



# Analysis of positive correlation in magnitude and time measurement for earthquake using electric signals

Oumaima Saidani<sup>1,2</sup> · Ramkumar Raja Manoharan<sup>3</sup> · Ahmed Samir Naje<sup>4</sup> · Ruby Mishra<sup>5</sup> · Arjun Subburaj<sup>6</sup> · Maheswari S<sup>7</sup> · Mohamed Yacin Sikkandar<sup>8</sup> · Sankar Ganesh Sundaram<sup>9</sup> · Regin Rajan<sup>10</sup> · Sudhakar Sengan<sup>11</sup> 

Received: 3 April 2021 / Accepted: 21 December 2021 / Published online: 8 January 2022  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

## Abstract

Earthquake Prediction has become a field of seismology concerned with the specification of time, location, and magnitude of earthquakes to take preventive measures that could help in worst-case scenarios, i.e., destruction of homes and lives. Short Term Earthquakes depends on anomalous events known as precursors that occur before an Earthquake. Precursors are considered a warning before an earthquake. This prediction system uses Earth's Electric Field Signal (EEFS) from Athens, Pyrgos, and Hios. Some examples are Ground Uplifting, Tilting, Emission of Radon Gas, Radio Waves, Magnetic Waves, and Earth's Electric. The EEFS can be considered a precursor to estimate the magnitude and time of a possible earthquake. A minute-by-minute reading of the Earth's Electric Field was taken, and few computations, models like ANN, SVM-ANN, and SVM-KNN, were applied. Few other models were created to estimate the time and magnitude of the earthquake. To discover an ideal model, results are compared without any constraints such as overfitting. The paper presents directions for estimating time and is directing for researchers to analyze in multiple dimensions.

**Keywords** Earthquake Prediction · Magnitude Estimation · Time Estimation · Ensemble Techniques · Artificial Neural Networks

Communicated by H. Babaie

✉ Maheswari S  
maheswari.s@vit.ac.in

✉ Sudhakar Sengan  
sudhasengan@gmail.com

Oumaima Saidani  
OCSaidani@pnu.edu.sa

Ramkumar Raja Manoharan  
rmanoharan@kku.edu.sa

Ahmed Samir Naje  
ahmednamesamir@yahoo.com

Ruby Mishra  
rubymishrafme@kiit.ac.in

Arjun Subburaj  
aj@sentialabs.io

Mohamed Yacin Sikkandar  
m.sikkandar@mu.edu.sa

Sankar Ganesh Sundaram  
ssganesa@gmail.com

Regin Rajan  
regin12006@yahoo.co.in

Nourah Bint Abdulrahman University, Riyadh 11564,  
Kingdom of Saudi Arabia

- 2 RIADI Laboratory, National School of Computer Science,  
University of Manouba, Manouba 2010, Tunisia
- 3 Department of Electrical Engineering, King Khalid  
University, Abha, Saudi Arabia
- 4 Collage of Water Resources Engineering, Al-Qasim Green  
University, Al-Qasim, Iraq
- 5 School of Mechanical Engineering, KIIT Deemed To Be  
University, Bhubaneswar, Odisha 751024, India
- 6 Tranxit Technology Solutions Private Limited,  
Chennai 600 006, Tamil Nadu, India
- 7 School of Computer Science and Engineering, VIT  
University, Chennai 600 127, Tamil Nadu, India
- 8 Department of Medical Equipment Technology, College  
of Applied Medical Sciences, Majmaah University,  
Al Majmaah 11952, Saudi Arabia
- 9 Department of Information Technology, Dr. N.G.P. Institute  
of Technology, Tamil Nadu, Coimbatore 641048, India
- 10 Department of Computer Science and Engineering,  
Adhiyamaan College of Engineering, Hosur, Tamil Nadu,  
India
- 11 Department of Computer Science and Engineering, PSN  
College of Engineering and Technology, Tirunelveli 627152,  
Tamil Nadu, India

<sup>1</sup> Information Systems Department, College  
of Computer and Information Sciences, Princess

## Introduction

Prediction of earthquakes can be classified into Short Term (SHT), Immediate Term (IMT), Long Term (LT). SHT requires non-seismic precursors for predicting earthquakes. It has been said that SHT is not achieved yet, according to Ghaedi and Ibrahim (2017) (Ghaedi et al. 2018). On the other hand, IMT uses CN, MSc, M8, M8S combined with precursors and seismicity monitoring. The standard procedure for these different algorithms uses generic concepts of pattern identification that allow dealing with several sets of EQ precursors and authorizes for regular seismicity monitoring and extensive testing of predictions, according to Uyeda et al. (2011). But based on the study mentioned by Ghaedi and Ibrahim (2017) and (Mallouhy et al. 2019). IMT estimations cannot prevent all damages that could be caused and further protect all human life. However, they may undertake specific affordable actions to decrease damage and losses and modify the post-disaster relief. Despite the strenuous efforts taken and the several models developed, no prosperous technique has been detected yet. Because of the randomization of EQs, it may not be possible to determine the exact location, magnitude, and time of the following fatal EQ. Concerning LT, though several models were designed, there isn't any prosperous technique discovered yet. An Machine Learning (ML) / Artificial Intelligence (AI) branch has proven some accuracy and is used in earthquake estimation. Yet, there isn't any beneficial approach to precisely detect Earthquakes yet. So, ML/AI can study the accuracy and predict the Earthquakes from existing data.

(Cicerone et al. 2009) listed a compilation of earthquake precursors that could potentially be used to predict earthquakes. They mentioned that electric and magnetic fields, gas emissions such as radon ultrasonic vibration models, could be used to predict earthquakes. (Hattori 2004) studied the Ultra-Low-Frequency (ULF) and cited that short-term earthquake prediction could be possible using ULF. (Ghosh et al. 2009) gave a brief review of the progress made in radon measurements in earth sciences in predicting earthquakes. Radon anomalies have been observed in the soil or spring before earthquakes had been studied. They even proposed models that tried to relate precursor time, epicentral distance, and earthquake magnitude. (Varotsos et al. 1986) came up with the VAN method, which measures the low-frequency electric signals. i.e., Seismic Electric Signals (SES). The idea of implementing the earth's electric field before an EQ for short-term EQ prediction was used in Greece by P. Varatsos. This is done by noting the potential change between the electric fields in East–West (E-W) and North–South (N-S) polarity gradients. Earthquake size can be described by intensity or magnitude. Distance from the hypocenter, intensity, and duration of the vibrations, the

varying characteristics are associated with new observations of earth behaviour together with other planets. Hence it becomes hard to use approaches like distance-based. The paper hence uses a Support vector machine for the analysis.

This study applied the same knowledge and calculated the potential change between the Electric fields in the E-W and N-S polarity gradients and further used it to design various models to estimate the earthquake's magnitude and time. Furthermore, it was used as a comparative study, which noted the accuracy and precision across the various models developed.

## Related works

Earthquake prediction could be made with the help of the calculation of magnitudes. Asim et al. (2017) studied earthquake prediction using ML techniques such as pattern recognition neural network, recurrent neural network, Random Forest (RF) and Ensemble of trees using LP Boost, which is a linear combination of many tree classifiers where each classifier was added iteratively to the set of selected classifiers until no tree was needed to be added (Adeli and Panakkat 2009). (Panakkat and Adeli 2009) tried to predict earthquakes with a probabilistic neural network. Their paper has applied Bayesian statistics and non-parametric density approximation to calculate and make a neural network model. There was accuracy for earthquakes in their model with a magnitude between 4.5 and 6.0, but they did not yield accurate results for magnitudes greater than 6.0. Zarou et al. (Zarour et al. 2012) used Earthquake Prediction using an artificial neural network, where they predicted earthquake magnitudes in the northern Red Sea region, such as the Gulf of Aqaba, Gulf of Suez, and the Sinai Peninsula. They used other forecasting methods such as moving average, standard distributed random predictor, and uniformly distributed random predictor (Freund et al. 2017).

They even used different statistical methods and data fitting, such as linear, quadratic, and cubic regression. The results showed that the neural network model provided a higher forecast accuracy than other proposed methods. The Neural network model was at least 32% better than other proposed methods. The neural network can capture non-linear relationships than statistical methods and other proposed methods (Varotsos et al. 1986).

(Vasti and Dev 2020) applied ML algorithms to analyze the earthquake data. (Zhou et al. 2019) analyzed earthquake detection using dictionary learning. (Galkina et al. 2019) surveyed to predict an earthquake using ML methods. Earthquake prediction could also be made using non-seismic precursors, which helped in Short-term earthquake prediction. (Sgrigna and Conti 2012) tried to find a deterministic approach to investigate earthquake predictions. They studied

the methodological aspects of damage prevention and prediction approaches. They also came up with an empirical approach to deterministic earthquake prediction based on medium-term and short-term ground and space precursory phenomena that have been given. (Moustra et al. 2011b) tried using Artificial Neural Networks (ANN) to predict the Earthquakes in Greece's region on different input data types. The model tried to predict the magnitude of the earthquake of the following day. (Xu et al. 2010) made use of a series of physical quantities measured by the DEMETER satellite, including Electron density, Electron temperature, ions temperature, and oxygen ion density, together with seismic belt information and then made sample sets for a back-propagation neural network. The neural network model was then to be used to conduct the prediction. (Jánský and Pasko 2018) tried to use Earthquake Lights as a precursor to predicting earthquakes. According to them, an Earthquake light occurs before, during, or after an earthquake (Alarifi et al. 2012; Moustra et al. 2011a) (Mallouhy et al. 2019).

