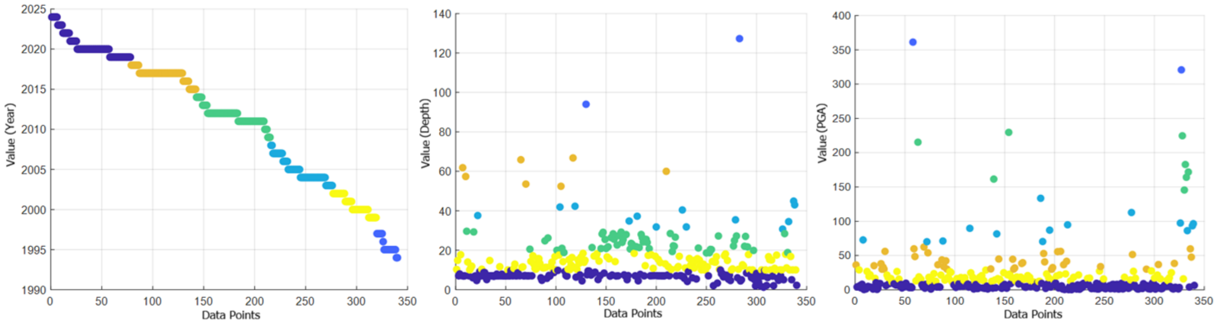


Figure 4. Cluster Representation of Data Points and Values for K-means Clustering Algorithm

3.1.2. Hierarchical Clustering Algorithm

In this study, hierarchical clustering was used as the second algorithm. In the prepared MATLAB code, all columns were analyzed using the hierarchical clustering algorithm, and the optimal number of clusters for each cluster was determined. First, the dataset (X) was defined, and each column was considered independently. The number of clusters was randomly selected between 2 and 7 and used in the clustering process for each column. In the hierarchical clustering algorithm, each data point initially formed its own cluster, and then the closest clusters were merged based on distance. Ward's method was used for this process. In this way, the aim is to minimize the variance within the clusters in each cluster merger and to achieve a balanced structure of the clusters. The linkage matrix in the clustering phase defines the distance and merging of the clusters. Then, a tree structure called a dendrogram was created, representing the order of data points and their merging distances. This structure shows which point the clusters are closer to and where they are beginning to merge. The clustering of data points was achieved according to the randomly determined number of clusters. With a predetermined number of clusters, each data point is assigned to a cluster. Following this process, a visualization of each data point within the cluster is performed. In this way, the clusters and the combinations of data included in each cluster are identified. The results obtained using the hierarchical clustering algorithm show that the data in each column has different structures, resulting in varying numbers of clusters. According to this algorithm, three different analyses were performed, and different results were observed. In the first analysis, 5 clusters were identified for year data, 4 for depth data, and 3 for PGA data. In the second experiment, 6 clusters were selected for year data, 3 for depth data, and 7 for PGA data. In the final analysis, 5 clusters were identified for year data, 6 for depth data, and 3 for PGA data.

