IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Application of Artificial Intelligence in Predicting Earthquakes: State-of-the-Art and Future Challenges

MD. HASAN AL BANNA[1], (Student Member, IEEE), KAZI ABU TAHER[2], M. SHAMIM KAISER[3], (Senior Member, IEEE), MUFTI MAHMUD[4], (Senior Member, IEEE), MD. SAZZADUR RAHMAN [5], (Member, IEEE), A. S. M. SANWAR HOSEN [6], (Member, IEEE), and GI HWAN CHO [7], (Member, IEEE)

[1]Bangladesh University of Professionals, Dhaka, Bangladesh (e-mail: 19541010@bup.edu.bd)
[2]Department of Information and Communication Technology, Bangladesh University of Professionals, Dhaka, Bangladesh (e-mail: kataher@bup.edu.bd)
[3]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh (e-mail: mskaiser@juniv.edu)
[4]Department of Computing and Technology, Nottingham Trent University, NG11 8NS – Nottingham, UK (e-mail: muftimahmud@gmail.com)
[5]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh (e-mail: sazzad@juniv.edu)
[6]Division of Computer Science and Engineering, Jeonbuk National University, Jeonju 54896, South Korea (e-mail: sanwar@jbnu.ac.kr)
[7]Division of Computer Science and Engineering, Jeonbuk National University, Jeonju 54896, South Korea (e-mail:ghcho@jbnu.ac.kr)

Corresponding author: Mufti Mahmud and Gi Hwan Cho.

ABSTRACT Predicting the time, location and magnitude of an earthquake is a challenging job as an earthquake does not show specific patterns resulting in inaccurate predictions. Techniques based on Artificial Intelligence (AI) are well known for their capability to find hidden patterns in data. In the case of earthquake prediction, these models also produce a promising outcome. This work systematically explores the contributions made to date in earthquake prediction using AI-based techniques. A total of 84 scientific research papers, which reported the use of AI-based techniques in earthquake prediction, have been selected from different academic databases. These studies include a range of AI techniques including rule-based methods, shallow machine learning and deep learning algorithms. Covering all existing AI-based techniques in earthquake prediction, this paper provides an account of the available methodologies and a comparative analysis of their performances. The performance comparison has been reported from the perspective of used datasets and evaluation metrics. Furthermore, using comparative analysis of performances the paper aims to facilitate the selection of appropriate techniques for earthquake prediction. Towards the end, it outlines some open challenges and potential research directions in the field.

INDEX TERMS AI, Deep Learning, Earthquake, Machine Learning, Review.

Nomenclature

Acronyms

P_0	negative predictive value
P_1	positive predictive value
Sn	sensitivity
Sp	specificity
ACC	ant-colony clustering
AdaBoost/LPBoost	adaptive/linear programming boost
AE	absolute error
AHC	agglomerative hierarchical clustering
AI	artificial intelligence
ANFIS/FIS	adaptive-network-based/fuzzy inference system
ANN	artificial neural network
AUC	area under the curve
BP	backpropagation
DL	deep learning
DNN	deep neural network
DT	decision tree
ELM	extreme learning machine
FAR	false alarm ratio
FLANN	functional link artificial neural network
FNN	fuzzy neural network
FUM	fuzzy user model
GA	genetic algorithm
GBV/PBV	global/personal best value
GFCV	generalized fuzzy clustering variety
GLM	generalized linear model
GMDH	group method of data handling

GP grid partitioning
HWT Haar wavelet transformation
IABC improved artificial bee colony
IASPEI international association of seismology and physics of the earth's interior
KNN K-nearest neighbors
LM Levenberg-Marquardt
LR logistic regression
LSTM long short-term memory
MAE mean absolute error
MFO moth flame optimization
ML machine learning
MLP multi-layer perceptron
MSE mean squared error
NARX nonlinear auto-regressive networks with exogenous input
NB Naive Bayes
NDAP neural dynamic optimization of Adeli and Park
NDC neural dynamic classification
NFS neuro-fuzzy system
PCA principal component analysis
PDF probability density function
PHMM poisson hidden Markov model
PNN probabilistic neural network
PR polynomial regression
PRNN pattern recognition neural network
PSO particle swarm optimization
RBFNN radial basis function neural network
RE relative error
RF random forest
RMSE root mean square error
ROC receiver operating characteristics
SC subtractive clustering
SES seismic electric signal
SVD singular value decomposition
SVM/R support vector machine/regressor
TEC total electron content
WIA Willmott's index of agreement

I. INTRODUCTION

EARTHQUAKE is a natural disaster caused by the movement of tectonic plates of earth due to the release of its substantial internal energy. A major earthquake with a magnitude greater than five can inflict massive death tolls and huge infrastructural damages costing billions of dollars. However, if the occurrences of an earthquake can be predicted, the magnitude of destruction can be minimized. A complete earthquake prediction procedure should have three types of information: the magnitude, location, and time of occurrence. Since 2005, there have been 28,400 occurrences of earthquakes with a magnitude of more than five around the world [1]. Fig. 1 presents the location of the occurrences from January to December 2019 [1]. Observing closely, it is possible to see some patterns in locations of earthquakes (denoted by red dots in Fig. 1). This kind of patterns may provide researchers with possibilities to accurately predict earthquakes.

