International Journal of Trend in Research and Development, Volume 13(4), ISSN: 2394-9333
www.ijtrd.com

Comparative Analysis of Machine Learning Models for Global Earthquake Data: MayType Classification and Magnitude Prediction

[1]Gonca Ozata and [2]Yavuz Selim Taspinar,
[1]Department of Management Information System, Selcuk University, Konya, Türkiye
[2]Assoc. Prof., Department of Mechatronic Engineering, Selcuk University, Konya, Türkiye

***Abstract*—** In this research, two fundamental artificial intelligence applications were implemented using data from earthquakes that occurred worldwide between 2000 and 2023. The measurement type of the earthquake magnitude was classified, and the magnitude value was estimated using the regression method. The dataset used consisted of approximately 614,000 observations sourced from the United States Geological Survey. In the study, both classification and regression models were created using Artificial Neural Networks, Random Forest, Support Vector Machines, and k-Nearest Neighbour algorithms. Classification performance was evaluated using accuracy, F1 score, precision, and recall; while regression models were evaluated using metrics such as MAE, MSE, RMSE, and R^2. The findings show that RF and ANN achieved the highest classification performance, with overall accuracy values reaching up to ninety eight percent, whereas the SVM model demonstrated considerably lower performance. In the regression task, the RF model outperformed the other algorithms, achieving the lowest error rates and the highest explanatory power. Additionally, experimental analyses conducted across three different scenarios demonstrated that rearranging class labels in accordance with USGS standards and using feature selection based on Information Gain and Gain Ratio methods significantly increased model performance. These results demonstrate that AI-based methods can provide effective predictions in large-scale, multidimensional seismic datasets and can make valuable contributions, particularly to critical areas such as disaster risk management.

***Keywords*—** *Earthquake; Classification; Regression; Prediction; Analysis*

I. INTRODUCTION

Earthquakes are among the most destructive natural hazards worldwide, frequently causing severe loss of life, economic damage and long-term societal disruption. Due to their sudden occurrence and complex geophysical nature, accurately predicting earthquake-related characteristics remains one of the most challenging problems in earth science. In recent years, the rapid growth of seismic data archives and advances in computational capacity have enables data-drives approaches to complement traditional seismological analyses. In this context, machine learning based methods have gained increasing attention for their ability to extract meaningful patterns from large scale and multidimensional earthquake datasets.

The motivation for applying machine learning techniques to seismic data lies in their capability to model nonlinear relationships and manage heterogeneous variables simultaneously. Earthquake catalogs typically include spatial, temporaly and physical parameters such as geographic coordinates, depth, magnitude, station distributions and measurement uncertainties. Conventional statistical approaches often face limitations when dealing with such complies interactions, whereas machine learning models can learn these relationships directly from data without requiring explicit assumptions. Consequently, even relatively small improvements in classification to seismic risk assessment, early warning strategies and disaster management planning.

Most studies in the existing literature focus on predicting a single earthquake related parameter, such as magnitude, peak ground acceleration or seismic intensity often using region-specific data set or limited temporal scopes. Although these studies offer important insights their applicability is frequently constrained by geographical or data-related limitations. Moreover, magnitude estimation is predominantly treated as a regression problem, while the classification of earthquake magnitude measurement types such as mb, ml, md or mw has received comparatively limited attention. However, magnitude measurement types are critical in seismological interpretation as different scales are designed for specific magnitude ranges, distances , wave characteristics and inconsistencies in these labels may directly affect analytical outcomes.

Another significant challenge in earthquake data analysis is the quality and structure of real-world datasets. Seismic catalogs may contain typographical inconsistencies, non-standardized class labels, missing observations and highly imbalanced class distributions. These issues can negatively influence model performance and reduce the reliability of predictive results if not properly addressed. Therefore, evaluating the effects of data preprocessing strategies, class reorganization and feature selection methods is essential for improving both model accuracy and robustness.

In this study, a comprehensive comparative analysis is conducted using global earthquake data recorded between 2000 and 2023. Two complementary machine learning tasks are addressed: (i) classification of earthquake magnitude

measurement types (Mag Type) and (ii) regression-based prediction of earthquake magnitude (Mag). ANN, RF, SVM and kNN algorithms are employed to assess their performance on a large scale seismic dataset. Unlike many previous studies, this research not only compares widely used machine learning models but also systematically examines the impact of class reorganization based on USGS standarts and feature selection techniques based on Information Gain and Gain Ratio.

The main contribution of this study lies in its integrated evaluation framework, which combines classification and regression analyses on a unified global dataset while explicitly

investigating the role of data preprocessing and feature selection in model performance. By providing a detailed comparative assessment of commonly used machine learning algorithms, this study aims to offer practical insights into their strengths and limitations for seismic data analysis. The findings are expected to support the development of more reliable data driven approaches for earthquake risk analysis and disaster management applications.

II. RELATED WORKS

In this section, relevant studies in the literature related to earthquake prediction and machine learning-based seismic analysis are reviewed.

Banna et al. [1] systematically examined the applications of artificial intelligence-based techniques in earthquake prediction. They evaluated 84 machine learning-based studies in the literature and compared the performance of various methods, from rule-based models to deep learning algorithms. The study aimed to identify the most suitable techniques for earthquake prediction by analysing different datasets and evaluation criteria. As a comprehensive survey study, this work provides a broad comparative overview of existing AI-based approaches; however it does not include an experimental evaluation of models under a unified dataset and consistent experimental framework. Biswas et al. [2] used techniques in their study, such as machine learning and artificial neural networks (ANNs) to estimate seismic hazards. This study, conducted on the February 2023 earthquakes in Turkiye, employed various algorithms, including decision tree regressors, Random Forest, XGBoost, LightGBM, and CatBoost. The predictive success of the developed models for 15 regions was mapped using geographic information systems (GIS), and the ANN models were noted for their high R^2 values. It was stated that the results obtained could be useful in disaster management and infrastructure planning. However, the study is based on a specific seismic event and a regional case study in Turkiye, and its reported performance may therefore require further validation on large scale and globally heterogeneous earthquake datasets. Demirelli et al. [3], in their study, used a dataset combining earthquake catalog data with geological and geodetic data to estimate earthquake magnitude. The dataset was divided into training and testing; models were trained using Random Forest, Extreme Gradient Boosting, decision tree, and k-Nearest Neighbour algorithms. The results indicated that RF and Extreme Gradient Boosting algorithms yielded the most successful results in terms of mean squared error (MSE). The study primarily addresses magnitude estimation through regression models, leaving other catalog based tasks such as magnitude type classification outside its scope. Doğan et al. [4] in their study, used machine learning techniques to predict the focal location and depth of potential earthquakes of magnitude 6 and above that could affect Izmir province. The input data was earthquake catalog data from 1900 to 2023 and the a and b coefficients obtained according to the Gutenberg-Richter law. Predictions made with Random Forest, Decision Tree, LightGBM, CatBoost, and SVM algorithms were evaluated using performance metrics such as RMSE, MAE, and R^2, and the results were visualized on a map. Similar to other region specific studies focusing on recent seismic events, [2] , this work emphasizes the prediction of hypocenter location and depth within a local context, rather than magnitude classification or global scale analyses. Karcı et al. [5] proposed a model to predict the magnitude and timing of future earthquakes using catalog data from earthquakes that occurred in Türkiye. The study used a Long Short-Term Memory (LSTM) model trained on time series data, and the prediction results were compared with multiple linear regression, polynomial regression, decision trees, and Random Forest algorithms. According to the findings, the LSTM model produced the highest accuracy. This study concentrates on time series prediction of earthquake magnitude and occurrence time within a national dataset, which differs from approaches based on global catalogs and multi-task classification and regression frame work. Zhao et al. [6] in their study, created a comprehensive artificial intelligence dataset called "DiTing" based on seismic catalog reports from the China Earthquake Network Center between 2013 and 2020. The dataset includes 2,734,748 three-component waveform traces and various seismic parameters from 787,010 seismic events recorded by broadband and short-period seismometers. This high-quality dataset can form the basis for machine learning models developed for earthquake detection, phase selection, magnitude estimation, and early warning systems. The main contribution of this work lies in dataset construction and benchmarking, while model performance evaluation is left to subsequent machine learning studies. Abdalzaher et al. [7] in their study, developed a machine learning model that aims to rapidly predict earthquake intensity using the first 2 seconds of data after the onset of the P wave. The model, trained on data from the Italian national seismic network, produced successful predictions with a 98.59% accuracy rate using 2-second three-component seismic window data. It has been suggested that this model, called 2S-ML-EIOS, could be used in early warning systems by integrating it with a centralized IoT system. Focusing on rapid intensity estimation within early warning framework, the approach emphasizes real time waveform processing rather than retrospective catalog based modelling. Joshi et al. [8] in their study, developed a cross-region prediction model called SeisEML to estimate the peak ground acceleration (PGA) at a specific point during an earthquake. The model was trained and tested with data from the Kyoshin Network in Japan. Evaluated by the mean absolute error (MAE) and root mean square error (RMSE) values, the model was found to have lower error rates than traditional attenuation relationships. The SeisEML model is notable for its applicability to seismic scenarios in different regions. Jena et al. [9] integrated the SHAP method with artificial neural network (ANN) and random forest (RF) algorithms to predict earthquake probability in the Indian subcontinent. In the study, the explainability of the models was increased using eight different factors, and the most influential variables were ranked. ANN achieved 96% accuracy, and RF achieved 98% accuracy. The researchers stated that explainable artificial intelligence (XAI) methods significantly contribute to model interpretability in earthquake prediction. The emphasis of this work is on model interpretability and probability assessment, which differs from performance oriented comparisons of classification and regression tasks on large scale earthquake catalogs. Bhatia et al. [10] proposed a smart earthquake prediction framework that integrates the Internet of Things (IoT), cloud, and edge computing technologies. Real-time sensor data was collected at the edge layer and classified using a Bayesian belief model, followed by the ANFIS algorithm to estimate the magnitude of earthquakes. The developed model produced highly accurate results with 95.26% reliability and 92.16% stability, and it was observed that edge computing significantly reduced computational latency. Such system oriented frameworks primarily address real time data acquisition and computational efficiency, which differs from catalog driven comparative analyses of classification and regression tasks. Zhang et al. [11] examined the use of artificial neural networks (ANNs) in earthquake prediction, highlighting the "black box" nature of these methods and the weaknesses of their benchmarks. In their study, the LSTM model, developed using human-generated nighttime light data and earthquake energy, demonstrated superior performance compared to the spatially constant Poisson (SUP) model. However, this success was shown to be misleading, with the results significantly changing when a more robust benchmark model, the spatially