(Grisoni 2017) theorized “Peroxy Defect Theory” to be capable of providing explanations for the multitude of pre-earthquake phenomena. They further examined different types of precursory signals that claim to have noticeable changes before an earthquake. (Ghosh et al. 2019) studied the possibility of lower ionospheric anomalies in very low-frequency earthquakes. However, they couldn't find a reason why lithospheric variabilities relate and result in ionospheric irregularities. Furthermore, using analytical theory, they created a simulation for their study and found that the possible setup for explaining EQL would be when the source current dipole's upper pole is shifted close to the Earth's surface. They also found that the VLF (Vasti and Dev 2020) wave propagation study could help understand the cause-and-effect scenario of seismo ionospheric coupling. The polarity gradients primarily relate to the earth parameters for associating the magnitude. Algorithms are available for finding the fractional variation in the magnitude. The classification is also related to the gradients.

Another way for earthquake prediction is using a hybrid technique, *i.e.*, combining different earthquake prediction methods. (Astuti et al. 2014) came up with an earthquake prediction technique involving a combination of the Singular Value Decomposition (SVD) technique for feature extraction and then used Support Vector Machines (SVM) to classify the EQ. (Zhou et al. 2017) proposed a system that combined SVM and neural networks. Its experimental results showed that the combined SVM and a neural network algorithm showed better predictability than the traditional SVM or neural network. The neural network, on its own, resulted in overfitting or underfitting. But the combination of SVM and neural networks improved that disadvantage.

(Saba et al. 2017) proposed an earthquake prediction technique using Bat Algorithm and Feed Forward Neural Network (FFNN). The Bat algorithm was used to train the weights, and the FFNN was used to predict future earthquakes based on past input data. Their experiment results showed that the proposed approach was comparable and more stable than the Back Propagation Neural Network. (Asim et al. 2018) proposed an earthquake classification system that combined Support Vector Regressor and Hybrid Neural Network to predict the earthquake. The HNN was a combination of three different Neural Networks, supported by an Enhanced Particle Swarm Optimization (EPSO), which offered weight optimization at each layer. Their numerical result showed improved prediction performance for all the considered regions than previous prediction studies (Jánský and Pasko 2018).

The reason for choosing the classifiers and algorithms is outlined below:

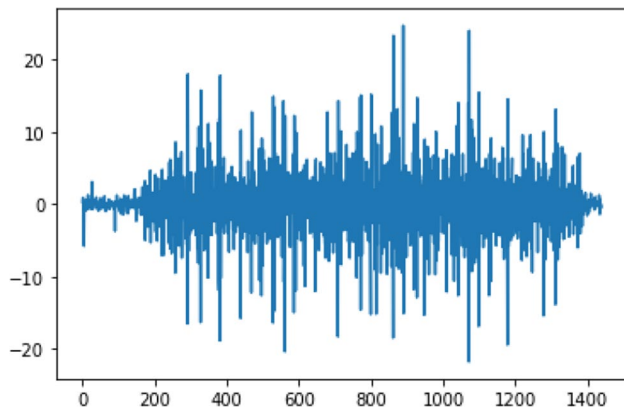
- a) Since the earthquake data can be time-varying, Neural Networks is chosen
- b) The features are numeric, and hence K-Means approach is chosen. It is better as it provides scope for choosing the value of K using the Elbow method.
- c) The approaches are mainly chosen to result in outliers, and hence the performance will not degrade to a new dataset.

## Implementation

This section elaborates on implementing the system's magnitude estimation and time estimation design with Earth's Electric Field.

### Preprocessing of the electric field signals

In this study, EEFS are taken from Athens (ATH) for the years 2004–2011, Pyrgos (PYR) for the years 2004–2011, and Hios (HIO) for the years 2007–2008 as the data to be preprocessed for the designing of the dataset. The years 2003 (for ATH), 2003 and 2012 (for PYR), and 2006 and 2009 (for HIO) is preprocessed and is kept separately for additional testing (Validation testing dataset) model after the model has been designed to get an unbiased estimate of the accuracy and precision. The usage of these testing data will be further elaborated in the subsequent sections. These datasets were collected from [www.earthquakeprediction.gr](http://www.earthquakeprediction.gr), which collected the EEFS from the three sites for research purposes.



**Fig. 1** GfDiff of January 1, 2005, from the Athens monitoring site

In the files used for training, each year has 365/366 days. Therefore, there is a total of 365/366 files for each year. Each file represents a day in a particular year. The file's data contains a minute-by-minute reading of the electric field signal, which counts to a total of 1440 samples per file. There are five columns in each file, which is described as shown below (as explained by [www.earthquakeprediction.gr](http://www.earthquakeprediction.gr)):

- 1) First Column – Time in the hour: minute format (hh: mm)
- 2) Second Column – EEFS data along the E-W direction
- 3) Third Column – Ignored
- 4) Fourth Column – EEFS data along the N-S direction
- 5) Fifth Column – Ignored

So, the first step to the preprocessing is to calculate the EEFS ( $|\vec{E}|$ ) each minute by adding the EEFS squares, E-S polarity, and N-S polarity and square rooting the summation. The formula is as shown below EQU (1):

$$|\vec{E}| = \sqrt{E_{EW}^2 + E_{NS}^2} \quad (1)$$

Once this formula is applied to all the data, the EEFS (GfDiff) difference is calculated each day. The formula for the same is as shown below EQU(2):

$$\Delta E = E[n] - E[n - 1] \quad (2)$$

After applying the formula, there would be a total of 1439 readings per day. The GfDiff of a day can be seen in Fig. 1. From these data, the peak value is found from each day, and the highest value and the date are recorded for further processing to create the Earthquake Dataset with which the various models are created for estimating the magnitude and time separately.

**Table 1** The Earthquake Dataset retrieved after the necessary extraction

Date	Latitude	Longitude	Magnitude
7/2/2004	36.04	26.91	5
1/3/2004	37.19	22.14	5
4/8/2004	36.92	27.7	5.4
4/8/2004	36.98	27.69	5.1
7/10/2004	36.5	26.81	5.2
4/11/2004	35.86	23.23	5
11/1/2005	37.01	27.85	5
31/1/2005	37.41	20.11	5.7
17/10/2005	38.13	26.59	5.5
17/10/2005	38.14	26.59	5.4
18/10/2005	37.58	20.86	5.6
20/10/2005	38.15	26.63	5.6
25/11/2005	35.02	23.32	5.2
8/1/2006	36.21	23.41	6.4
4/4/2006	37.58	20.93	5.2
11/4/2006	37.64	20.92	5.2
11/4/2006	37.68	20.91	5.4
12/4/2006	37.61	20.95	5.4
3/2/2007	35.8	22.58	5
25/3/2007	38.34	20.42	5.5
29/6/2007	39.25	20.26	5.2
23/9/2007	35.27	27.12	5.1
9/11/2007	38.77	25.77	5.1
6/1/2008	37.11	22.78	6.1
4/2/2008	38.09	21.94	5
14/2/2008	36.5	21.78	6.2
14/2/2008	36.22	21.75	6.1
19/2/2008	36.19	21.77	5.1
20/2/2008	36.18	21.72	6
26/2/2008	35.96	21.7	5.2
19/3/2008	38.92	24.17	5
28/3/2008	35.01	25.33	5.1
10/5/2008	36.3	22.24	5.1
8/6/2008	37.98	21.51	6.5
12/6/2008	35.11	26.19	5
18/6/2008	37.67	22.78	5.1
21/6/2008	36.03	21.83	5.5
15/7/2008	35.85	27.92	6.2
30/7/2008	38.02	20.12	5
3/8/2008	39.59	23.9	5.3
14/10/2008	38.85	23.62	5.6
14/10/2008	38.87	23.61	5.1
13/12/2008	38.72	22.57	5.2
13/1/2009	35.66	26.39	5.2
16/2/2009	37.13	20.78	5.5
26/6/2009	36.53	25.49	5
3/11/2009	37.39	20.35	5.6
11/11/2009	37.47	20.47	5.3
17/1/2010	35.26	27.86	5

**Table 1** (continued)

Date	Latitude	Longitude	Magnitude
18/1/2010	38.41	21.95	5.2
22/1/2010	38.42	21.97	5.1
9/3/2010	38.87	23.65	5.1
16/7/2010	36.81	26.91	5.1
16/7/2010	39.32	24.02	5.1
17/12/2010	36.94	24	5.1
1/4/2011	35.64	26.56	6.2

**Collecting the earthquake data**

A dataset from Kaggle was used for collecting the data required to create the final dataset on which the models are applied. In this dataset collected from Kaggle, the dataset has 8 columns as described below:

- 1) First Column: Year (yyyy)
- 2) Second Column: Month
- 3) Third Column: Date
- 4) Fourth Column: Hours [Column has been ignored]
- 5) Fifth Column: Minutes [Column has been ignored]
- 6) Sixth Column: Latitude
- 7) Seventh Column: Longitude
- 8) Eight Column: Magnitude (Richter scale)

The dataset has all the Earthquake recordings for Greece from the year 1901–2018. Out of this, for the final dataset to be designed, Earthquakes from the year 2004–2011 have been extracted with a magnitude greater than 5.0, latitude ranging from 35–39.9, and a longitude ranging from 20–27.9. The data retrieved of the dates an Earthquake takes place can be seen in Table 1.

The date is related to the observed Date; the other parameters, namely latitude, longitude, and magnitude, are related to the earth's movements. It is a standard dataset, and the

ground truth of the values and constraints exists. Hence no preprocessing is performed on the dataset.

The features do not present any constraint in terms of noise and values. There is no feature selection approach followed in the paper. However, in the future Principal Component Analysis can be used.

**Data pruning**

After retrieving dates of when an earthquake took place between 2004 and 2011, an algorithm is designed to extract the dates that have a significant GfDiff value days before the earthquake such that the days are 30 days or less. The algorithm is explained using the flowchart in Fig. 2.