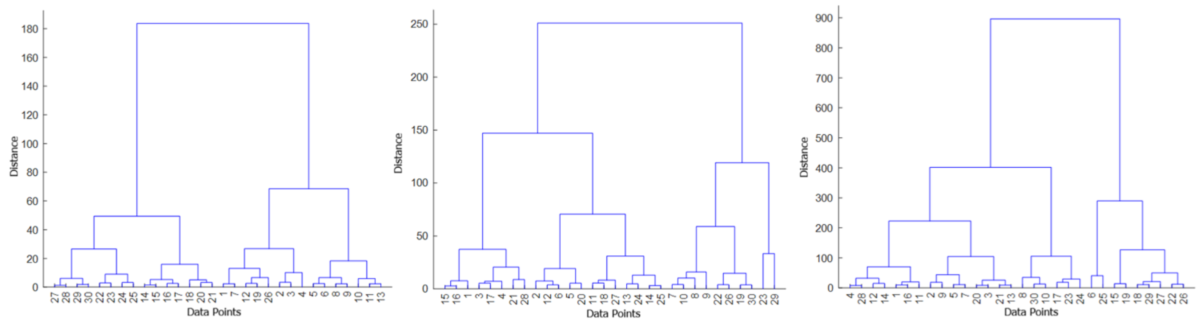


Figure 5. Hierarchical Clustering Dendrogram for Input Data

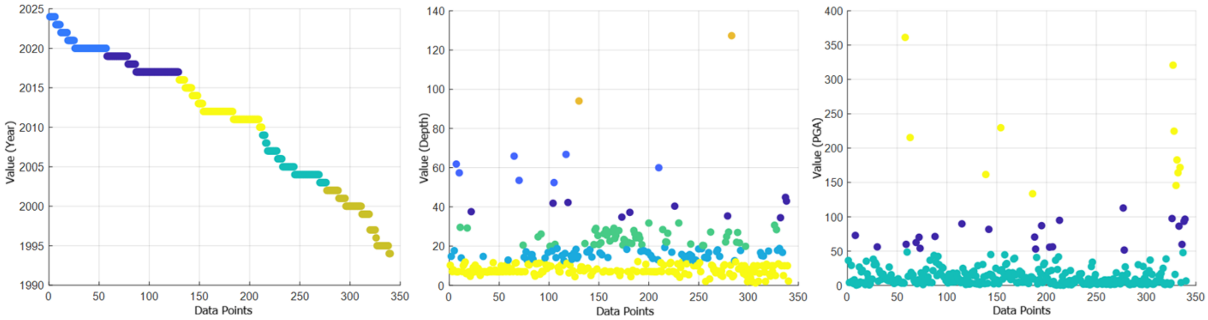


Figure 6. Cluster Representation of Data Points and Values Hierarchical Clustering Algorithm

3.1.3. DBSCAN Clustering Algorithm

In the MATLAB code prepared for the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithm, each column was initially processed independently, and clustering parameters were defined for each. The min_points value, one of the most important parameters of the DBSCAN algorithm, was chosen as 5, while the epsilon (eps) value was determined between 0.1 and 2.0. During the clustering process, the eps value was increased in 0.1 increments to ensure clustering for each value. Because DBSCAN is a density-based algorithm, each data point was assigned to clusters based on the density of its neighbors, and points that did not meet certain criteria were defined as noise.

If noise is not included in the calculations during clustering, the resulting cluster count is calculated, and this number is considered valid between 2 and 7. The most suitable eps value and the corresponding clustering result are recorded. Finally, the number of clusters for each column and which element belongs to which cluster are visualized; This process helps in understanding the natural structure of the data, how clusters are formed, and also in identifying points of noise. As a result of the analyses, 2 clusters were identified for the Year data, 6 clusters for the Depth data, and 7 clusters for the PGA data.

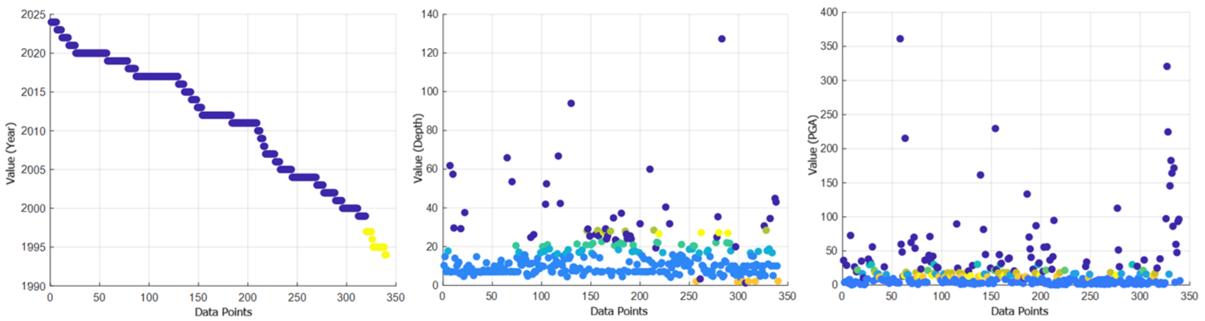


Figure 7. Cluster Representation of Data Points and Values for DBSCAN Clustering Algorithm