varying Poisson (SVP), was used. Zhu et al. [12] focused on analysing gravity changes observed before earthquakes by establishing a mobile gravity monitoring network in mainland China. The research developed methods to assess the usability of gravity data in earthquake prediction, and the resulting data was used in annual earthquake trend forecasts. The study demonstrated how physical measurement data can be used in earthquake prediction, providing a basis for integrating such data with artificial intelligence-based approaches. Jiao et al. [13] presented a comprehensive assessment of the current status of artificial intelligence applications in seismology. The study examined the performance of machine learning and deep learning techniques in seismic data analysis and also discussed the potential future applications of deep learning-based seismological systems integrated with Internet of Things (IoT) platforms. In this context, the development directions of AI-assisted earthquake engineering were presented. Saad et al. [14] developed a real-time earthquake prediction framework for use in seismogenic regions in southwestern China. The model processed large volumes of data obtained from electromagnetic and geo-acoustic sensors using principal component analysis and a random forest algorithm. The system, trained on data from 2016 to 2020, was tested throughout 2021 and achieved approximately 70% accuracy. The results offered the potential to predict earthquake location and magnitude on a weekly basis. Gitis et al. [15] compared various modelling approaches for forecasting strong earthquakes in Japan and California. Based on the minimum alert area method, the study tested five different forecast models: random guessing, spatial density of seismic epicenters, GPS data, seismological data, and a combination of these two. The findings demonstrated that integrating data sources can improve forecast accuracy. The line of research highlights the role of multisource data integration for regional earthquake forecasting, whereas the present study focuses on classification and regression analyses using global earthquake data.

- A review of the studies in the literature reveals the following contributions to this study:
- Predictions were also performed using Python with different machine learning algorithms.
- Both classification and regression models were implemented to demonstrate the applicability of the AI-based approach in earthquake prediction.
- The target variable for classification was "Mag-Type—the measurement type of earthquake magnitude," and for regression, the target variable was "Mag—earthquake magnitude."
- Models were trained using RF, ANN, SVM, and KNN machine learning algorithms; results were compared; and performance analyses were conducted.
- Performance metrics used to evaluate model success were accuracy, MAE, MSE, RMSE, R-squared, confusion matrix, classification performance evolutions, IG, and GR.

The rest of the article is planned as follows: The third section introduces the dataset, analysis method, and methods used in the study. The fourth section presents experimental results obtained through classification and regression applications. The fifth section presents discussion and general conclusions.

III. METHODS

This study aims to estimate earthquake magnitude measurement type and magnitude using artificial intelligence methods using the Worldwide Earthquake Analysis Dataset. To this end, machine learning models such as Random Forest (RF), Artificial Neural Networks (ANN), K-Nearest Neighbour (kNN), and Support Vector Machines (SVM) were applied in classification and regression analyses. The characteristics of the dataset, the preferred machine learning methods, and the metrics used to evaluate model performance are presented in detail in the relevant section. The general flowchart for the data preprocessing, modelling, and evaluation processes used in the study is presented in Fig 1. Data cleaning was performed first, followed by classification and regression analyses. After selecting the most successful model for classification, various scenarios were compared.

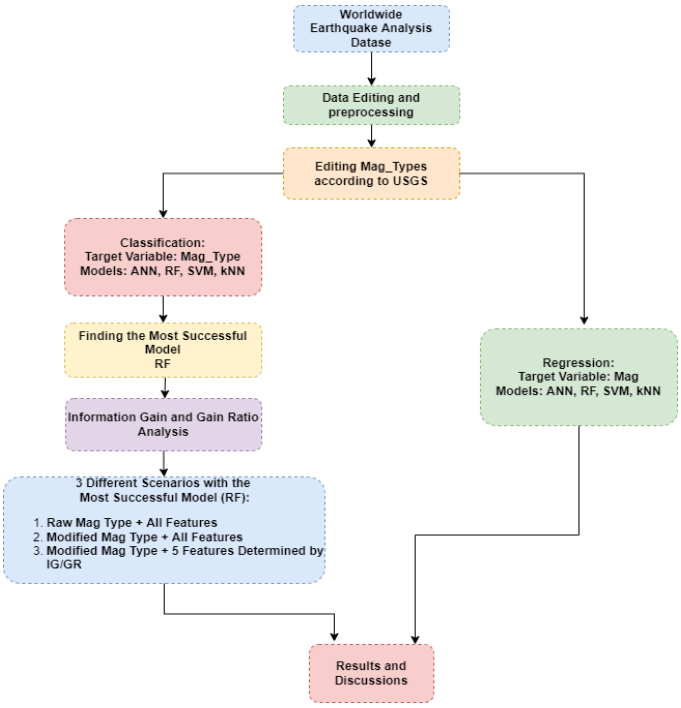

Fig. 1. Flowchart of the study

A. *Worldwide Earthquake Analysis Dataset*

The dataset used in this study is the "Worldwide Earthquake Analysis" dataset, which is openly available on the Kaggle platform at https://www.kaggle.com/datasets/grigol1/earthquakes-2000-2023. The dataset is obtained from the United States Geological Survey (USGS) Earthquake Catalog. The dataset contains details of approximately 0.6 million earthquakes of magnitude 2.5 or greater that occurred worldwide between January 1, 2000, and February 11, 2023. The raw dataset contains a total of 613,787 rows and 23 columns. However, in this study, only features relevant to the analysis were considered, missing or insignificant columns were removed, and only observations with complete data were selected. This study addressed two different machine learning problems using the dataset:

1. Classification: In this study, the variable mag_type (magnitude measurement type) was used as the target variable in the classification models. This variable represents the different scale types used to measure earthquake magnitude (e.g., mb, ms, mw, ml, etc.). Each measurement type corresponds to a different class label in the dataset. However, it was observed that the mag_type values in the dataset occasionally contained typographical errors, or that measurement types that should be included in the same category were expressed in different formats. Therefore, to ensure accurate classification, the mag_type variable was reorganized in accordance with the standards defined by the United States Geological Survey (USGS). Data with very low representation in the dataset or those that were unidentified were marked as "unknown," and data were grouped under the "other" heading during the analysis process. The mag_type classes defined by the USGS are presented in Table 1, and the

reorganization process performed in this study is detailed in Table 2.