After finding the significant GfDiff that occurs 30 days or less, that Date is recorded, and the number of days between the dates is recorded. The final dataset designed can be described as the following:

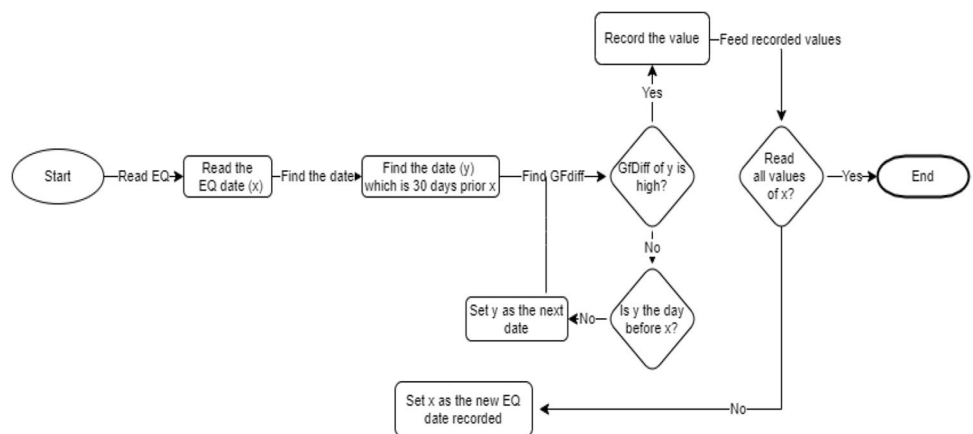
- 1) First Column: Date of the EQ
- 2) Second Column: Date Prior EQ with Significant GfDiff
- 3) Third Column: No. of Days
- 4) Fourth Column: Latitude
- 5) Fifth Column: Longitude
- 6) Sixth Column: Magnitude

The total number of readings that have been recorded from the three sites is mentioned below:

- ATH: 56 Readings
- PYR: 56 Readings
- HIO: 25 Readings

All the above Readings have been combined into one dataset, having a total of 137 readings. The first 20 rows from the dataset are visible in Table 2. Following the dataset's formation, the dataset was split into training and testing

**Fig. 2** Flow diagram of identifying and recording the significant GfDiff that occurs before an Earthquake





**Table 2** Table Displaying the First 20 Readings from the Final Dataset That Was Formed

Date of EQ	Date with Significant GfDiff	No. of Days	Latitude	Longitude	Magnitude
7/2/2004	14/1/2004	24	36.04	26.91	5
1/3/2004	7/2/2004	23	37.19	22.14	5
4/8/2004	26/7/2004	9	36.92	27.7	5.4
4/8/2004	26/7/2004	9	36.98	27.69	5.1
7/10/2004	28/9/2004	9	36.5	26.81	5.2
4/11/2004	26/10/2004	9	35.86	23.23	5
11/1/2005	16/12/2004	26	37.01	27.85	5
31/1/2005	21/1/2005	10	37.41	20.11	5.7
17/10/2005	24/9/2005	23	38.13	26.59	5.5
17/10/2005	24/9/2005	23	38.14	26.59	5.4
18/10/2005	24/9/2005	24	37.58	20.86	5.6
20/10/2005	24/9/2005	26	38.15	26.63	5.6
25/11/2005	19/11/2005	6	35.02	23.32	5.2
8/1/2006	2/1/2006	6	36.21	23.41	6.4
4/4/2006	1/4/2006	3	37.58	20.93	5.2
11/4/2006	6/4/2006	5	37.64	20.92	5.2
11/4/2006	6/4/2006	5	37.68	20.91	5.4
12/4/2006	6/4/2006	6	37.61	20.95	5.4
3/2/2007	15/1/2007	19	35.8	22.58	5
25/3/2007	13/3/2007	12	38.34	20.42	5.5

**Table 3** Earthquake Magnitude Classes

Class	Magnitude
Great	8 or more
Major	7 – 7.9
Strong	6 – 6.9
Moderate	5 – 5.9

**Table 4** Classification Table for Time Estimation

Class	Time
0	One Week (1–7 Days)
1	Two Weeks (8–14 Days)
2	Three Weeks (15–21 Days)
3	Four Weeks (22 or more Days)

datasets with a test size of 33%. This was stored separately to be implemented on the models, i.e., a formal training and testing set was used for designing the models. These models were used to estimate the magnitudes and estimate the time. This will be further elaborated using in the following subsections.

Figure 2 has start and End, indicating the beginning and end of the model. The flow paths are connected in operation, and there is no associated condition. The conditional aspects of Yes/No have already been labelled on paper.

### Magnitude estimation

The magnitudes can be classified, as shown in Table 3. On studying the dataset on which the models would be training and testing on, it was noticed that the number of reading with a magnitude greater than 6 seemed to be small. This seemed to affect the models' accuracy, which was designed as there were, resulting in an ill-defined F1-score.

Hence, during the implementation of magnitude estimation, the model tried two separate classification classes. The first one followed the classification as mentioned below:

- 5 – 5.9 – Class 0
- 6 – 6.9 – Class 1
- 7 – 7.9 – Class 2
- 8 or more – Class 3

However, the second classification classes were designed, as shown below:

- 5.0 – 5.5 – Class 0
- 5.6 – 5.9 – Class 1
- 6.0 – 6.5 – Class 2
- 6.5 Or more – Class 3

This was done to study any differences in accuracy and precision when testing the model's predictability. This will be elaborated on further in the subsequent sections.

### Time estimation

To implement this, the dataset was classified into four classes, as seen in Table 4. This classification was used in training the various models.

As seen in Table 4, the dataset observes and defines the class according to the number of days. For instance, when we refer to Table 2, the first reading stated that the number of days between the Date of EQ and the Date with significant GfDiff before an EQ is 24 days; these places this reading into class 3. Similarly, all the readings have been given their class, and then the dataset is divided into training and testing data on which the models are trained. The accuracy, precision, and observations made have been further elaborated in the subsequent sections. The dataset does not have any errors. The round-off operation alone is performed on the data for algorithms to work.

Table 4 defines the class level rules. It is required for classifying the earthquake features. It is related to Table 3 as it forms the basis for classifying the earthquake data. More details on the same are available in EQ (Zhou et al. 2017).

### Artificial Neural Network

By definition, ANN can be considered an information processing model inspired by humans' biological nervous

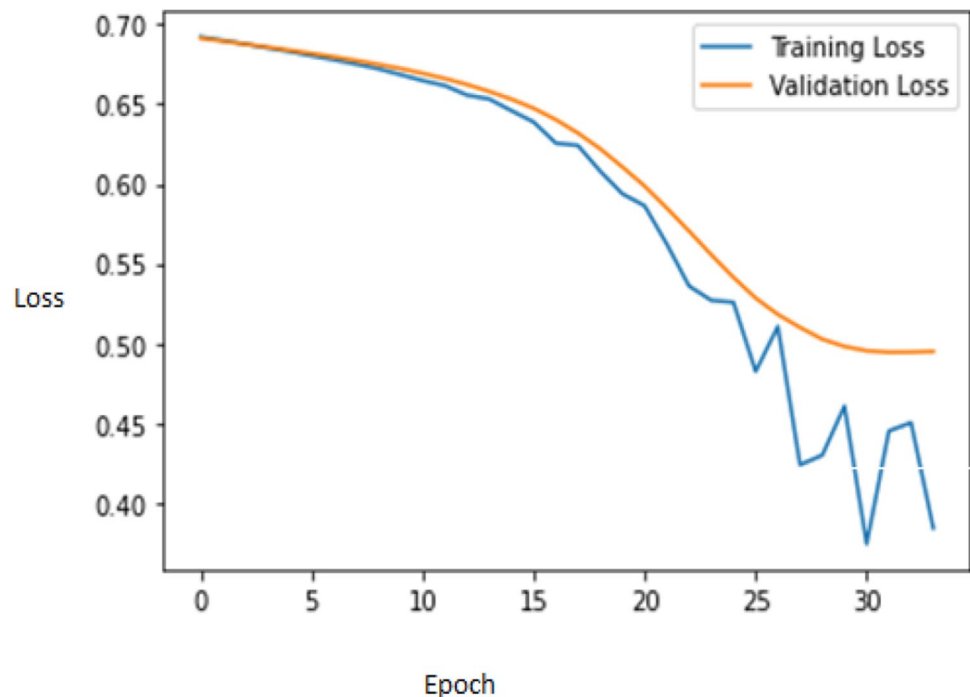
system, just like how the brain processes information. It comprises highly interconnected processing elements (i.e., neurons) to find a pattern to solving a problem.

In this study, two models of ANN are designed. One is to estimate the magnitude; the other is to estimate the time. To create this network, a high-level API, Keras, is used to design the model. Both models are similar in design. There are 4 input dimensions in the input layer. There are three hidden layers, and finally, an output layer of one unit with sigmoid as the activation function. Sometimes, overfitting can be a problem in neural networks. One of the steps taken to avoid this is adding a dropout, a regularization technique to reduce overfitting by changing the network. When fitting the model, the epochs are set to 500. Another form of overfitting is memorization, which could be possible since the entire dataset is passed forward and backwards through the neural network for each epoch.