3.1.4. GMM Clustering Algorithm

In the GMM (Gaussian Mixture Model) algorithm, the dataset is first normalized using the Z-Score standardization. In this step, the mean of the relevant column is calculated for all data points and divided by its standard deviation. This results in a unit variance, where all data points are zeroed out relative to the mean. Then, the GMM model is trained with between 2 and 7

clusters for each column. While GMM attempts to model each data column with multiple Gaussian distributions, the "fitgmdist" function creates a model for the number of clusters for the columns. This function uses a "full" corvariance type, utilizing a GMM model with a different corvariance structure, and selects starting points using the "plus" method. This allows the clustering algorithm to operate more efficiently and stably. Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values were calculated to test the quality of model fit for all GMM calculations. These criteria express the fit between the model and the data, while also considering the complexity of the model, in other words, the number of parameters. Low AIC and BIC values indicate a more successful model. Models with the lowest AIC and BIC values were preferred in determining the number of clusters, and the cluster function was used to determine which clusters the data points belonged to.
In three trials using the GMM clustering algorithm, the optimal number of clusters for each column was determined. In the first trial, 3 clusters were determined for the year data, 5 for the depth data, and 5 for the PGA data. In the second trial, 7 clusters were determined for the year data, 5 for the depth data, and 5 for the PGA data. In the third trial, GMM determined 7 clusters for the year data, 6 for the depth data, and 4 for the PGA data. The results show that the GMM algorithm generally suggested similar cluster numbers in all three trials and that the number of clusters was determined consistently.

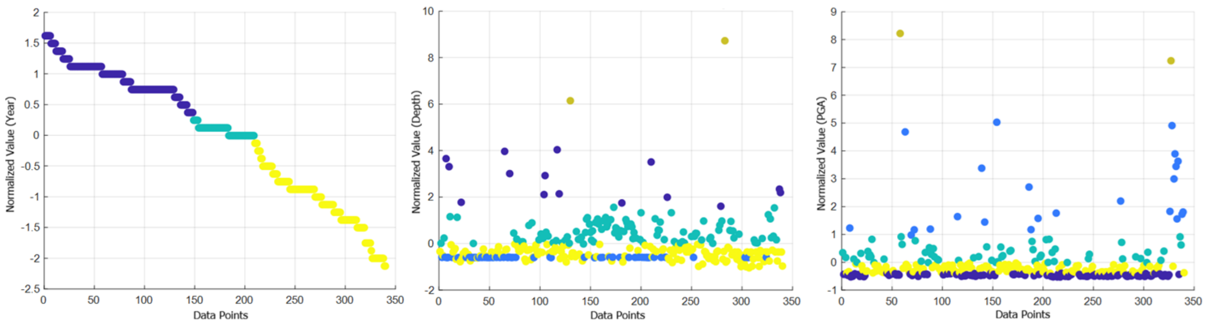


Figure 8. Cluster Representation of Data Points and Values for GMM Clustering Algorithm

3.2. Performance Evaluation and Selection of the Best ANFIS Models

The performance metrics used in this study were chosen to investigate both the model's prediction accuracy and error magnitude. The R^2 value represents the model's explanatory power, while the RMSE and MAE values indicate the magnitude of the prediction error. In this context, a high R^2 value and low RMSE and MAE values indicate more successful model performance.

An examination of the performance results obtained from the training dataset reveals significant differences in accuracy among the models. Specifically, the m355 model stands out with the highest coefficient of determination and the lowest error values on the training data. The calculated R^2 value for this model is approximately 0.9488, while its RMSE and MAE values are quite low. Additionally, the m555 and m444 models also stand out with their high accuracy values and low error metrics. However, the m777 model, which has a higher number of membership functions, performed relatively poorly on the training dataset. This indicates that increasing model complexity does not always lead to better prediction performance.

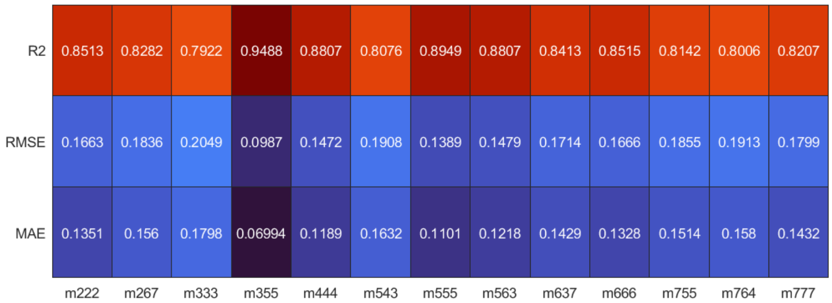


Figure 9. Performance Metrics for Training Data

When the test results are examined, it is seen that the m355 model again has the highest R^2 value. The R^2 value calculated for this model in the test dataset was approximately 0.9117. Furthermore, the RMSE and MAE values are lower compared to other models. Similarly, the m444 and m555 models were also found to have successful results and high prediction accuracy on the test dataset. However, it was observed that some models, despite having high accuracy in the training data, performed worse in the test dataset. This indicates that increasing model complexity can increase the risk of overfitting.