TABLE I. USGS MAGNITUDE TYPES

Mag. Type	Mag. Range	Distance Range	Comments
Mww (Generic Notation Mw) (Moment W-phase)	5.0 and larger	1 -90 degrees	It is calculated globally for earthquakes of magnitude 5.0 and above. It provides reliable results up to M4.5 with broadband stations.
Mwc (Centroid)	~5.5 and larger	20 - 180 degrees	Calculated based on surface waves for earthquakes of magnitude 6.0 and above. Generally used if Mww is not available.
Mwb (Body Wave)	~5.5 to ~7.0	30 - 90 degrees	It is based on body waves for earthquakes between magnitudes 5.5 and 7.0. It is activated if Mww and Mwc cannot be calculated. It has limited applicability for earthquakes of magnitudes 7.5 and above.
Mwr (Regional)	~4.0 to ~6.5	0 - 10 degrees	It is calculated using regional data for earthquakes between magnitudes 4.0 and 6.5. In some areas of the United States, it can be applied up to magnitude 3.5.
Ms20 or Ms (20sec surface wave)	~5.0 to ~8.5	20 - 160 degrees	It is used for shallow and large earthquakes between magnitudes 5.0 and 8.5. However, it is generally not preferred when Mw is present.
mb (short-period body wave)	~4.0 to ~6.5	15 - 100 degrees	It is calculated based on the short-period P-wave amplitude for earthquakes between magnitudes 4.0 and 6.5. Saturation occurs after magnitude 6.5.
Mfa(felt-area magnitude)	Any	Any	It is a magnitude estimate for earthquakes that were felt in the past but have no instrumental record. It is based on felt area data.
ML Ml, or ml (local)	~2.0 to ~6.5	0 - 600 km	It is calculated using the classical Richter method for local earthquakes between magnitudes 2.0 and 6.5. It is generally suitable for small and localized events.
mb_Lg, mb_lg, or MLg (short-period surface wave)	~3.5 to ~7.0	150 – 1110 km (10 degrees)	A magnitude calculated based on Lg waves for small earthquakes in the Eastern and Central United States. It is typically used when mb or Mw are unavailable.
Md or md (duration)	~4 or smaller	0 - 400 km	For earthquakes of magnitude 4.0 and below, it is calculated based on signal duration. In the past, it was preferred due to analog recording limitations.
Mi or Mwp (integrated p-wave)	~5.0 to ~8.0	All	It is the magnitude calculated based on the displacement integral of the P wave. It is used with broadband instruments.
Me (energy)	~3.5 and larger	All	The magnitude of an earthquake is based on the seismic energy released by the earthquake. It is estimated from digital data
Mh	Any	Any	It is a non-standard provisional magnitude, used when other methods are inadequate.
Finite Fault Modeling	~7.0 and larger	30 - 90 degrees	It provides slip and moment release modelling for earthquakes of magnitude 7.0 and above. It is used in applications such as ShakeMap and stress analysis.
Mint(intensity magnitude)	Any	Any	It is the magnitude estimated based on perceived intensities, especially before instrumental data.

TABLE II. RAW DATA ADJUSTED ACCORDING TO THE USGS STANDARD

Raw Mag_Type Data in the Data Set	Mag_Type Data Adjusted According to USGS
Mww,mvb,mwc,mwp,mwr	mw
Md, md	md
ml,mlg,mlr,mlv	ml
mb, mb_lg ,mblg	mb
Ms,ms	ms
Ms-20, ms_20	ms_20
m, ma, mc, mh, mi, MI,	Other
Unknown	Other

2. Regression: The Magnitude variable was used as the target variable in the regression model. This feature is a continuous numerical value indicating the earthquake's intensity. Both analyses were conducted independently, examining the effects of different input features on classification accuracy and regression success. Descriptions of all features used in the dataset are presented in Table 3.

TABLE III. FEATURES AND DESCRIPTIONS IN THE WORLDWIDE EARTHQUAKE ANALYSIS DATASET

Features	Descriptions
Time	Earthquake Occurrence Time (UTC)
Latitude	Latitude Information of the Earthquake Center
Longitude	Longitude Information of the Earthquake Center
Depth	The Depth of the Earthquake below the Earth's surface (in kilometers).
Mag	The magnitude of the earthquake (in numerical value).
MagType	The type of quantity measurement used (e.g. mb, ml, mw, md, etc.)
NST	The number of stations used to detect the earthquake.
GAP	The spacing angle of the seismic stations (in degrees).
dMin	Minimum station distance from the earthquake epicenter (in degrees).
RMS	Root mean square error of measurements.

NET	Short name of the network that reports the earthquake.
ID	The unique identification number of the earthquake.
Updated	Date and time the data was last updated.
Place	Description of the location where the earthquake occurred.
Type	The type of event (often referred to as "earthquake").
Horizontal Error	Horizontal margin of error in location information (in kilometers).
Depth Error	Margin of error in depth measurement.
Mag Error	Margin of error in size measurement
MagNST	Number of stations used in the magnitude calculation.
Status	The status of the data (e.g. reviewed, automatic).
Locations	The source network that provides location information.
Mag Source	The source network providing the size information.

B. Random Forest Algorithm (RF)

The algorithm used to make better predictions in samples that result in the formation of multiple decision trees is called the Random Forest Algorithm. The RF (Random Forest) technique was developed and combined by Breinman [16]. The decision trees resulting from each sample set are randomly selected without any specific basis, so it is called "Random Forest" [16]. It is one of the methods widely used in scientific research where a large amount of statistical data is available [17]. The working principle of the RF (Random Forest) system can be briefly described as follows [18]:

1. Sampling: Randomly creates subsets from the training dataset collected for scientific study.
2. Creating a decision tree: A different decision tree is created for each subset. In this stage, randomly selected features are used to determine the best split at each node.
3. Making predictions: Two types of methods are used when making predictions. In the classification method, the predictions for all trees are collected and the result with the most votes is presented, or when the regression method is used, the average prediction is presented as the result.

C. k-Nearest Neighbor (kNN)

The foundations of k-Nearest Neighbor (kNN) were first laid by Cover and Hart in the mid-20th century [19]. The k-Nearest Neighbor method was proposed in the field of pattern recognition and developed as a basic statistical classification method [20]. Today, k-Nearest Neighbors (kNN) continues to be a widely used method in machine learning and statistics [21]. The basic principle of the kNN algorithm is to examine the nearest neighbors of a data point to determine its class or value. k-Nearest Neighbor (kNN) classification is one of the most basic and simple classification methods. When little or no prior knowledge of the dataset is available, kNN is often the first choice. This method emerged as an alternative to the need for discriminant analysis in cases where reliable parametric estimates of probability densities are unknown or difficult to obtain [22]. This method works by measuring the proximity (K) of samples in a dataset to each other. The performance of the algorithm depends on the chosen K value and the distance metrics used (Euclidean, Manhattan, etc.) [23].

D. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification algorithm used in machine learning. Its main goal is to find the most optimal boundary (decision boundary) separating data points belonging to different classes. Support Vector Machine (SVM) is a machine learning method developed by Vapnik based on the fundamentals of statistical learning theory (Vapnik-Chervonenkis (VC) theory) to solve classification, pattern recognition, and regression problems. Support Vector Machine (SVM) algorithm is probably one of the most widely used kernel-based learning algorithms. It provides effective pattern recognition performance by utilizing robust concepts of optimization theory [24].

Support Vector Machine (SVM) is a supervised machine learning method based on statistical learning theory and the principle of structural risk minimization. For binary classification, SVM maximizes the distance between the closest instances of the classes in the training data while generating optimal boundary planes that separate members of the two classes. However, most real-world problems are not linearly separable. To address nonlinearities, kernel functions transform the input data into a high-dimensional feature space where the data is assumed to be linearly separable. The training points closest to the optimal boundary plane are called support vectors. The performance of Support Vector Machine (SVM) depends heavily on the selection of an appropriate kernel and regularization parameters. Linear, polynomial, RBF (Gaussian), and sigmoid are the four Support Vector Machine (SVM) kernel functions frequently used in the literature [25].