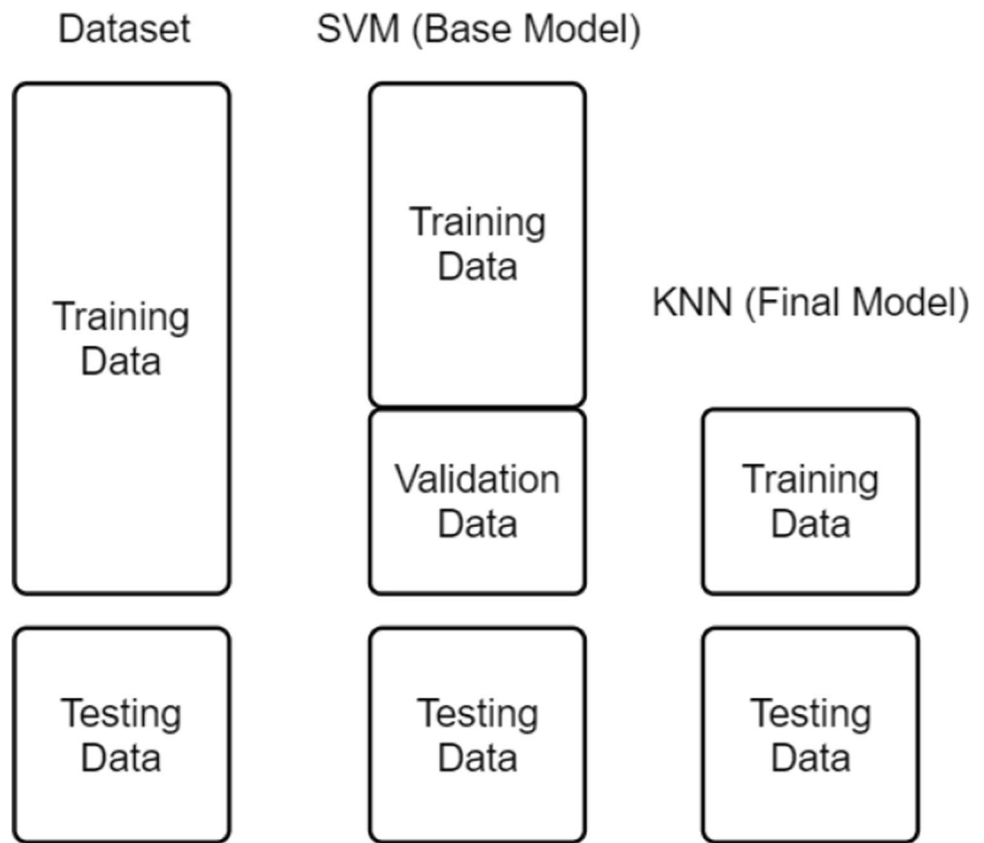
Another point that needs to be kept in mind is that underfitting should not happen either. To check this, the training and validation loss during each iteration could be noted. There could be overfitting if the training loss is less than the validation loss, underfitting if the training loss is more significant than validation loss. The 'just right' is possible when the training and validation loss are almost similar. (Fig. 3).

The x-axis represents the epochs, and the y-axis represents the measured loss. So, in order to achieve this, the iterations of the epochs need to be controlled. If the number of epochs required for fitting the model is low, but the number of epochs set is like 500, there could be overfitting or underfitting. So even after regularization was

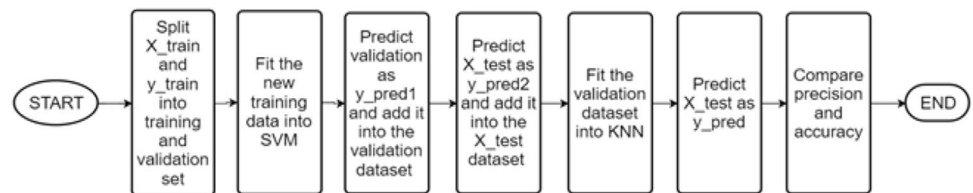
**Fig. 3** Training Loss and Validation Loss Plotted in a graph. Values are quite similar, so there is no overfitting or underfitting



**Fig. 4** Illustration of the Blending of the SVM-KNN Model



**Fig. 5** Schematic Diagram of the Blending of SVM and KNN



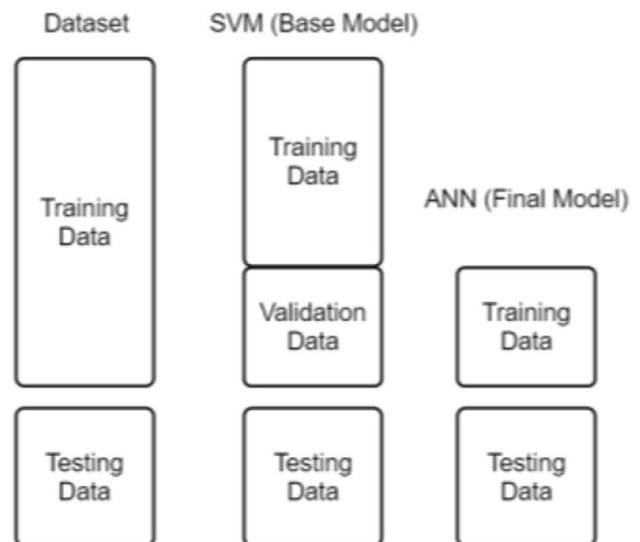
applied, early stopping was also set. It is also another form of regularization where the model stops iterating if it starts to degrade. As seen in Fig. 3, even though the number of epochs set is 500, the iteration stops after 30 because the performance will be underfitting if the iteration continues.

Thus, keeping all this in mind, the ANN was designed, and the accuracy and precision are observed and noted. This will be further elaborated in the subsequent sections.

**SVM-KNN**

In order to design this model, an advanced technique of ensemble learning is called blending, as shown in Fig. 4.

Here the training data is further divided into training data and validation data. The new training data is fitted into the SVM, base, and testing validation data. The previous testing data is also tested into the model. The



**Fig. 6** Illustration of the blending of SVM and ANN Models



prediction results from the validation testing and the testing set are added as features and the new dataset, as seen in Fig. 5.

As seen in Fig. 5, the features are added to the validation dataset and the testing test. The validation dataset becomes the new training data. This training data is then fit into the KNN model. The testing data is applied to the prediction model, and the accuracy and prediction are observed.

The dataset is splatted into Training, Validation, and Test Data. The validation data is used for defining the parameters for the model. This is done as there could be a possibility of losing some classes, leading to the ill-definition of individual classes.

**SVM-ANN**

This model is also similar to the SVM-KNN model, as seen in Fig. 6. Here, the SVM model will be the base model, and ANN will be the final model. The schematic diagram (Fig. 5) explains how the process has been implemented. In ANN, the possibility of overfitting and underfitting needs to be taken into account. So, in order to prevent the same, regularization by adding dropout, which modifies the neural network, is added. Another method applied here is the early stopping, which stops the iteration when training and validation loss differ significantly.

The ANN has an input layer with the input dimension set to 5 as the prediction result from SVM is added into the validation dataset, thus increasing the size of the input dimension by a unit. There are three hidden layers in the neural network, and the output layer has only one output,

which uses the activation function sigmoid. The accuracy and precision of this model are observed and elaborated on in the subsequent sections.

A drawback is that in the final model is that the dataset is changed, as mentioned previously in 3.7. This is mainly because a small portion of the training data, i.e., the validation data, is used as the final model's training data, i.e., ANN. There could be a possibility of losing some classes, which further leads to the ill-definition of individual classes.

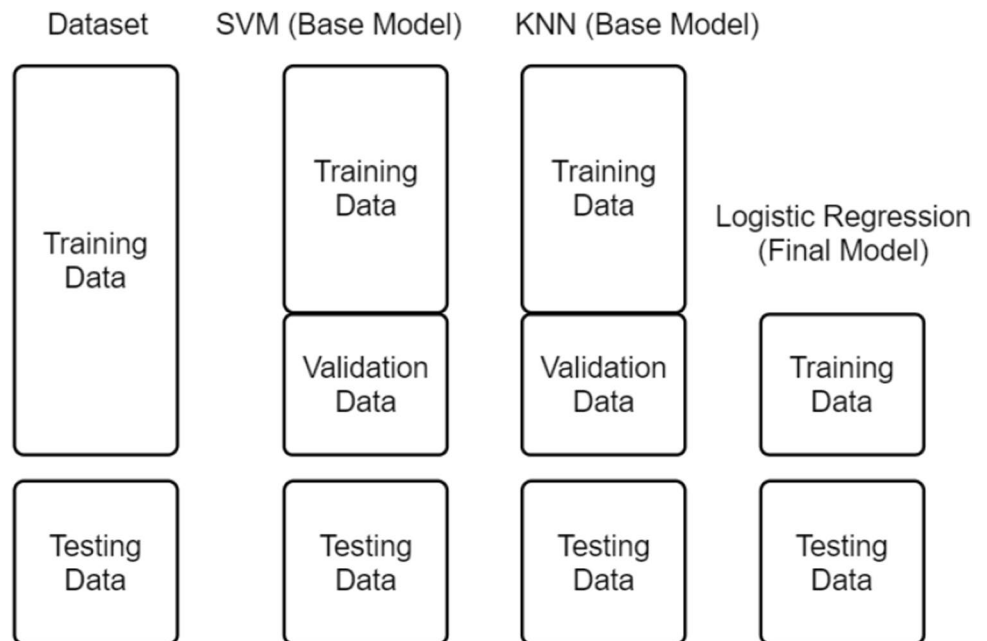
**SVM-KNN-Logistic Regression**

This model uses SVM and KNN as the base class and then Logistic Regression as the final model. An illustration of how it works is depicted in Fig. 7.