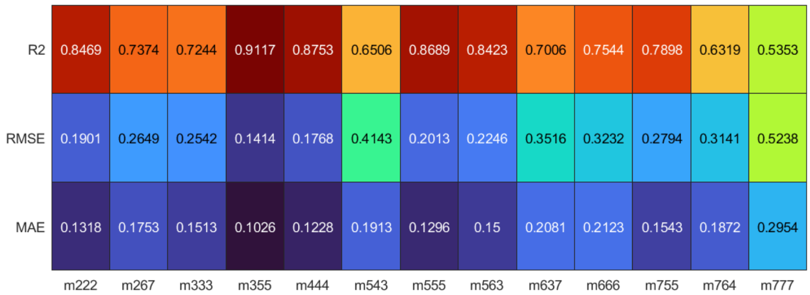


Figure 10. Performance Metrics for Test Data

To comprehensively evaluate the performance of the models, a combined success score was calculated by examining the training and test datasets together. In calculating this score, each performance metric was first scaled using min-max normalization to ensure that performance values at different scales could be compared. Then, R^2, RMSE, and MAE values were combined into a single success score using specific weighting coefficients. In this calculation, a higher weight was given to the R^2 metric, which represents the explanatory power of the model, while RMSE and MAE metrics, which represent the magnitude of the error, were calculated with lower weights. Based on the ranking according to the combined success scores, model m355 was determined to have the highest success score. This model is followed by m555 and m444, respectively. The results show that model structures created with a clustering approach based on the Gaussian Mixture Model (GMM) can provide higher accuracy in earthquake magnitude prediction. However, it was found that the generalization performance of models containing a high number of membership functions decreases, and model complexity can negatively affect prediction accuracy.

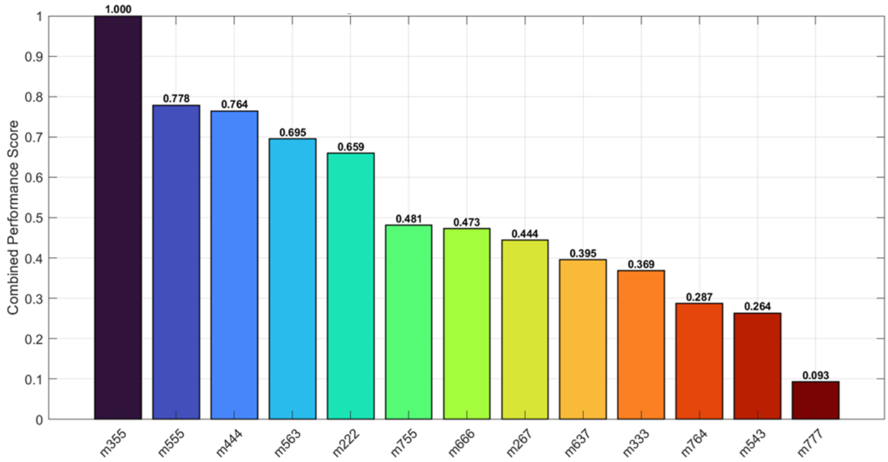


Figure 11. Combined Ranking of Models Based on Training and Test Performance

Figure 12 presents the regression graphs for the training dataset created for the three most successful models. Examination of the graphs reveals a strong correlation between the observed magnitude values and the magnitude values predicted by the model. The concentration of data points largely around the 45° reference line indicates that the models can predict the detected values with high accuracy. In particular, the m355 model shows a low difference between the regression line and the reference line, and exhibits a more regular distribution of data points. Examining the data density, indicated by the color scale, it is noteworthy that data density is higher, especially in the medium magnitude range.