E. Artificial Neural Network (ANN)

Throughout history, scientists have been curious about the workings of the human brain and have sought to understand and imitate it. Artificial Neural Networks (ANNs), as the name suggests, are logical software designed to mimic biological—natural—neural networks, simulating the brain's ability to learn, remember, generalize, and generate new knowledge [26]. The concept of Artificial Intelligence was introduced by John McCarthy in 1956 and was based on the possibility that machines could mimic human behaviour and think. It was also previously discussed by Alan Turing, who developed the Turing test to identify the differences between humans and machines. Under the leadership of pioneers such as Marvin Minsky and Frank Rosenblatt, who first described artificial neural networks, many fields of medicine, especially neuroscience, have been deeply studied thanks to this technology. Additionally, due to various reasons, such as the lack of algorithms capable of covering a large portion of real-world problems and the fundamental difficulties of data storage and computation, the need for developing these methods has increased. Over time, fundamental conceptual discoveries such as backpropagation and convolution have shaped the advancement of this field. Current neural network architectures can be categorized into three main categories: feedforward neural networks, feedback neural networks, and self-organizing neural networks. Each category is based on a different philosophy and follows different principles. Generally speaking, describing a system as a "neural network" generally implies that it is capable of learning. Learning is the process by which neural systems acquire the ability to perform specific tasks. During this process, the system adjusts its internal parameters according to a specific learning pattern. Depending on the specific neural network architecture considered, learning can occur in a supervised or unsupervised manner [27].

F. Mean Squared Error (MSE)

Mean Squared Error (MSE) is a frequently used error metric in statistics and machine learning. MSE is calculated by

averaging the squared differences between the true values and the predicted values. This metric is used to measure how much the model's predictions deviate from the true values. The Mean Squared Error (MSE) equation is shown below as (1), where "n" represents the number of points and "e" represents the error [28].

$$MSE = \frac{1}{n}\sum_{j=1}^{n} e_j^2 \quad (1)$$

G. Root Mean Squared Error (RMSE)

The second parameter used is the Root Mean Squared Error (RMSE). This metric is used to measure the magnitude of the difference between the predicted values and the actual values. Root Mean Squared Error (RMSE) is defined as the standard deviation of the errors and its value varies between 0 and infinity. The closer the value is to 0, the better the estimator performs. RMSE is calculated by taking the square root of the MSE, and its equation is given by (2) below [28].

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} e_j^2} \quad (2)$$

H. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is the most frequently used basic error metric in the literature, which measures how well a forecasting model performs. MAE is the average of the absolute values of the differences between the actual and predicted values. In other words, it measures how much each forecast deviates, but it does not take into account the direction of the deviation (positive or negative). The formula for calculating MAE is given by (3), where "p" represents the actual value, "r" represents the predicted value, and "N" represents the number of observations [29].

$$MAE = \frac{\sum_{(u,i)} |p_{u,i} - r_{u,i}|}{N} \quad (3)$$

The closer the result is to zero, the better the model is considered [30].

I. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) is one of the most common metrics used to measure the accuracy of forecasting models. MAPE is obtained by averaging the Absolute Percentage Errors (APE) calculated for each observation. Here, for each data point, y_t is the true value and $\hat{y}_t$ is the predicted value, and MAPE is defined by (4):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left|\frac{y_t - \ddot{y}_t}{y_t}\right| \quad (4)$$

Coefficient of Determination / R- Squared (R2)

One of the most commonly used metrics for evaluating a model's success in regression analysis is R-squared (R^2). R^2 indicates how much of the total variance in the dependent variable is explained by the independent variable(s). Mathematically, it expresses the model's explanatory power with a value between 0 and 1:

$$R2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (5)$$

Here, SS_{res} represents the sum of squares of the model's prediction errors (Residual Sum of Squares), and SS_{res} represents the sum of squares of the deviations of the observed values from the mean (Total Sum of Squares).

The resulting R^2 value:

• As it approaches 1, it indicates that the model better explains the data.

• As it approaches 0, it indicates that the model's explanatory power is weaker.

One of the most important advantages of R^2 is that it is scale-independent, making it easier to intuitively interpret the model's performance. For this reason, it is widely preferred in applied sciences and especially in industrial projects. However, many studies emphasize that error metrics such as MSE, RMSE, MAE, and MAPE have limited interpretability, whereas R^2 provides more informative and meaningful results [30].

In particular, while error metrics tend to deviate when true values are close to zero, R^2 is less affected by such limitations because it only calculates the overall variance. Therefore, the use of R^2 as a standard metric for evaluating regression models is increasingly considered appropriate.

J. Confusion Matrix

The Confusion Matrix is a fundamental tool used to evaluate the performance of classification models. This table, which details the model's correct and incorrect classifications, forms the basis for calculating classification performance metrics. Common performance metrics such as accuracy, precision, recall, and F1-score are obtained directly from the confusion matrix [32].

This matrix consists of four basic values:

- True Positive (TP): Data belonging to the positive class are correctly classified as positive.
- True Negative (TN): Data belonging to the negative class are correctly classified as negative.
- False Positive (FP): Data belonging to the negative class are incorrectly classified as positive.
- False Negative (FN): Data belonging to the positive class are incorrectly classified as negative.

The metrics obtained from these values allow us to evaluate the classification performance of the model from different perspectives.

K. Classification Performance Evolutions

Various metrics are used to evaluate the performance of classification algorithms from different perspectives. In this study, common metrics such as accuracy, precision, recall, and F1-score were chosen. These metrics are calculated based on the TP, TN, FP, and FN values in the confusion matrix [33]. Because evaluating using a single metric can be misleading, especially in imbalanced datasets, multiple performance metrics were considered in this study. Accuracy; this represents the overall accuracy of the model. It measures the total number of correct predictions in both positive and negative classes, divided by the total number of predictions. It provides a reliable assessment in balanced datasets. Accuracy formula is given by (6).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision; measures the accuracy of positive predictions. It indicates the fraction of instances predicted as positive by the model that are actually positive. It is a priority in applications where false positive predictions are costly. Precision formula is given by (7).

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall (Sensitivity); shows how much of the data belonging to the true positive class was correctly predicted. This is important in situations where missing a positive class is critical (e.g., disease detection). Recall formula is given by (8).

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

F1-Score; the harmonic average of the precision and recall values. This metric provides a more realistic measure of success by balancing these two metrics, especially in datasets with class imbalances. F1-Score equation is given by (9).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

L. Information Gain

Information Gain is a fundamental metric used specifically in the construction of decision trees and is used to determine how much a feature reduces uncertainty (entropy) in a dataset. It measures the ability of a given feature to divide the dataset into more homogeneous subgroups, revealing its informativeness for classification. This metric is used to determine which feature will be included in the top node of a decision tree, and the feature with the highest information gain is selected first [34]. Information gain is calculated by calculating the difference between the average entropy before applying a feature and the average entropy after it is applied. A decrease in entropy indicates that the dataset is more organized and the model can make more effective classifications.

M. Gain Ratio

Gain Ratio supports the selection of features that divide the dataset into more homogeneous subgroups while also providing a balanced separation. This contributes to the creation of a more generalizable and meaningful decision tree [34].

IV. PEXPERIMENTAL RESULTS

This section presents the results of the classification and regression analyses. First, the classification operations for the Mag_Type variable were evaluated separately using the (ANN), (RF), (SVM), and (KNN) models. Based on the results, the most successful model was rerun using both IG and GR, and the results were compared for all three scenarios. Then, regression analysis for the Mag variable was included, and model performance was compared.

A. Mag_Type Classification Results

This section presents the findings of the classification analyses for the Mag_Type variable. The results for the ANN, RF, SVM, and KNN models are evaluated, respectively.

1) Classification Results According to The ANN Model

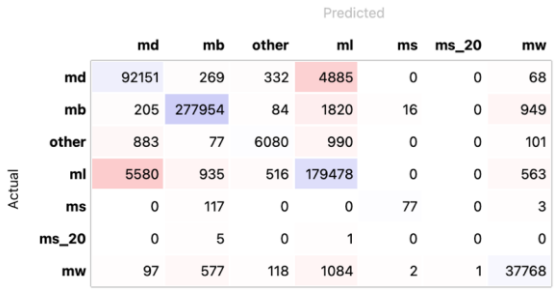

Fig. 2. Confusion matrix generated by the ANN model for Mag_Type classification

Above, based on the confusion matrix generated by the ANN model from a total of 613,786 observations (see Fig.2), interpretations based on TP, TN, FP, and FN values for each Mag_Type class are presented separately, as well as analyzed using general evaluation metrics. Significant differences are observed in the model's classification performance for different Maga_Type classes.