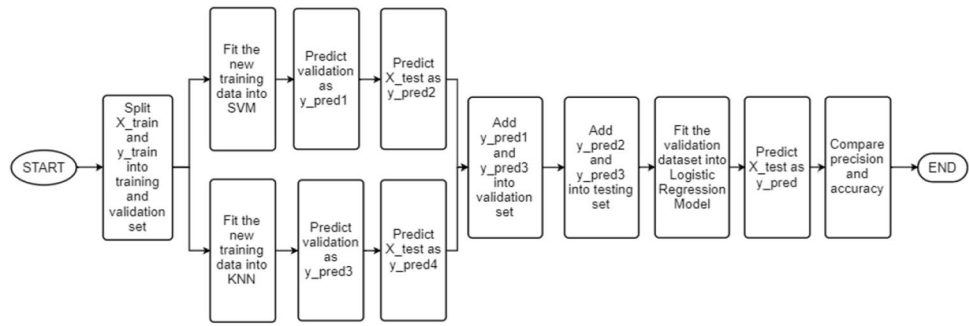
As shown in Fig. 7, the dataset is split into training and testing data. This training data is further split into training and validation data. This new training data is fitted into the SVM model, and then this model is tested using the validation data. The results yielded from the prediction of the validation data are recorded. After this, the new training data is fitted into the KNN, a base model. The KNN model is tested with the help of the validation data. The results yielded from the prediction of the validation data are also recorded.

In the previous case, only the SVM validation data's predicted results were added into the validation data, but in this case, the prediction results of both SVM and KNN are added. These recorded results from the SVM prediction and the KNN are added to the validation data, thus increasing the input dimensionality by 2. The original testing data is also predicted by the base models SVM and KNN. The results yielded from the prediction are added to the original testing

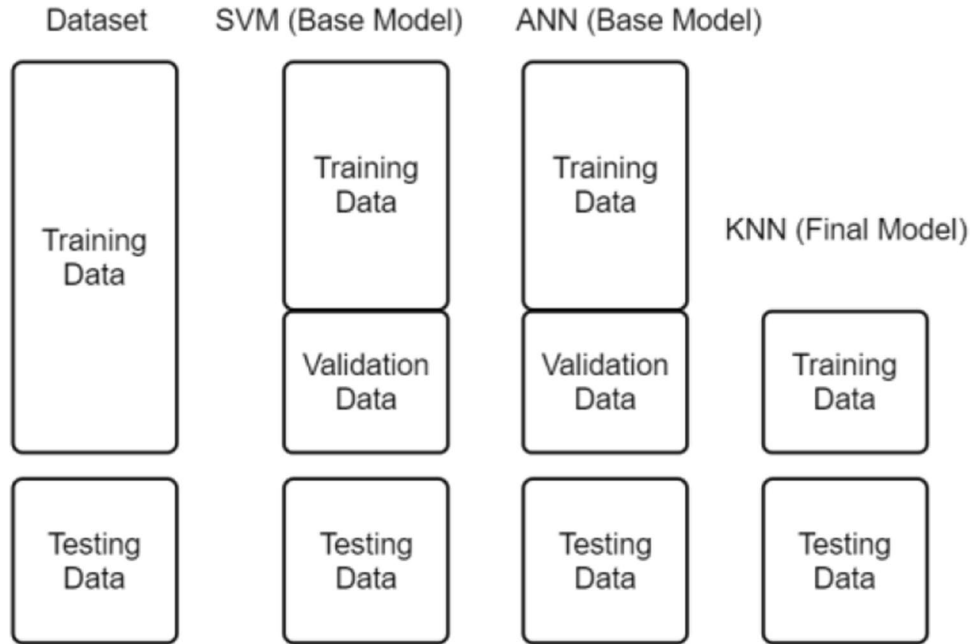
**Fig. 7** Illustration of the Blending of SVM, KNN, and Logistic Regression



**Fig. 8** Schematic Diagram of the Blending of SVM, KNN, and Logistic Regression



**Fig. 9** Illustration of the Blending of SVM, ANN, and KNN



data, which results in an increase in the dimensionality by two. This testing data is used to test the final model. The validation data is now the training data, fitted into the final model, the Logistic Regression model. This model is then tested with the original testing data, and the accuracy and the precision in the predictability are observed. This entire process is illustrated in the schematic diagram (Fig. 8). It can be observed that this process is similar to the process of blending SVM and ANN or even SVM and KNN, which was illustrated in Fig. 5.

As mentioned before, in Sect. 3.7 and Sect. 3.8, a drawback is that in the final model is that the dataset is changed. This is mainly because a small portion of the training data, i.e., the validation data, is used as the final model's training data, i.e., ANN. There could be a possibility of losing some classes, which further leads to the ill-definition of individual classes. This issue is usually prevalent when the number of training and testing data is not so huge. Since the dataset only contains 137 readings, this issue could be seen.

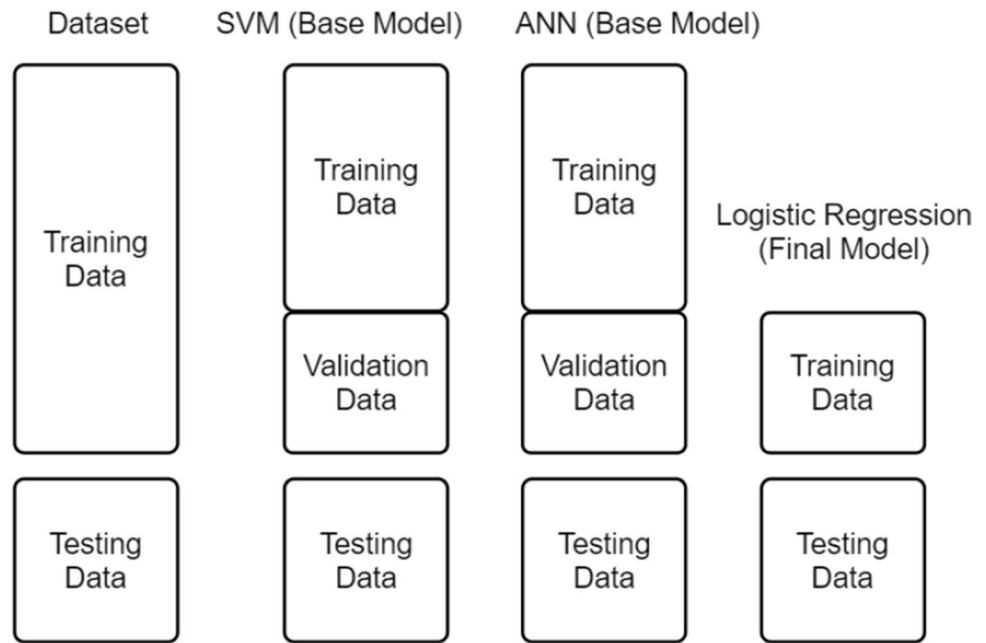
The dataset's size could not be increased as there were not enough EEFS Readings to increase the dataset's size.

**SVM-ANN-KNN**

As shown in Fig. 9, similar to Sect. 3.9, the dataset is divided into training and testing datasets with a test size of 33%. The training data is divided into training and validation data, with a test size of 33%. This new training data is fitted into the base model SVM, and thus the SVM model is created, then the same new training data is fitted into the following base model, ANN.

Here, the ANN model contains an input layer with an input dimension of 4, there are 3 hidden layers, and finally, the output layer has only one output with the activation function sigmoid. Here, overfitting and underfitting are taken into account. So, to prevent the same, early stopping stops the iterations when a vast difference between the training and validation loss exists and dropout layers have been added to modify and regularize the neural network.

**Fig. 10** Illustration of the Blending of SVM, ANN, and Logistic Regression



After the training data is fitted into the base models, the validation data is used to test the model's predictability. The results obtained from the base models are added to the validation data, thus increasing the dimensionality by two units. The original test data is also used for the base models, and the results are added to the test data. The new validation data becomes the training data for the KNN model; the new test data is used to test the model's predictability. Then the accuracy and precision are observed and elaborated in subsequent Sections.

The Schematic Diagram in Fig. 8 can explain this process, but the difference is that instead of KNN being a base model, it is the final model and the base model is ANN and SVM instead. The drawbacks are the same as mentioned in Sect. 3.10 and Sect. 3.9, i.e., the loss in classes that affect the precision, accuracy, and F1-scores, making them ill-defined.

**SVM-ANN-Logistic Regression**

As seen in the illustration above (Fig. 10), the dataset is divided into training and testing data. This training data is further split into training and validation data.

This new training data is fitted into the base models SVM and ANN. Then the validation data is tested on these base models. The results obtained from these base models' predictions are added to the validation data, increasing the dimensions of the two units. The same is applied to the testing data, and the results are added to the testing dataset.

Here, the ANN model contains an input layer with an input dimension of 4, there are 3 hidden layers, and finally, the output layer has only one output with the activation function sigmoid. Here, overfitting and underfitting are taken

into account. To prevent the same, early stopping stops the iterations when a considerable difference between the training and validation loss exists and dropout layers have been added to modify and regularize the neural network.

The validation data is used as the training dataset on the final model with Logistic Regression. Further testing data is used to test the predictability. Finally, after results from the prediction of the validation data and the base models' testing data and the results being added to the validation data and testing data, respectively. Then the accuracy and precision are observed and elaborated in subsequent sections.

The Schematic Diagram in Fig. 8 can explain this process, but the difference is that instead of being a base model, the base model is ANN and SVM instead, with the final model remaining the same. The drawbacks are the same as Sect. 3.10 and Sect. 3.9, i.e., the loss in classes that affect the Precision, Accuracy, and F1-scores, making them ill-defined.