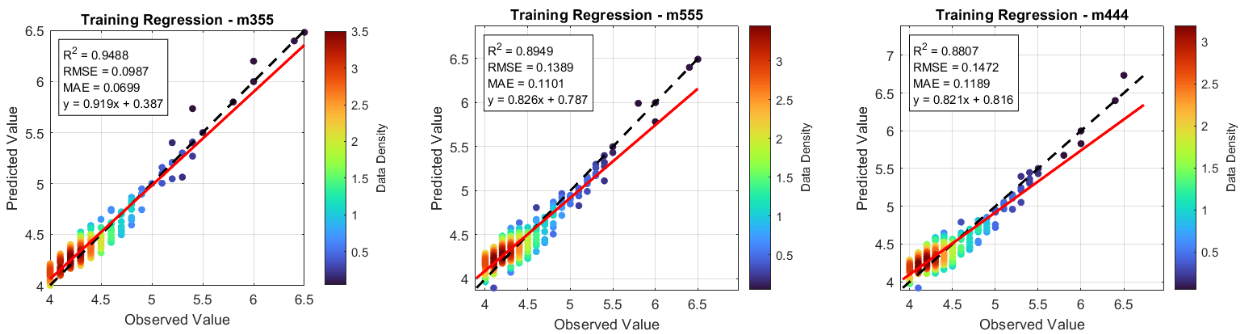


Figure 12. Regression plots for training data from the top three models

Figure 13 shows the regression graphs of the test datasets for the three most successful models. The test dataset is important for evaluating how accurately the models can produce predictions on data they did not see during training. Examining the graphs, it can be seen that the data points are generally distributed around the reference line and that the models can produce consistent predictions on the test data as well. In particular, the high accuracy values produced by the m355 model in the test dataset demonstrate its strong generalization ability. Additionally, it was observed that the m444 and m555 models also successfully represented the relationship between observed and predicted values.

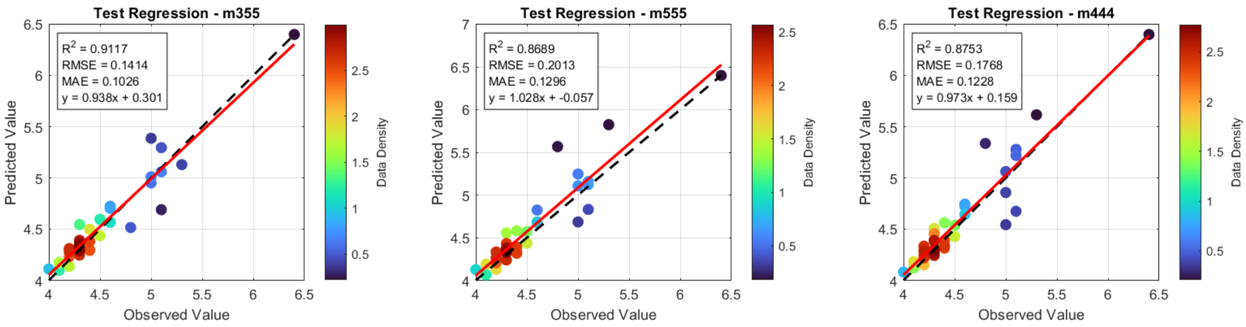


Figure 13. Regression plots for test data from the top three models

Figure 14 presents a comparison of the observed and predicted magnitude values for the three most successful models within the test samples. These graphs show both the actual magnitude values and the predicted values generated by the model for each test sample. Examination of the graphs reveals that the predicted values generally follow the observed magnitude values.