Earthquake prediction can be classified into the short-term and long-term process. Short-term prediction is very complicated as it predicts earthquakes within days or weeks of their occurrences. Therefore, it should be precise and accurate, and fewer false alarms are appreciated. Generally, short-term predictions are used for evacuation of an area before an earthquake. On the other hand, long-term earthquakes are predicted based on earthquakes periodical arrival, which carries a few pieces of information. Still, they can help to set standards for building code and designing disaster response plans. In 2009, L'Aquila city of Italy was struck by a 5.9 magnitude earthquake, taking away the life of 308 citizens. However, the earthquake forecast commission of Italy predicted that there would be no damage, and they did not evacuate the city. Such faulty prediction can lead to a massive massacre taking away lives and damaging lots of infrastructures. The scientists involved in that incident were punished with six years of imprisonment [2].

The earthquake prediction models perform well with earthquakes having medium magnitudes, but while the shocks have high magnitude, the outcomes achieved are poor. Major earthquakes cause most damages and bring the most concern. The reason behind this scenario is that there is a smaller number of earthquakes with high magnitude, and without data, the prediction becomes very difficult. The researches on the prediction use historical data involving an earthquake's energy, depth, location, and magnitude from the earthquake catalogs. Based on the magnitude of completeness value, the area-specific earthquake parameters like b-value parameters are calculated. Machine learning (ML) based algorithms mainly calculate the seismic indicators like Gutenberg Richter b-values, time lag, earthquakes energy, mean magnitude, etc. [3]. Instead deep learning (DL) based models can calculate thousands of sophisticated features by themselves [4], [5]. Since ML and DL based models are data-driven and major earthquakes happen in a few cases, it is challenging to predict them based on historical data. Some methods predict the major earthquakes by separately training them or adding weights to them, but these models need many improvements [6].

Another way for successful prediction is to find some precursors of a major earthquake. Precursors are the changes in elements in nature before the occurrence of an earthquake. Earthquake scientists suggest that concentration of Radon gas, strange cloud formation, earth's electromagnetic field variations, humidity, temperature of the soil, crustal change, etc. can be the possible candidate precursors [7]. Such generalization may be misleading because there were many cases found where these precursors were present without the occurrence of an earthquake, and earthquakes took place even though there was an absence of these precursors. According to the International Association of Seismology and Physics of the Earth's Interior (IASPEI), precursor-based earthquake research should have some qualities like- it should be observed from more than one site and instruments and should