The model's predictive performance on the "mb" class is observed to be quite high. The vast majority of instances belonging to this class were correctly classified (True Positive -TP), while the number of incorrectly classified instances (False Negative -FN) was quite low. Furthermore, only a limited number of observations from instances not belonging to this class were incorrectly labelled as "mb" (False Positive - FP). The remaining numerous instances were correctly identified as not belonging to the "mb" class (True Negative - TN). These findings demonstrate that the model can distinguish the "mb" class with high accuracy.

Similarly, the model demonstrates high performance in the evaluations for the "md," "ml," and "mw" classes. For these classes, TP values are quite high, while FN and FP values are relatively low. This demonstrates that the model can successfully learn the patterns of these classes and effectively distinguish between them.

On the other hand, the model's performance is relatively lower for the "other" class. Although a certain number of correct predictions (TP) were made, a large number of instances belonging to this class were misclassified (FN), and some observations belonging to different classes were incorrectly assigned to this class (FP). This may be due to the inherent uncertainty in the "other" class or its heterogeneous sample size.

In the "ms" and "ms_20" classes, the model's discrimination success is quite low. In the "ms" class, in particular, the number of correctly classified examples is very low, and many observations are mixed with other classes. Although the FP value is limited, it appears that the model generally fails to correctly label examples belonging to this class. For the "ms_20" class, the model failed to produce any accurate predictions (TP=0). Not only were all observations belonging to this class assigned to different classes (FN>0), but also some examples not belonging to this class were incorrectly included (FP>0). These results indicate that the model failed to adequately learn the patterns associated with this class.

The model performs with high accuracy in classes with a large sample size and distinct attributes (e.g., "mb," "md," "ml," and "mw"). However, it makes significant classification errors in classes with a limited sample size or weak discrimination power (especially "ms" and "ms_20"). This may be due to an imbalance in the class distribution in the dataset or insufficient distinguishing features for some classes.

In the evaluations made on the confusion matrix, it was observed that the model exhibited a strong classification performance in classes with high sample numbers such as "mb", "md", "ml" and "mw", whereas the performance remained relatively low in classes such as "ms", "ms_20" and "other".

In addition to these class-based evaluations, the overall success of the model was also supported by basic classification metrics, and these metrics are given in Table 4.

TABLE IV. MAG_TYPE ANN MODEL CLASS-BASED EVALUATION METRICS

Model	Ca	F1	Precision	Recall
ANN	0.967	0.967	0.967	0.967

Accordingly, the overall accuracy rate (Classification Accuracy – CA) of the ANN model was calculated as 96.7%. Precision, Recall, and F1-score values, which indicate the extent to which the model achieved inter-class balance, were equally high at 0.967. These rates demonstrate that the model offers strong generalization capabilities along with high accuracy.

The classification process performed with the ANN model demonstrates successful performance in terms of both detailed class-based and general metrics and is particularly effective for accurately classifying commonly used Mag_Type types. For minority classes, further performance improvement is possible with a more balanced data structure or class-weighted learning strategies.

2) Classification Results According to The RF Model

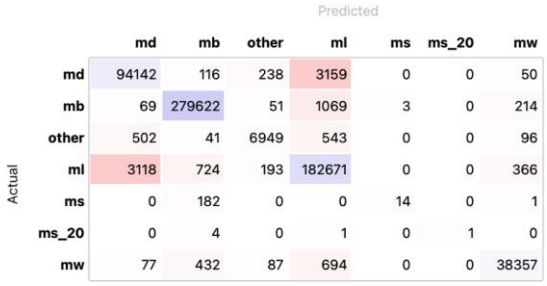

Fig. 3. Confusion matrix generated by the RF model for Mag_Type classification

The confusion matrix generated using the RF model, based on 613,786 observations (see Fig.3), provides separate interpretations based on TP, TN, FP, and FN values for each mag type class. A comprehensive analysis was also conducted based on overall performance metrics.

The model's classification performance in the "md" class is quite successful. A total of 94,142 samples were correctly identified as "md" (TP), while 3,563 were incorrectly assigned to other classes (FN). Furthermore, 3,766 samples unrelated to the "md" class were incorrectly assigned to this class (FP). The remaining 512,315 samples were correctly predicted for classes other than "md" (TN). These results demonstrate that the model can identify the "md" category with high accuracy.

The model's classification accuracy in the "mb" class is also quite high. A total of 279,622 samples were correctly identified as "mb" (TP), while 1,406 were incorrectly assigned to other classes (FN). Furthermore, 1,499 samples unrelated to "mb" were incorrectly assigned to this class (FP). The remaining 331,259 samples were correctly identified as classes other than "mb" (TN). These results demonstrate that the model is highly effective in distinguishing the "mb" class.

For the "other" class, the model's performance was moderate. 6,949 instances were correctly identified (TP), but 1,182 were incorrectly classified (FN). Conversely, 569 instances were incorrectly assigned to this class (FP). 605,086 instances were correctly identified as non-"other" (TN). These results indicate that the model was partially successful in identifying the boundaries of this class, but it encountered some difficulties.

The model performed quite well in the "ml" class. There were 182,671 correct predictions (TP) and 4,401 incorrect classifications (FN). 5,466 examples were incorrectly assigned to this class (FP). 421,248 examples were correctly classified as non-"ml" (TN). This data suggests that the model generally correctly identified this class, but confusion occurred in some cases.

For the "ms" class, the model's performance was quite limited. Only 14 correct predictions (TP) were made, and 183 examples were misclassified (FN). Three examples were incorrectly assigned to this class (FP), while 613,586 examples were correctly identified as non-"ms" (TN). This suggests that the model is inadequate at distinguishing this class.

The model's performance is quite poor on the "ms_20" class. Only one correct prediction (TP) was made, and five instances were misclassified (FN). Despite no false inclusions (FP), 613,780 instances were correctly identified as non-"ms_20" (TN). These results clearly demonstrate that the model barely recognizes this class.

For the "mw" class, the model's performance was generally positive. There were 38,357 correct predictions (TP) and 1,290 misclassifications (FN). 727 examples were incorrectly assigned to this class (FP), while 573,412 were correctly assigned to non-"mw" (TN). This data suggests that the model largely correctly identified this class, but there were some minor confusions.

In addition to these class-based evaluations, the overall success of the model was supported by basic classification metrics, and these metrics are given in Table 5.

TABLE V. MAG_TYPE RF MODEL CLASS-BASED EVALUATION METRICS

Model	Ca	F1	Prec.	Recall
RF	0.980	0.980	0.980	0.971

Following the classification process performed with the Random Forest (RF) model, the overall performance of the model was evaluated, and the following basic classification metrics were calculated. According to the results, the model's overall accuracy rate (Accuracy) was determined to be 98.0%. This rate demonstrates the extent to which the model made accurate predictions across all observations, indicating a very high level of success. The Precision, Recall, and F1-score values were all 0.980, indicating that the model performed both accurate and sensitive classification. This demonstrates that the model not only selected the correct classes but also successfully identified the vast majority of examples belonging to the relevant classes. In conclusion, these overall metrics obtained with the RF model confirm that the model achieved a highly successful classification in terms of both accuracy and balance, and this success is clearly visible on a class-by-class basis in the confusion matrix.

3) Classification Results According to The SVM Model

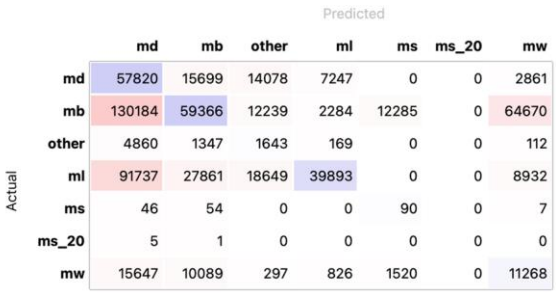

Fig. 4. Confusion Matrix Generated By The SVM Model For Mag_Type Classification.