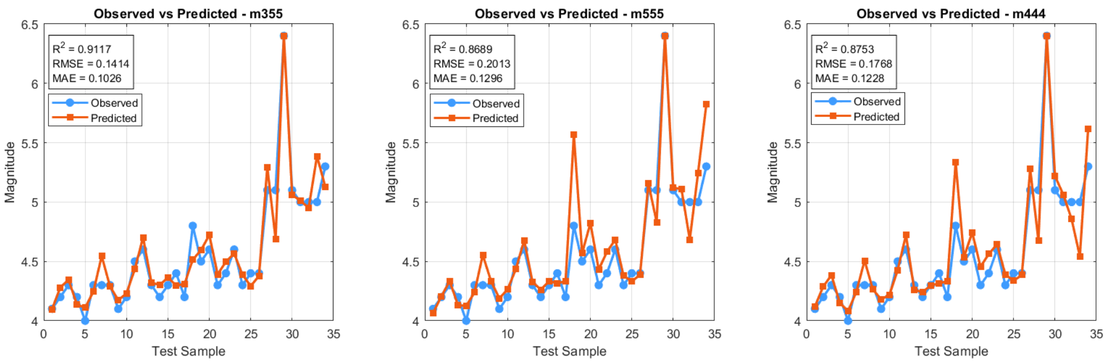


Figure 14. Observed and predicted values of test data for the top three models

It was found that the models produced stable predictions, particularly in the medium magnitude range. While small deviations were observed in some samples, the overall trend shows that the models adequately represent the true values. These findings demonstrate that the proposed ANFIS-based modeling approach is a viable method for earthquake magnitude prediction.

4. RESULTS AND DISCUSSIONS

This study develops an ANFIS-based modeling approach for estimating earthquake magnitude and examines the effects of different clustering algorithms on model performance. In this context, clustering structures of the dataset were determined using K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Model (GMM) algorithms, and various ANFIS model configurations were constructed accordingly. Model performance was evaluated on both training and test datasets using R^2, RMSE, and MAE metrics. The results indicate that clustering algorithms significantly influence ANFIS model performance. In particular, models generated with GMM-based clustering produced more successful results compared to the other methods. Among all configurations, the m355 model achieved the highest accuracy on both training and test datasets, while the m555 and m444 models also showed strong predictive performance. On the other hand, models with a very high number of membership functions exhibited decreased performance, suggesting that increasing model complexity may lead to overfitting.

Regression plots comparing observed and predicted values demonstrate that the best-performing models estimate earthquake magnitude with high accuracy, as most data points are concentrated around the reference line. Overall, the findings confirm that the ANFIS-based modeling approach is effective for earthquake magnitude prediction, and its performance can be improved when combined with appropriate clustering algorithms. Future studies may further enhance prediction accuracy by incorporating additional seismological parameters and alternative artificial intelligence techniques.

5. STATEMENT AND ETHICAL DECLARATION

This study was prepared by benefiting from the thesis titled *"Evaluation of Seismic Hazard Analyses of Isparta Province Using Artificial Intelligence Techniques"* conducted at Suleyman Demirel University. The authors declare that this study was conducted in accordance with ethical research and publication principles.

REFERENCES

[1] Akkar, S., Çağnan, Z., Yenier, E., Erdoğan, Ö., Sandıkkaya, M. A., & Gülkan, P. (2010). The recently compiled Turkish strong motion database: preliminary investigation for seismological parameters. journal of Seismology, 14(3), 457-479.

[2] Azar, A. T. (Ed.). (2010). Fuzzy systems. BoD–Books on Demand.

[3] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

[4] Bozkurt, E. (2001). Neotectonics of Turkey–a synthesis. Geodinamica acta, 14(1-3), 3-30.

[5] Cabalar, A. F., Cevik, A., & Gokceoglu, C. (2012). Some applications of adaptive neuro-fuzzy inference system (ANFIS) in geotechnical engineering. Computers and Geotechnics, 40, 14-33.

[6] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development, 7(3), 1247-1250.

[7] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. Peerj computer science, 7, e623.

[8] Deng, D. (2020, September). DBSCAN clustering algorithm based on density. In 2020 7th international forum on electrical engineering and automation (IFEEA) (pp. 949-953). IEEE.

[9] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B, 39(1), 1–38.

[10] Disaster and Emergency Management Authority (AFAD). (2025). Turkish Strong Motion Database (TADAS). https://tadas.afad.gov.tr

[11] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[12] Ferraro, M. B., & Giordani, P. (2020). Soft clustering. Wiley Interdisciplinary Reviews: Computational Statistics, 12(1), e1480.

[13] Hanks, T. C., & Kanamori, H. (1979). A moment magnitude scale. Journal of Geophysical Research: Solid Earth, 84(B5), 2348-2350.

[14] Harchaoui, Z., Moulines, E., & Bach, F. (2008). Kernel change-point analysis. Advances in neural information processing systems, 21.