**Performance evaluation**

This section discusses the evaluation of the performance of the models designed and tested. A validation dataset had also been created for unbiased observation and estimation of the model. The data is collected from the sites Athens (ATH), Pyrgos (PYR), and Hios (HIO). These years contain inconsistent data as some of the days' readings of the EEFS is not available. The years of data taken from the sites and the available range of dates are mentioned below:

**Table 5** A dataset that was used for Validation Testing

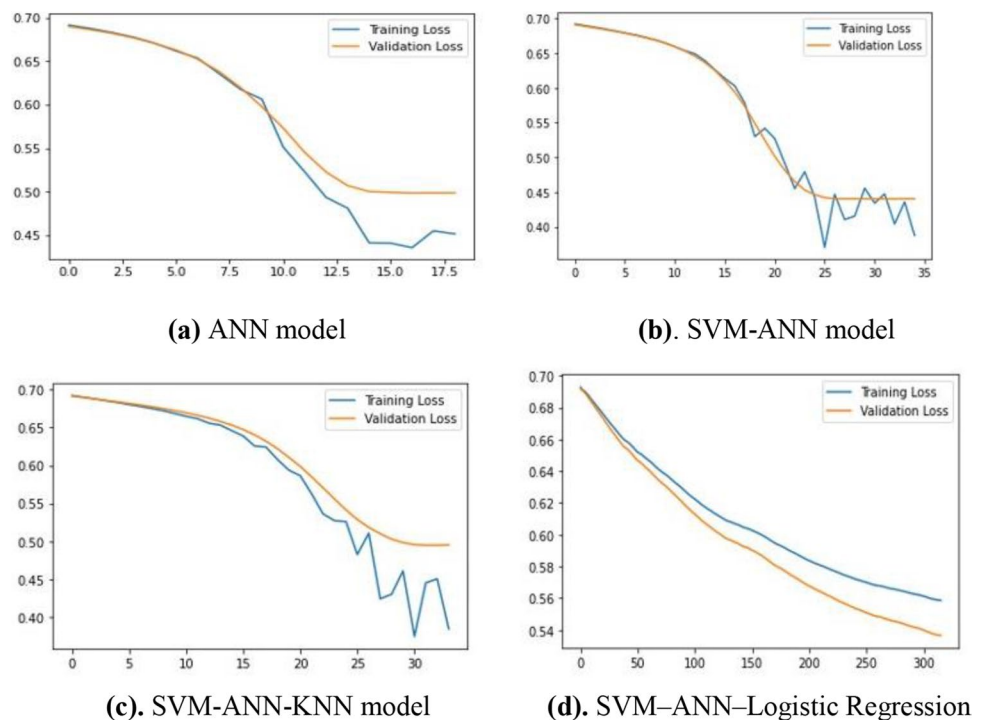
Date of EQ	Date with Significant GfDiff	No. of Days	Latitude	Longitude	Magnitude
17/4/2004	16/4/2004	1	38.19	26.9	5
29/4/2004	20/4/2004	9	36.83	21.72	5
9/6/2004	21/5/2004	19	39.94	22.35	5
14/8/2004	31/7/2004	14	38.79	20.56	5.9
14/8/2004	31/7/2004	14	38.76	20.67	5.1
14/8/2004	31/7/2004	14	38.76	20.67	5.2
17/10/2004	20/9/2004	27	35.96	22.25	5.3
4/4/2006	23/3/2006	12	37.58	20.93	5.2
11/4/2006	23/3/2006	19	37.64	20.92	5.2
11/4/2006	23/3/2006	19	37.68	20.91	5.4
12/4/2006	11/4/2006	1	37.61	20.95	5.4
13/1/2009	2/1/2009	11	35.66	26.39	5.2
16/2/2009	29/1/2009	18	37.13	20.78	5.5
26/6/2009	9/6/2009	17	36.53	25.49	5
3/11/2009	10/10/2009	24	37.39	20.35	5.6
11/11/2009	8/11/2009	3	37.47	20.47	5.3
26/1/2012	3/1/2012	23	36.06	25.07	5.3
27/1/2012	3/1/2012	24	36.06	25.13	5.2

- ATH – Contains data from the year 2003, from April 15 to December 31
- PYR – Contains data from the year 2003, from May 23 to December 31, and data from the year 2012, from January 1 to March 17
- HIO – Contains data from the year 2006, from March 18 to December 31, and data from the year 2009, from January 1 to November 11. But since this was used just for validation purposes, this issue was overlooked.

In this folder, specific dates were found to be missing.

The final validation dataset formed after the preprocessing of the EEFS, extraction of the earthquake dates, and

**Fig. 11** (a) ANN (b). SVM-ANN model (c). SVM-ANN-KNN model (d). SVM-ANN-Logistic Regression



combining the dates with the dates having a significant GfDiff before 30 days or less can be seen in Table 5.

### Magnitude Estimation type 1

The classification of the magnitude of type 1 is as follows:

- Class 0 – Magnitude between 5 and 5.9
- Class 1 – Magnitude between 6 and 6.9
- Class 2 – Magnitude between 7 and 7.9
- Class 3 – Magnitude 8 or more

The dataset was split into training and testing sets, and the models have trained accordingly. In the models which included an ANN, overfitting and underfitting had been taken into consideration. The number of epochs when training the model had been set to 500. A dropout layer has been added to regularize each layer. Additionally, there was an early stopping feature to prevent that the training and validation loss is similar. When the training loss is greater than the validation loss, then it is a case of underfitting. When the training loss is lesser than the validation loss, then it is a case of overfitting. The ‘just right’ condition would be when the training loss is similar. So, the first stopping feature has been implemented where the model will keep stop training when the performance ceases to improve. But, in some instances, there could be a chance for the model’s performance to worsen, and then it starts to improve. So, a patience argument has been set, and a callback feature saves the best model observed, and that model is loaded. The training and validation losses have been plotted in Fig. 11 (a) for the ANN model, Fig. 11 (b) for the SVM-ANN model, Fig. 11 (c) for the SVM-ANN-KNN model, Fig. 11 (d) for the SVM-ANN-Logistic regression, along with the number of epochs used while training the data for Magnitude Estimation of type 1.

On the other hand, accuracy checks the percentage of accurate predictions, and hence as shown in Fig. 11 (a), (b), (c), and (d) early stopping, and callback helped create a model with the training and validation loss as minimal as possible. Thus, with the two’s help, ANN stops training after 19 epochs; SVM-ANN, after 316 epochs; SVM-ANN-KNN, after 34 epochs and finally, SVM-KNN-Logistic regression,

after 35 epochs. The error measurements help estimate the performance of a model. In order to comparatively analyze the performance of the models created, accuracy, macro-precision, and macro-F1 score are analyzed. Precision measures the number of relevant results returned than the irrelevant ones. But precision doesn’t necessarily mean that the model is good depending on the precision alone. It is not the best performance measurement. F1-score, on the other hand, is a harmonic mean between precision and recall. Since the label sizes are not balanced in the dataset, macro-precision and macro F1-scores are analyzed. Macro measurements ensure that there is a bias against the least popular labels. The performance of the models of the magnitude estimation of type 1 can be seen in Table 6. Further inferences are also presented as below:

- 1) The original dataset does not have the class label. The class label learned with unsupervised learning allows for classifying the Earthquake dataset. It supports the class label feature as part of the training dataset.
- 2) Hence during testing, the dataset is predicted with a suitable class label using ANN.

Since ANN is used, the learning rate is related to the Error learned during the training process. It will allow for reducing the loss further.

On checking the testing data’s performance, it can be observed that for the magnitude estimation of type 1, the SVM-ANN-Logistic Regression seems to have better performance in predicting the magnitude than the other models. While on the other hand, all the other models seem to have the same performance result. This could be mainly because of the unbalance label size, i.e., the amount of data of Class 1 must be less than Class 0, due to which there is an ill-definition in the prediction of samples. However, the validation test’s performance evaluation was observed to have 100% accuracy. When one looks at the validation data set (Refer Table 5), by looking at the magnitude, we can see that fall in the Class 0 category is based on the magnitude classification of type 1. Thus, validation testing is not helpful in the performance evaluation of the magnitude classification of type 1.

**Table 6** Performance Evaluation of The Testing Data on The Different Models Designed for Magnitude Estimation of type 1

Model	Accuracy	Macro-Precision	Macro-F1 Score
ANN	0.80	0.40	0.45
SVM-KNN	0.80	0.40	0.45
SVM-ANN	0.80	0.40	0.45
SVM-KNN-Logistic Regression	0.80	0.40	0.45
SVM-ANN-KNN	0.80	0.40	0.45
SVM-ANN-Logistic Regression	0.89	0.94	0.78

### Magnitude Estimation type 2

The classification of the magnitude of type 2 is as follows:

- Class 0 – Magnitude between 5 and 5.9
- Class 1 – Magnitude between 6 and 6.9
- Class 2 – Magnitude between 7 and 7.9
- Class 3 – Magnitude 8 or more