Above, based on the confusion matrix created from 613,786 observations (see Fig.4) using the SVM model, separate comments based on TP, TN, FP and FN values for each Mag_Type class were included; in addition, a comprehensive analysis was performed on general performance metrics.

Based on the classification results performed with the SVM model, the model's performance in distinguishing various Mag_Type classes was evaluated using TP, FN, FP, and TN values.

For the "md" class, the model correctly classified 57,820 samples (TP), but incorrectly assigned 39,885 samples to other classes (FN). Furthermore, the model incorrectly assigned 242,479 samples that did not actually belong to the "md" class (FP). The remaining 273,602 samples were correctly classified as non-"md" (TN). This suggests that the model can recognize

the "md" class with limited accuracy, but also tends to produce a high number of false positives for this class.

A similar pattern emerges for the "mb" class. The model correctly classified 59,366 instances (TP), but assigned 221,662 instances to different classes (FN). Furthermore, the model predicted 55,051 instances as "mb" even though they did not belong to this class (FP). 277,707 instances were correctly identified as non-"mb" (TN). These results indicate that the model also experienced significant learning difficulties in the "mb" class.

The model's performance for the "other" class was quite low. Only 1,643 samples were correctly classified (TP), while 6,488 were incorrectly predicted (FN). Furthermore, 45,263 samples were assigned to the "other" class by the model even though they did not actually belong to the class (FP). 560,392 samples were correctly predicted as non-"other" (TN). It appears that the model experienced significant difficulty in predicting this class, producing high errors due to unclear class boundaries or unbalanced data distribution.

For the "ml" class, the model correctly predicted 39,893 instances (TP), while 147,179 instances were incorrectly assigned to classes (FN). Furthermore, 10,526 instances were incorrectly classified as "ml" (FP) even though they did not actually belong to that class. 416,188 instances were correctly identified as non-"ml" (TN). This table demonstrates that the model also failed to discriminate in the "ml" class.

The model's classification performance for the "ms" class was quite low. Only 90 samples were correctly classified, while 107 were assigned to different classes (FN). Furthermore, 13,805 samples were incorrectly predicted as "ms" even though they did not belong to this class (FP). 599,784 samples were correctly excluded from this class (TN). This suggests a serious learning and discrimination problem for the "ms" class.

For the "ms_20" class, the model made no correct classifications (TP = 0). All samples belonging to this class were assigned to different classes, and no incorrect predictions were made for this class from other classes (FP = 0). This indicates that the model was unable to learn or distinguish this class at all.

For the "mw" class, the model correctly predicted 11,268 instances (TP). However, 28,379 instances were assigned to different classes (FN), and 76,582 instances were labelled as such despite not belonging to that class (FP). This table demonstrates that the model has a significant overprediction bias in the "mw" class. The model's key metrics are presented in Table 6, and an overall performance evaluation is based on these data.

TABLE VI. MAG_TYPE SVM MODEL CLASS-BASED EVALUATION METRICS

Model	Ca	F1	Prec.	Recall
SVM	0.277	0.298	0.518	0.277

An examination of the SVM model's overall classification metrics reveals that its classification performance is quite low. The model's accuracy rate was 27.7%, indicating that only one in four samples was correctly classified. The F1-score value was 0.298, indicating a poor balance between precision and recall. A particularly low recall rate of 0.277 suggests that the model failed to recognize many correct classes. However, the precision value was relatively high at 0.518, indicating that the model selectively predicted the classes it correctly predicted, but this success was not reflected overall. These data indicate that the SVM model is inadequate in distinguishing between classes in the current dataset and requires further development.

4) Mag_Type Classification Results According to kNN Model

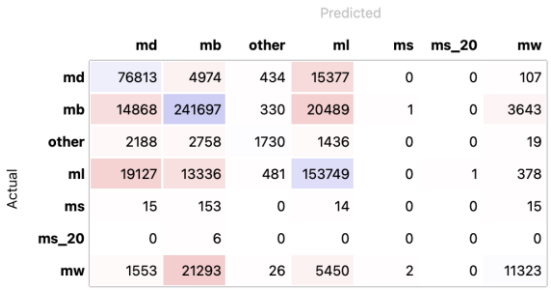

Fig. 5. Confusion matrix generated by the kNN model for Mag_Type classification

Above, detailed class-wise analysis is presented based on True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values for each Mag_Type in the light of confusion matrix on 613,786 samples with KNN algorithm (see Fig.5); in addition, the overall success of the model is evaluated comprehensively using various performance metrics.

When the classification results made with the KNN model were examined, the success of the model was evaluated based on the TP, FN, FP and TN values of each Mag_Type class.

For the "md" class, the model correctly classified 76,813 instances (TP), but confused 20,892 instances with other classes (FN). Furthermore, the model incorrectly assigned 37,751 instances that did not belong to the "md" class (FP). The remaining 478,330 instances were correctly classified as non-"md" (TN). This table shows that the model has a moderate ability to recognize the "md" class and has a significant tendency to produce false positives.

The model's performance is relatively high for the "mb" class. 241,697 samples were correctly classified (TP), but 39,331 were mislabelled to other classes (FN). A further 42,520 samples were incorrectly predicted as "mb" (FP). The remaining 290,238 samples were correctly classified as non-"mb" (TN). These results demonstrate that the model generally performs strongly in the "mb" class.

For the "other" class, the model made 1,730 correct predictions (TP), but 6,401 instances were confused with other classes (FN). Additionally, 1,271 instances were incorrectly labelled "other" (FP). 604,384 instances were correctly excluded from this class (TN). This table demonstrates that the model's ability to distinguish this class is limited.

In the "ml" class, the model correctly predicted 153,749 instances (TP), but assigned 33,323 instances to other classes (FN). At the same time, 42,766 instances were incorrectly classified as "ml" (FP), and 383,948 instances were correctly excluded from this class (TN). This table indicates that the model performed a generally successful classification for the "ml" class, but with a risk of confusion.

For the "ms" class, the model correctly predicted 14 instances (TP) and assigned 183 instances to different classes (FN). FP was negative (-11), likely due to a very small rounding or counting error in the data; functionally, FP should be considered ≈ 0. TN was calculated as 613,600. These data indicate that the model struggled to correctly recognize this class and was unable to distinguish it due to the small number of instances in the class.

For the class "ms_20", the model could not make any correct classification (TP = 0), the FN value is 6, and the FP value can be assumed as 1 – 0 = 1. Completely ignoring this class reveals that the model could not learn any examples belonging to this class.

For the "mw" class, the model correctly predicted 11,323 instances (TP) but incorrectly classified 28,324 instances (FN). Furthermore, 4,162 instances were incorrectly assigned to this class despite not belonging to it (FP). The remaining 570,977 instances were correctly excluded from this class (TN). These results indicate that the model achieved moderate success in the "mw" class but missed many instances, particularly due to the high FN value. The overall performance of the kNN model according to the basic metrics given in Table 7 is as follows.

TABLE VII. MAG_TYPE KNN MODEL CLASS-BASED EVALUATION METRICS

Model	Ca	F1	Prec.	Recall
kNN	0.791	0.782	0.789	0.791

An evaluation of the kNN model's overall classification metrics reveals that the model performs quite satisfactorily on the dataset. The model's accuracy rate is 79.1%, indicating that the vast majority of observations were correctly classified. The F1-score of 0.782, Precision of 0.789, and Recall of 0.791 demonstrate that the model not only accurately predicts the correct classes but also successfully captures examples belonging to these classes to a large extent. These balanced results indicate that the model implements both a selective and inclusive classification strategy.

Overall, the kNN model showed consistent performance on both the prevalent and the minority classes, offering a distinct advantage over weaker models such as SVM.

In this study, four different machine learning models (ANN, RF, SVM, and kNN) were used to classify Mag_Type classes, and the models were evaluated using both class-based confusion matrices and basic performance metrics (Accuracy, Precision, Recall, and F1-score). The Artificial Neural Network (ANN) model demonstrated very high performance, reaching 96.7% in overall accuracy (Accuracy), F1-score, Precision, and Recall, demonstrating the model's robust and balanced discrimination between classes. Very high correct classification rates were achieved, particularly for common classes.