[15] Ienco, D., & Bordogna, G. (2018). Fuzzy extensions of the DBScan clustering algorithm. Soft Computing, 22(5), 1719-1730.

[16] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.

[17] Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3), 665-685.

[18] Kamath, R. S., & Kamat, R. K. (2017). Earthquake magnitude prediction for Andaman-Nicobar Islands: Adaptive neuro fuzzy modeling with fuzzy subtractive clustering approach. J. Chem. Pharm. Sci, 10(3), 1228-1233.

[19] Kanamori, H. (1977). The energy release in great earthquakes. Journal of geophysical research, 82(20), 2981-2987.

[20] Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. Seismological research letters, 90(1), 3-14.

[21] Kose, U., & Arslan, A. (2017). Forecasting chaotic time series via anfis supported by vortex optimization algorithm: Applications on electroencephalogram time series. Arabian Journal for Science and Engineering, 42(8), 3103-3114.

[22] Maheswaran, G. (2013). K Means Clustering Algorithms: A Comparitive Study.

[23] Mimmack, G. M., Mason, S. J., & Galpin, J. S. (2001). Choice of distance matrices in cluster analysis: Defining regions. Journal of climate, 14(12), 2790-2797.

[24] Mirrashid, M. (2014). Earthquake magnitude prediction by adaptive neuro-fuzzy inference system (ANFIS) based on fuzzy C-means algorithm. Natural hazards, 74(3), 1577-1593.

[25] Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nature communications, 11(1), 3952.

[26] Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2(1), 86-97.

[27] Nielsen, F. (2016). Hierarchical clustering. In Introduction to HPC with MPI for Data Science (pp. 195-211). Cham: Springer International Publishing.

[28] Noor, S. Q., & Mustafa, T. K. (2024). Comparing K-Means, Nearest Neighbor, and Lloyd's Clustering Algorithms. Iraqi Journal of Science, 6688-6704.

[29] Nurlu, M., Fahjan, Y., Eravcı, B., Baykal, M., Yenilmez, G., Yalçın, D., ... & Pakdamar, F. (2014, August). Rapid estimation of earthquake losses in Turkey using AFAD-RED system. In 2nd European conference on earthquake engineering and seismology, Istanbul.

[30] Ozer, D. J. (1985). Correlation and the coefficient of determination. Psychological bulletin, 97(2), 307.

[31] Pandit, A., & Biswal, K. C. (2019). Prediction of earthquake magnitude using adaptive neuro fuzzy inference system. Earth Science Informatics, 12(4), 513-524.

[32] Riddell, R. (1979). Statistical analysis of the response of nonlinear systems subjected to earthquakes. University of Illinois at Urbana-Champaign.

[33] Salleh, M. N. M., Talpur, N., & Hussain, K. (2017, June). Adaptive neuro-fuzzy inference system: Overview, strengths, limitations, and solutions. In International conference on data mining and big data (pp. 527-535). Cham: Springer International Publishing.

[34] Saplıoğlu, K., & Acar, R. (2020). K-means kümeleme algoritması kullanılarak oluşturulan yapay zeka modelleri ile sediment taşınımının tespiti. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 9(1), 306-322.

[35] Saplıoğlu, K., & Uzundurukan, S. (2019). Bilimsel çalışmalarda kullanılan bazı yapay zeka uygulamalarının ve trendlerinin incelenmesi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(1), 249-262.

[36] Shearer, P. M. (2019). Introduction to seismology. Cambridge university press.

[37] Şengör, A. M. C., Görür, N., & Şaroğlu, F. (1985). Strike-slip faulting and related basin formation in zones of tectonic escape: Turkey as a case study.

[38] The MathWorks Inc. (2026). MATLAB (Version R2023b) [Computer software]. Natick, MA, USA.

[39] Walia, N., Singh, H., & Sharma, A. (2015). ANFIS: Adaptive neuro-fuzzy inference system-a survey. International Journal of Computer Applications, 123(13).

[40] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

[41] Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of data science, 2(2), 165-193.

[42] Zhang, Y., Li, M., Wang, S., Dai, S., Luo, L., Zhu, E., ... & Zhou, H. (2021). Gaussian mixture model clustering with incomplete data. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 17(1s), 1-14.