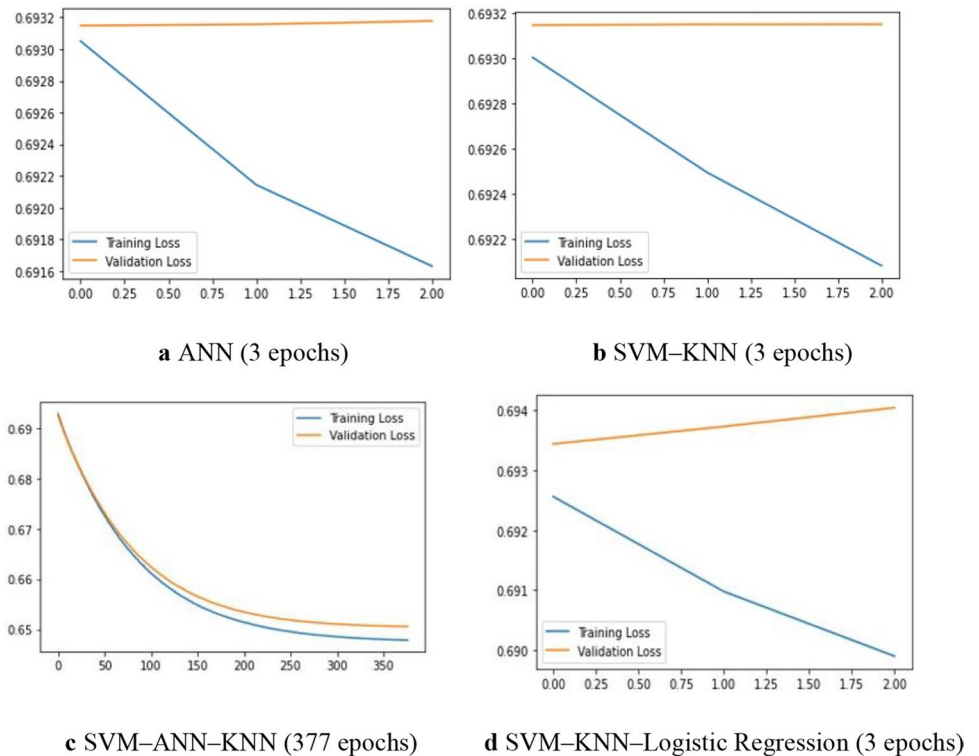
Magnitude Estimations of type 2 are similar to that of type 1. The only difference lies in the classification of the classes. Thus, when training the models with ANN, the number of epochs while fitting the model changes, and the performance changes. In these models, dropout layers have been added to regularize the neural networks. The number of epochs when training the model had been set to 500. In order to avoid overfitting and underfitting, early stopping and callbacks have been implemented. The primary function of early

stopping is to stop the training process when the model’s performance worsens. When the training loss is greater than the validation loss, then it is a case of underfitting.

When the training loss is lesser than the validation loss, then it is a case of overfitting. The ‘just right’ condition would be when the training loss is similar. So, the first stopping feature has been implemented where the model will keep stop training when the performance ceases to improve. But, in some instances, there could be a chance for the model’s performance to worsen, and then it starts to improve. So, a patience argument has been set, and a callback feature saves the best model observed, and that model is loaded.

The training loss versus validation loss graphs of the models involving an artificial neural network, i.e., ANN, SVM-ANN, SVM-ANN-KNN, and even SVM-ANN-Logistic regression, can be seen in Fig. 12 (a), (b), (c), and (d) along with the number of epochs used while training the data for Magnitude Estimation of type 2. As shown in Fig. 12 (a),

**Fig. 12** a ANN (3 epochs) b SVM–KNN (3 epochs) c SVM–ANN–KNN (377 epochs) d SVM–KNN–Logistic Regression (3 epochs)



**Table 7** Performance Evaluation of The Testing Data on The Different Designed for Magnitude Estimation of Type 2

Model	Accuracy	Macro-Precision	Macro-F1 Score
ANN	0.72	0.24	0.28
SVM-KNN	0.72	0.24	0.28
SVM-ANN	0.72	0.24	0.28
SVM-KNN-Logistic Regression	0.76	0.58	0.40
SVM-ANN-KNN	0.67	0.23	0.27
SVM-ANN-Logistic Regression	0.96	0.98	0.91



(b), (c), and (d), early stopping and callback helped create a model with the training loss and validation loss as minimal as possible. Thus, with the help of the two, ANN stops training after 3 epochs; SVM-ANN, after 3 epochs; SVM-ANN-KNN, after 377 epochs and finally, SVM-KNN-Logistic regression, after 3 epochs.

The error measurements help estimate the performance of a model. In order to comparatively analyze the performance of the models created, accuracy, macro-precision, and macro F1-score are analyzed. Precision measures the number of relevant results returned than the irrelevant ones. But precision doesn't necessarily mean that the model is good depending on the precision alone. On the other hand, accuracy checks the percentage of accurate predictions, and hence it is not the best performance measurement. F1-score, on the other hand, is a harmonic mean between precision and recall. Since the label sizes are not balanced in the dataset, macro-precision and macro F1-scores are analyzed. Macro measurements ensure that there is a bias against the least popular labels. The performance of the models of the magnitude estimation of type 2 can be seen in Table 7.

The x-axis is varied with epochs. The y axis is associated with the rate of learning. From the performance evaluation of the testing data, a difference can be seen in the performance compared to the previous Magnitude Estimation models. Here, ANN, SVM-KNN, and SVM-ANN seem to have the same performance, while SVM-KNN-Logistic Regression, SVM-ANN-KNN, and SVM-ANN-Logistic

Regression have different performances compared to the other three. Here, the worst performance is shown by SVM-ANN-KNN, while on the other hand, SVM-ANN-Logistic regression seems to be the best performing model for both types of Magnitude classification.

On testing the performance of all the models with the validation dataset, all of the models seem to have the same percentage of performance, i.e., 0.89 accuracies, 0.44 macro-precision, and 0.47 macro F1-score except for SVM-KNN-Logistic regression, which differs in the micro-precision and so it has a micro-precision of 0.48. The possible reason why the performance is the same is that the validation data contains Earthquakes with a magnitude between 5 and 6 (Table 5). There are only two readings of class 1, and the balance of 16 belongs to class 0. Thus, performance evaluation on the validation test data set is less functional.

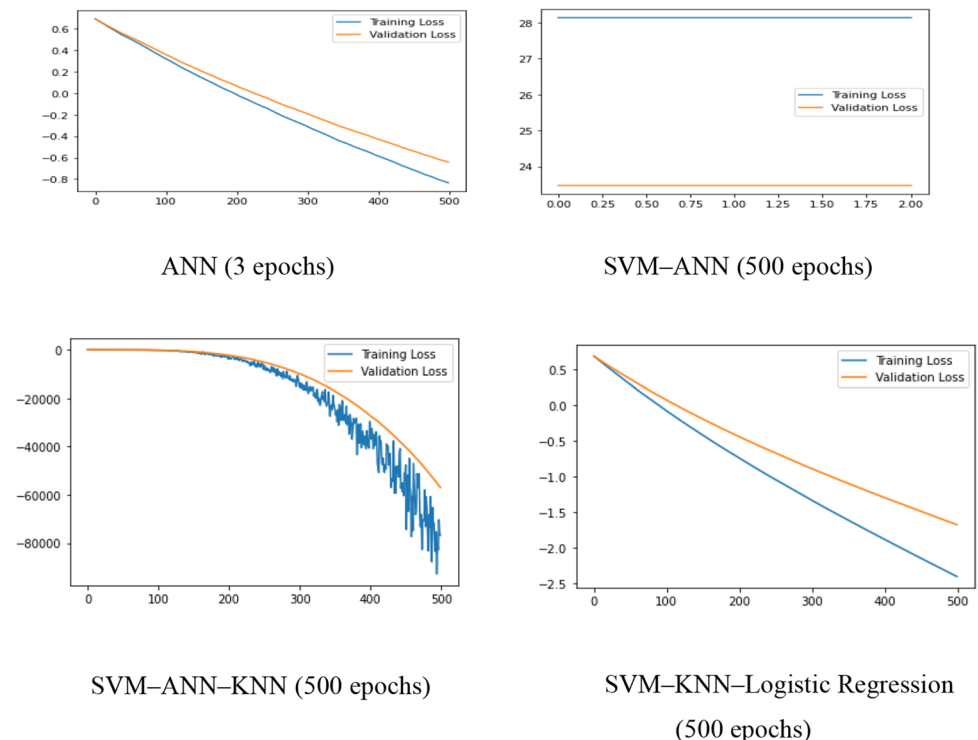
**Time estimation**

The classification of time for the estimation of the time of a model's earthquake is shown below:

- Class 0 – 1–7 days
- Class 1 – 8–14 days
- Class 2 – 15–21 days
- Class 3 – 22 or more

Here, the dataset is split into training and testing data used to design the models. In models that involve ANN,

**Fig. 13** ANN (3 epochs) **b** SVM-ANN (500 epochs) **c.** SVM-ANN-KNN (500 epochs) **d.** SVM-KNN-Logistic Regression (500 epochs)



**Table 8** Performance Evaluation of the Testing Data Designed on the Different Models Designed for Time Estimation

Model	Accuracy	Macro-Precision	Macro-F1 Score
ANN	0.30	0.08	0.12
SVM-KNN	0.87	0.85	0.82
SVM-ANN	0.22	0.05	0.09
SVM-KNN-Logistic Regression	0.91	0.89	0.90
SVM-ANN-KNN	0.93	0.92	0.91
SVM-ANN-Logistic Regression	0.87	0.86	0.86

the possibility of overfitting or underfitting is taken into account. So, the layers are regularized using a dropout layer. Additionally, for training the model, the number of epochs has been set to 500. One epoch represents how a dataset is passed forward and backwards through the neural network once, so 500 epochs mean 500 times the dataset is passed through the neural network. The major problem is the possibility of memorization where the neural network can memorize the result, which leads to overfitting.

So, during each iteration, the training and validation loss can be observed. If the training loss is lesser than the validation loss, then overfitting occurs; if the training loss is greater than the validation loss, then underfitting occurs. A ‘just right’ condition would be when the training and the validation loss. So, if the model's performance keeps getting worse after each iteration, then underfitting or overfitting could have taken place. In order to prevent this, early stopping has been set up, which stops the training if the model's performance starts to worsen. Additionally, callback helps in choosing the model which best suits the performance.