The Random Forest (RF) model also delivered similarly strong performance. Accuracy was 98%, and F1-score, precision, and recall were 0.980. The RF model produced highly successful results in classes with high sample balance and acceptable results in minority classes.

In contrast, the Support Vector Machine (SVM) model exhibited very poor classification performance. Low levels of key metrics such as Accuracy (27.7%) and Recall (0.277) indicate that the model cannot adequately distinguish between classes. The inter-class confusion rate is quite high, and poor results are observed, especially for minority and imbalanced classes. This suggests that SVM is not an appropriate classifier for this dataset.

The k-Nearest Neighbors (kNN) model, on the other hand, demonstrated a balanced performance profile. Accuracy was measured as 79.1%; F1-score as 0.782; Precision as 0.789; and Recall as 0.791. While the kNN model did not achieve as high a performance as ANN and RF, it was able to make a balanced distinction between classes, particularly sensitive to the structure of the dataset, and therefore stood out as a moderately to highly reliable classifier.

5) Effect of Feature Selection and Class Organization on Performance

This section examines how model performance for Mag_Type classification changes with data preprocessing and feature selection. Three different scenarios were created within the analysis, using the Random Forest model, which yielded the most successful classification results, as a reference. Each scenario differs depending on the level of processing applied to the dataset.

In the first scenario, the Random Forest model was applied directly to the dataset without any manipulation. This baseline scenario serves as a reference to see how the model performs with the raw data.

In the second scenario, the classes in the Mag_Type variable were reorganized according to USGS standards. This reorganization corrected typographical errors, combined classes with the same meaning but written differently, and grouped undefined classes under the "other" heading. This step aimed to increase the model's discriminatory power by reducing class imbalances.

In the third and final scenario, both the Mag_Type configuration and Information Gain and Gain Ratio analyses were performed on the features in the dataset, and the results are presented in Table 8. Based on these analyses, the features with the highest information value were identified (e.g., mag, net, dmin, horizontalError, longitude), and the Random Forest model was retrained using only these features. This test tested the model's success with only the most informative variables.

TABLE VIII. FEATURE SELECTION RESULTS (INFORMATION GAIN & GAIN RATIO)

Features	Information Gain	Gain Ratio
mag	0.731	0.366
net	0.475	0.319
dmin	0.460	0.230
HorizontalError	0.453	0.226
Longitude	0.372	0.186

The basic classification metrics obtained from the scenarios are presented in Table 9.

TABLE IX. COMPARISON OF PERFORMANCE METRICS CALCULATED FOR ALL SCENARIOS

Scenario	Explanation	Ca	F1-score	Precision	Recall
1	Mag_Type classes are not adjusted, all features are used.	0.877	0.873	0.870	0.877
2	Mag_Type classes are adjeusted, all features are used.	0.980	0.980	0.980	0.980
3	Mag_Type was adjusted, the 5 most informative features were selected with IG/GR and the others were ignored	0.889	0.887	0.885	0.889

The results show that rearranging the Mag_Type labels significantly improved model performance. Furthermore, selecting and incorporating the most informative features into the model further improved classification performance. Although the gains in the third scenario were limited, the model performed more consistently in terms of both accuracy and overall balance (precision-recall).

These findings reveal that both the integration of class labels and feature selection play an important role in improving

classification accuracy, especially in multi-class and unbalanced datasets.

B. Mag Regression Results

In this study, regression analysis was performed to estimate the magnitude of the earthquake (Mag) and four different machine learning models (ANN, RF, SVM, and KNN) were compared. The performance of the models, along with training and test error values and standard regression evaluation metrics such as MSE, RMSE, MAE, and R^2, are presented in Table 10 and are evaluated in detail below.

TABLE X. REGRESSION PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS

MODEL	MSE	RMSE	MAE	R2
ANN	0.092	0.303	0.217	0.878
RANDOM FOREST	0.059	0.242	0.173	0.922
SVM	1.245	1.116	0.875	- 0.648
KNN	0.257	0.507	0.362	0.660

Although the Artificial Neural Network model produced high error rates on the training dataset, it performed satisfactorily on the test dataset. The model's MSE was calculated as 0.092, RMSE as 0.303, and MAE as 0.217. Furthermore, the R^2 value was 0.878, indicating that the model explained 87.8% of the variance in the dependent variable. While the results demonstrate acceptable generalization, the significant difference between training and test errors suggests that the model may be prone to overfitting.

The RF algorithm demonstrated statistically significant superiority across all performance metrics compared to other models used in regression analysis. The 0.059 MSE and 0.922 R^2 values obtained on the test dataset demonstrate that the model explains 92.2% of the total variance in the dependent variable and achieves a very high level of accuracy in its predictions. Furthermore, the very low error metrics of 0.242 RMSE and 0.173 MAE demonstrate minimal deviation in both absolute magnitude and variance in its predictions and demonstrate consistent performance. These results indicate that the RF model exhibits strong generalization ability not only to the training dataset but also to unseen data. These characteristics of the model can be explained by its ability to effectively learn complex data structures and minimize the risk of overfitting. Therefore, the RF algorithm stands out as the most suitable model for this analysis.

The SVM model had the lowest performance among the models examined. The negative R^2 value (-0.648) indicates that the model failed to explain the variance in the dependent variable and performed worse than the fixed-mean estimation. Furthermore, the MSE value was 1.245, the RMSE value was 1.116, and the MAE value was 0.875. These high MSE, RMSE, and MAE values indicate that the model's prediction errors were large. This suggests that the SVM model was either inappropriate for the current dataset or the model settings were not appropriate for the data.

While the model's R^2 value of 0.660 indicates that it can explain approximately 66% of the total variance in the dependent variable, this seemingly reasonable value can be misleading in the context of test performance. Specifically, the RMSE of 0.507 and the MAE of 0.382 suggest that the model produces errors in its predictions that are beyond acceptable limits.

CONCLUSIONS

In this study, a detailed comparative evaluation of machine learning – based classification and regression models was conducted using global earthquake data obtained from the USGS covering the period between 2000 and 2023. Approximately 614,000 earthquake magnitude measurement types (Mag_Type) and regression based prediction of earthquake magnitude (Mag).

For the Mag_Type classification, RF and ANN models achieved the highest performance among the evaluated algorithms. RF produced the highest overall accuracy, reaching approx. 98%, while ANN achieved a similarly high accuracy level. Precision, recall and F1-Score values for these models were consistently high across major magnitude classes, indicating balanced classification performance. In contrast, SVM and kNN models yielded comparatively for less frequent magnitude types.

For the magnitude regression, RF again demonstrated superior performance. The RF regression model achieved the lowest error values with MAE and RMSE values significantly lower than those of ANN, SVM and kNN models and attained the highest explanatory power with a coefficient of determination $R^2 \approx 0.92$. ANN regression models showed competitive but slightly lower performance while kNN and SVM models were more sensitive to feature distribution and data heterogeneity resulting in higher prediction errors.

The impact of data preprocessing and feature selection was also systematically evaluated. Feature selection on IG and GR methods led to noticeable improvements in both classification and regression results. In particular, reorganising Mag_Type labels in accordance with USGS standarts reduced class confusion and increased classification accuracy and F1-score values across all models. Experimental analyses conducted under different scenarios confirmed that appropriate feature reduction and data organisation play a critical role in enhancing model stability and predictive accuracy.

Overall the results indicate that ensemble based nonlinear machine learning models, especially RF and ANN are highly effective for earthquake prediction tasks when applied to large scale and heterogenous seismic datasets. Although no novel machine learning algorithm was proposed, the study provides a comprehensive quantitative comparison of widely used models and demonstrates hoe preprocessing strategies and feature selection significantly influence predictive performance.

The finding of this study contribute to the understanding of how machine learning models can be effectively applied to earthquake magnitude classification and estimation tasks using global data. Future studies may extend this framework by incorporating waveform-based features, real-time sensors data or hybrid modelling approaches to further improve earthquake prediction performance.

References

[1] Md. H. A. Banna *et al.*, "Application of Artificial Intelligence in Predicting Earthquakes: State-of-the-Art and Future Challenges," *IEEE Access*, vol. 8, pp. 192880–192923, 2020, doi: 10.1109/ACCESS.2020.3029859.

[2] S. Biswas, D. KUMAR, and U. K. Bera, "Prediction of earthquake magnitude and seismic vulnerability mapping using artificial intelligence techniques: a case study of Turkey," 2023, Accessed: Mar. 24, 2025. [Online]. Available: https://www.researchsquare.com/article/rs-2863887/latest

[3] E. Demirelli, H. İ. Solak, and İ. Tiryakioglu, "Makine öğrenmesi algoritmaları ile deprem katalogları kullanılarak deprem tahmini," *Gümüşhane Üniversitesi Fen Bilim. Derg.*, vol. 13, no. 4, pp. 979–989, 2023.

[4] A. Doğan, "İzmir'i etkileyebilecek büyük depremlerin makine öğrenimi yöntemleriyle tahmin edilmesi," *Yerbilimleri*, vol. 45, no. 2, pp. 93–106, 2024.

[5] M. KARCI and İ. ŞAHİN, "Derin Öğrenme Yöntemleri Kullanılarak Deprem Tahmini Gerçekleştirilmesi", Accessed: Mar. 23, 2025. [Online]. Available: https://www.researchgate.net/profile/Ismail-Sahin-7/publication/362044265_Derin_Ogrenme_Yontemleri_Kullanilarak_Deprem_Tahmini_Gerceklestirilmesi/links/62d2dd2266bd1654d66a0919/D

erin-Oegrenme-Yoentemleri-Kullanilarak-Deprem-Tahmini-Gerceklestirilmesi.pdf

[6] M. Zhao, Z. Xiao, S. Chen, and L. Fang, "DiTing: A large-scale Chinese seismic benchmark dataset for artificial intelligence in seismology," *Earthq. Sci.*, vol. 36, no. 2, pp. 84–94, 2023.

[7] M. S. Abdalzaher, M. S. Soliman, and S. M. El-Hady, "Seismic intensity estimation for earthquake early warning using optimized machine learning model," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–11, 2023.

[8] A. Joshi, B. Raman, C. K. Mohan, and L. R. Cenkeramaddi, "Application of a new machine learning model to improve earthquake ground motion predictions," *Nat. Hazards*, vol. 120, no. 1, pp. 729–753, Jan. 2024, doi: 10.1007/s11069-023-06230-4.

[9] R. Jena, B. Pradhan, S. Gite, A. Alamri, and H.-J. Park, "A new method to promptly evaluate spatial earthquake probability mapping using an explainable artificial intelligence (XAI) model," *Gondwana Res.*, vol. 123, pp. 54–67, 2023.

[10] M. Bhatia, T. A. Ahanger, and A. Manocha, "Artificial intelligence based real-time earthquake prediction," *Eng. Appl. Artif. Intell.*, vol. 120, p. 105856, 2023.

[11] Y. Zhang, C. Zhan, Q. Huang, and D. Sornette, "Seismically Informed Reference Models Enhance AI-Based Earthquake Prediction Systems," *J. Geophys. Res. Solid Earth*, vol. 129, no. 3, p. e2023JB028037, Mar. 2024, doi: 10.1029/2023JB028037.

[12] Y. ZHU, F. LIU, G. ZHANG, Y. ZHAO, and S. WEI, "Mobile gravity monitoring and earthquake prediction in China," *Geomat. Inf. Sci. Wuhan Univ.*, vol. 47, no. 6, pp. 820–829, 2022.

[13] P. Jiao and A. H. Alavi, "Artificial intelligence in seismology: advent, performance and future trends," *Geosci. Front.*, vol. 11, no. 3, pp. 739–744, 2020.

[14] O. M. Saad *et al.*, "Earthquake forecasting using big data and artificial intelligence: A 30-week real-time case study in China," *Bull. Seismol. Soc. Am.*, vol. 113, no. 6, pp. 2461–2478, 2023.

[15] V. Gitis, A. Derendyaev, and K. Petrov, "Analyzing the performance of GPS data for earthquake prediction," *Remote Sens.*, vol. 13, no. 9, p. 1842, 2021.

[16] "randomforest2001.pdf." Accessed: Mar. 24, 2025. [Online]. Available: https://www.stat.berkeley.edu/~breiman/randomforest2001.pd

[17] H. Zhang and M. Wang, "Search for the smallest random forest," *Stat. Interface*, vol. 2, no. 3, p. 381, Jan. 2009.

[18] Y. Liu, "Random forest algorithm in big data environment," *Comput. Model. New Technol.*, vol. 18, no. 12A, pp. 147–151, 2014.

[19] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[20] T. M. Cover, "LEARNING IN PATTERN RECOGNITION," in *Methodologies of Pattern Recognition*, S. Watanabe, Ed., Academic Press, 1969, pp. 111–132. doi: 10.1016/B978-1-4832-3093-1.50012-2.

[21] A. Feyzioglu and Y. S. Taspinar, "Beef Quality Classification with Reduced E-Nose Data Features According to Beef Cut Types," *Sensors*, vol. 23, no. 4, p. 2222, Jan. 2023, doi: 10.3390/s23042222.

[22] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[23] K. Chomboon, P. Chujai, P. Teerarassamee, K. Kerdprasop, and N. Kerdprasop, "An empirical study of distance metrics for k-nearest neighbor algorithm," in *Proceedings of the 3rd international conference on industrial application engineering*, 2015, p. 4. Accessed: Mar. 25, 2025. [Online]. Available: https://www.academia.edu/download/109567471/22363962afda432435cdbb478857547a17b5.pdf

[24] A. Feyzioğlu and Y. S. Taspınar, "MALICIOUS UAVS CLASSIFICATION USING VARIOUS CNN ARCHITECTURES FEATURES AND MACHINE LEARNING ALGORITHMS," *Int. J. 3D Print. Technol. Digit. Ind.*, vol. 7, no. 2, pp. 277–285, Aug. 2023, doi: 10.46519/ij3dptdi.1268605.

[25] M. Yousefzadeh, S. A. Hosseini, and M. Farnaghi, "Spatiotemporally explicit earthquake prediction using deep neural network," *Soil Dyn. Earthq. Eng.*, vol. 144, p. 106663, 2021.

[26] E. Dağlı, M. Büber, and Y. S. Taspınar, "Detection of accident situation by machine learning methods using traffic announcements: the case of metropol Istanbul," *Int. J. Appl. Math. Electron. Comput.*, vol. 10, no. 3, pp. 61–67, Sept. 2022, doi: 10.18100/ijamec.1145293.

[27] N. Karayiannis and A. N. Venetsanopoulos, *Artificial neural networks: learning algorithms, performance evaluation, and applications*, vol. 209. Springer Science & Business Media, 2013. Accessed: Mar. 26, 2025. [Online]. Available: https://books.google.com/books?hl=tr&lr=&id=K4XdBwAAQBAJ&oi=fnd&pg=PP11&dq=artificial+neural+networks+learning&ots=6QszbmigYv&sig=xlOD7bl1ui4AJ1ynPxQ5vSjL0js

[28] Y. S. Taspinar, M. Koklu, and M. Altin, "Acoustic-Driven Airflow Flame Extinguishing System Design and Analysis of Capabilities of Low Frequency in Different Fuels," *Fire Technol.*, vol. 58, no. 3, pp. 1579–1597, May 2022, doi: 10.1007/s10694-021-01208-9.

[29] F. Berisha, "Quality of the predictions: mean absolute error, accuracy and coverage," 2017, doi: 10.13140/RG.2.2.26870.65606.

[30] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, July 2021, doi: 10.7717/peerj-cs.623.

[31] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *Int. J. Forecast.*, vol. 32, no. 3, pp. 669–679, July 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[32] Y. S. Taspinar, I. Cinar, and M. Koklu, "Classification by a stacking model using CNN features for COVID-19 infection diagnosis," *J. X-Ray Sci. Technol.*, vol. 30, no. 1, pp. 73–88, Jan. 2022, doi: 10.3233/XST-211031.

[33] M. Koklu, I. Cinar, Y. S. Taspinar, and R. Kursun, "Identification of Sheep Breeds by CNN- Based Pre-Trained Inceptionv3 Model," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, June 2022, pp. 01–04. doi: 10.1109/MECO55406.2022.9797214.

[34] E. Harris, "Information Gain Versus Gain Ratio: A Study of Split Method Biases.," in *AI&M*, 2002. Accessed: Apr. 18, 2025. [Online]. Available: https://www.mitre.org/sites/default/files/pdf/harris_biases.pdf