Their training versus validation loss has been plotted for the models with an ANN, as shown in Fig. 13. These plots show how the model stops training after a certain number of epochs based on the training performance. The default value of epochs has been set to 500. As seen in Fig. 13, except for ANN, the models have 500 epochs for the models' training. The error measures observed are used for the estimation of the performance of the models. The error measures used are macro-precision, accuracy, and macro F1-score. The reason macro is chosen is that the label sizes of the dataset are not perfectly balanced. So, the macro measures tend to bias the least popular ones. Thus, macro-precision is observed, and macro F1-score too. Accuracy is used to measure how many

readings were predicted accurately. The performance measures of the models are observed in Table 8.

From the model's performance evaluation, as seen in Table 8, it is visible that ANN and SVM-ANN show the worst performance. Both models have low accuracies, macro-precision, and macro F1-scores. While on the other hand, the SVM-ANN-KNN and SVM-KNN-Logistic Regression seem to show good performance. SVM-KNN and SVM-ANN-Logistic Regression seem to perform reasonably well too. The model parameters have been based on Galkina et al. (2019).

The validation testing data was used to check the performance of the model. Unlike the magnitude estimation, the estimation of time in the different models showed different performance results. Even if the dataset had an imbalance, it was not as prominent as the magnitude estimation.

From Table 9, as expected, ANN and SVM-ANN performed poorly. SVM-KNN showed a decent predictability performance. On the other hand, the validation data was correctly predicted by SVM-KNN-Logistic Regression and SVM-ANN-Logistic Regression, with SVM-ANN-KNN following. Thus, SVM-KNN-Logistic Regression, SVM-ANN-KNN, and SVM-ANN-Logistic Regression can be considered ideal models in [time estimation](#).

## Conclusion

This study aimed to design models to estimate the magnitude and to estimate the time. This prediction system uses Earth's Electric Field signals from Athens, Pyrgos, and Hios. The dataset of the electric field taken for processing was from the years 2004–2011. The electric field for each reading was

**Table 9** Performance Evaluation of the Validation Test Data on The Different Models Designed for Time Estimation

Model	Accuracy	Macro-Precision	Macro-F1 Score
ANN	0.17	0.04	0.07
SVM-KNN	0.72	0.79	0.69
SVM-ANN	0.33	0.08	0.12
SVM-KNN-Logistic Regression	1.0	1.0	1.0
SVM-ANN-KNN	0.89	0.92	0.89
SVM-ANN-Logistic Regression	1.0	1.0	1.0

calculated, and the GfDiff was calculated. Then the maximum peak value of each day is found. Then the day with the highest significant GfDiff that occurs 30 days or less from the date of the earthquake is placed is noted down, and the number of days between the date of the earthquake and the Date with significant GfDiff is calculated. The regression analysis is used as an initial step towards classifying earthquake magnitude. Further, as the dimensions are greater than 3, other classifications and learning models have been used in the paper. Once the dataset is made, the dataset is split into training and testing data with a test size of 33%.

Then the training data is fit into the models designed for magnitude estimation and time estimation. The models designed were ANN, SVM-ANN, SVM-KNN, SVM-KNN-Logistic Regression, SVM-ANN-KNN, and SVM-ANN-KNN. These models' performances were evaluated. It was observed that the data labels were not balanced for the magnitude estimation models, due to which the performance metrics were ill-scored. But reviewing the models for magnitude estimation, the model that did reasonably well was the SVM-ANN-Logistic Regression Model. SVM-KNN-Logistic Regression, SVM-ANN-KNN, and SVM-ANN-Logistic Regression models performed well with good prediction results for the time estimation model. The model tries to predict the earthquake's magnitude and predict when it would occur for future work. It will allow for safety precautions and will direct the community with warning levels.

**Funding** Not Applicable.

**Availability of Data and Material** Not applicable.

## Declarations

**Conflicts of Interest** Not applicable.

**Code Availability** Not Applicable.

## References

- Adeli H, Panakkat A (2009) A probabilistic neural network for earthquake magnitude prediction. *Neural Netw* 22(7):1018–1024
- Alarifi AS, Alarifi NS, Al-Humidan S (2012) Earthquakes magnitude predication using artificial neural network in northern Red Sea area. *Journal of King Saud University-Science* 24(4):301–313
- Asim KM, Martínez-Álvarez F, Basit A, Iqbal T (2017) Earthquake magnitude prediction in Hindukush region using machine learning techniques. *Nat Hazards* 85(1):471–486
- Asim KM, Idris A, Iqbal T, Martínez-Álvarez F (2018) Earthquake prediction model using support vector regressor and hybrid neural networks. *PLoS one*, 13(7)
- Astuti W, Akmeiliawati R, Sediono W, Salami MJE (2014) Hybrid technique using singular value decomposition (SVD) and support vector machine (SVM) approach for earthquake prediction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7(5):1719–1728
- Cicerone RD, Ebel JE, Britton J (2009) A systematic compilation of earthquake precursors. *Tectonophysics* 476(3–4):371–396
- Freund F, Ouillon G, Scoville J, Sornette D (2017) Earthquake precursors in the light of peroxy defects theory: critical review of systematic observations. *arXiv preprint arXiv:1711.01780*
- Galkina A, Grafeeva N (2019) Machine learning methods for earthquake prediction: A survey. In *Proceedings of the Fourth Conference on Software Engineering and Information Management (SEIM-2019)*, Saint Petersburg, Russia (Vol. 13, p. 25)
- Ghaedi K, Hejazi F, Ibrahim Z et al (2018) Flexible Foundation Effect on Seismic Analysis of Roller Compacted Concrete (RCC) Dams Using Finite Element Method. *KSCE J Civ Eng* 22:1275–1287. <https://doi.org/10.1007/s12205-017-1088-6>
- Ghaedi, K., & Ibrahim, Z. (2017). Earthquake prediction. *Earthquakes-Tectonics, Hazard and Risk Mitigation*, 205–227.
- Ghosh D, Deb A, Sengupta R (2009) Anomalous radon emission as the precursor of earthquake. *J Appl Geophys* 69(2):67–81
- Ghosh S, Chakraborty S, Sasmal S, Basak T, Chakrabarti SK, Samanta A (2019) Comparative study of the possible lower ionospheric anomalies in very low frequency (VLF) signal during Honshu, 2011 and Nepal, 2015 earthquakes. *Geomat Nat Haz Risk* 10(1):1596–1612
- Grisoni L (2017) Tally McCormick Miller, Friedemann Pulvermüller, Neural Correlates of Semantic Prediction and Resolution in Sentence Processing. *J Neurosci* 37(18):4848–4858. <https://doi.org/10.1523/JNEUROSCI.2800-16.2017>
- Hattori K (2004) ULF geomagnetic changes associated with large earthquakes. *Terrestrial Atmospheric and Oceanic Sciences* 15(3):329–360
- Jánský J, Pasko VP (2018) Earthquake lights: Mechanism of electrical coupling of Earth's crust to the lower atmosphere. *Journal of Geophysical Research: Atmospheres* 123(17):8901–8914
- Mallouhy R, Jaoude CA, Guyeux C, Makhoul A (2019) Major earthquake event prediction using various machine learning algorithms. *International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, Paris, France, pp. 1–7
- Moustra Maria (2011) Marios N Avraamides, Chris Christodoulou, Artificial neural networks for earthquake prediction using time series magnitude data or Seismic Electric Signals, *Expert Systems with Applications: An. International Journal* 28:12. <https://doi.org/10.1016/j.eswa.2011.05.043>
- Moustra M, Avraamides M, Christodoulou C (2011a) Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals. *Expert Syst Appl* 38(12):15032–15039
- Panakkat Ashif, Adeli Hojjat (2009) Recurrent Neural Network for Approximate Earthquake Time and Location Prediction Using Multiple Seismicity Indicators. *Computer-Aided Civil and Infrastructure Engineering* 24:4. <https://doi.org/10.1111/j.1467-8667.2009.00595.x>
- Saba S, Ahsan F, Mohsin S (2017) BAT-ANN based earthquake prediction for Pakistan region. *Soft Comput* 21(19):5805–5813
- Sgrigna V, Conti L (2012) A deterministic approach to earthquake prediction. *International Journal of Geophysics*, 2012
- Uyeda S, Nagao T, Kamogawa M (2011) Earthquake precursors and predictions. *Encyclopedia of Solid Earth Geophysics* 1:168–177
- Varotsos P, Alexopoulos K, Nomicos K et al (1986) Earthquake prediction, and electric signals. *Nature* 322:120. <https://doi.org/10.1038/322120a0>
- Vasti M, Dev A (2020) Classification and Analysis of Real-World Earthquake Data Using Various Machine Learning Algorithms. In *Advances in Data Sciences, Security and Applications* (pp. 1–14). Springer, Singapore

- Xu F, Song X, Wang X, Su J (2010) Neural network model for earthquake prediction using DMETER data and seismic belt information. In 2010 Second WRI Global Congress on Intelligent Systems (Vol. 3, pp. 180–183). IEEE
- Zarour M, Desharnais JM, Alarifi A, Habra N, Cassiers G, Robaey A (2012) Gained Experience by Making Intervention to Improve Software Process in Very Small Organizations. In: Mas A., Mesquida A., Rout T., O'Connor R.V., Dorling A. (eds) Software Process Improvement and Capability Determination. SPICE 2012. Communications in Computer and Information Science, vol 290. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-30439-2\\_5](https://doi.org/10.1007/978-3-642-30439-2_5)
- Zhou Z, Lin Y, Zhang Z, Wu Y, Johnson P (2019) Earthquake detection in 1D time-series data with feature selection and dictionary learning. *Seismol Res Lett* 90(2A):563–572
- Zhou WZ, Kan JS, Sun S (2017) Study on seismic magnitude prediction based on combination algorithm. In 2017 9th International Conference on Modelling, Identification, and Control (ICMIC), pp. 539–544, IEEE

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.