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

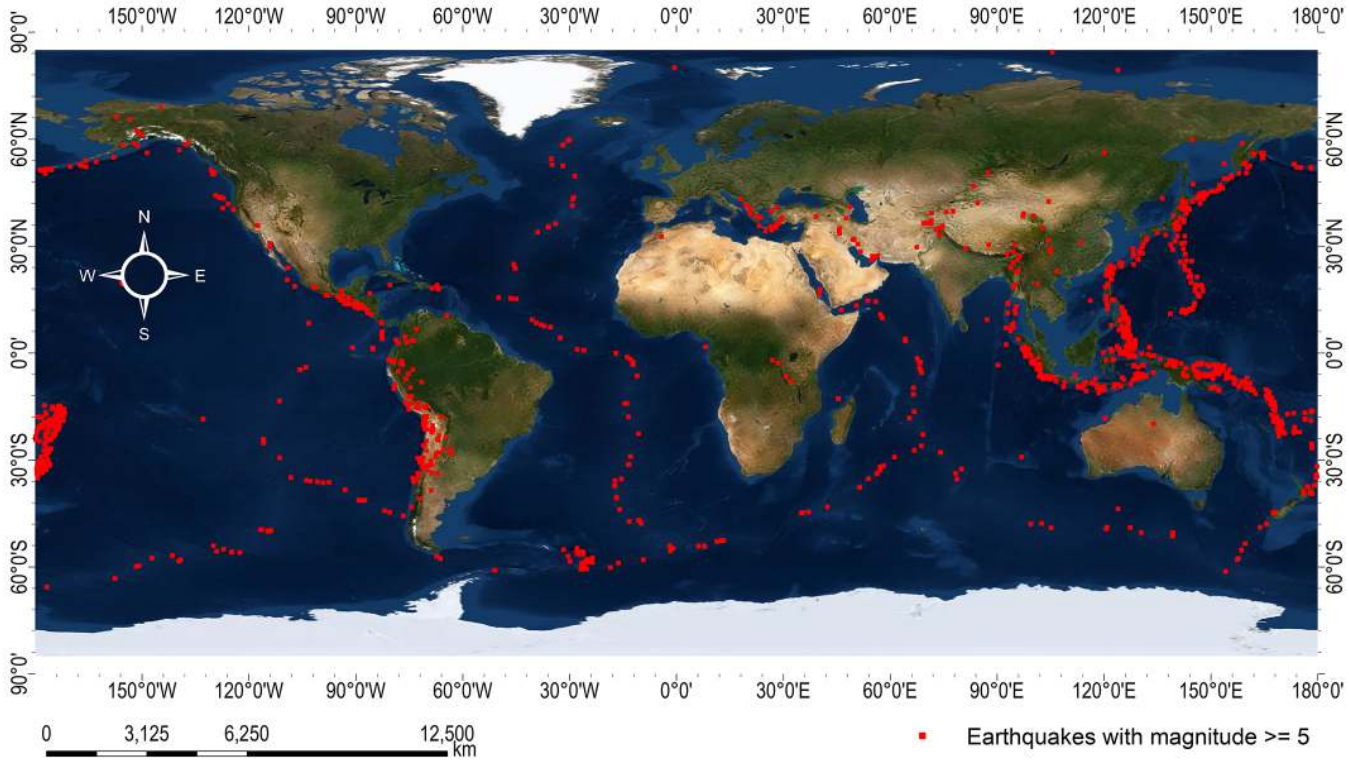

FIGURE 1. Earthquakes occurred around the world from January 2019 to December 2019 with magnitude greater or equal to five. In twelve months, 1637 earthquakes happened around the world. The data were collected from the United States Geological Surveys and plotted using ArcGIS software. The red square represents the epicenter of occurrence of the earthquake.

be related to stress and strains of the earth [8]. No precursor with definite proof of predicting earthquake is found yet. Fig. 2 depicts the necessary process of prediction of earthquakes with AI based methods. Some AI classifiers are used for this prediction process, along with the input parameters and preprocessing.

Evaluation of an earthquake prediction method can be carried out using different metrics such as positive and negative predictive values (P_1, P_0), specificity (Sp), sensitivity (Sn), accuracy, false alarm rate (FAR), R-score, root mean square error (RMSE), mean squared error (MSE), relative error (RE), mean absolute error (MAE), area under the curve (AUC), chi-square testing, and so on. Earthquake models are dependent on the area from where the data are collected. That is why there is a need for a standard dataset of an earthquake on which the researchers can calculate the evaluation metrics for comparing their models with previous studies.

There are some review articles available that evaluated earthquake prediction studies. In some of the reviews, the precursory based researches are criticized based on their scientific values [10]. How these precursors can be used in earthquake prediction is also elaborated [11]. The use of Radon concentration for the prediction of an earthquake is also investigated [17]. Data mining techniques are discussed in the study [15]. Classical ML techniques are reviewed, and their evaluation techniques are discussed in the study [20]. How the rule-based techniques can work in this field are investigated in [21]. Mignan and Broccardo [22] discussed the DL techniques in this field. There is a missing study where all these techniques are accumulated together, which can be an excellent resource for AI researchers in the field of earthquake prediction.

For this review, earthquake prediction studies that include AI-based methods are searched in databases like IEEE Xplore digital library, Science Direct, and Google Scholar. Initially, 292 papers were found. After removing duplicates and reviewing the abstract of these papers, 148 papers were selected for full-text review. This study includes both journal and conference articles because the conference proceedings also present substantial content vital in the prediction process. After full-text reviewed, 64 papers were excluded as they were not specialized researches of earthquake prediction. Finally, the 84 papers are studied in this research. Fig. 3 illustrates the selection procedure of the articles for this study using a Prisma diagram. Fig. 4(a) depicts the buzzwords of earthquake researches. Fig. 4(b) represents the pie diagram showing the distribution of AI algorithms, and it clearly shows that the Artificial Neural Network (ANN) is used in most studies. Fig. 4(c) shows the yearly distribution of the reviewed articles. Studies of the last 15 years were

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

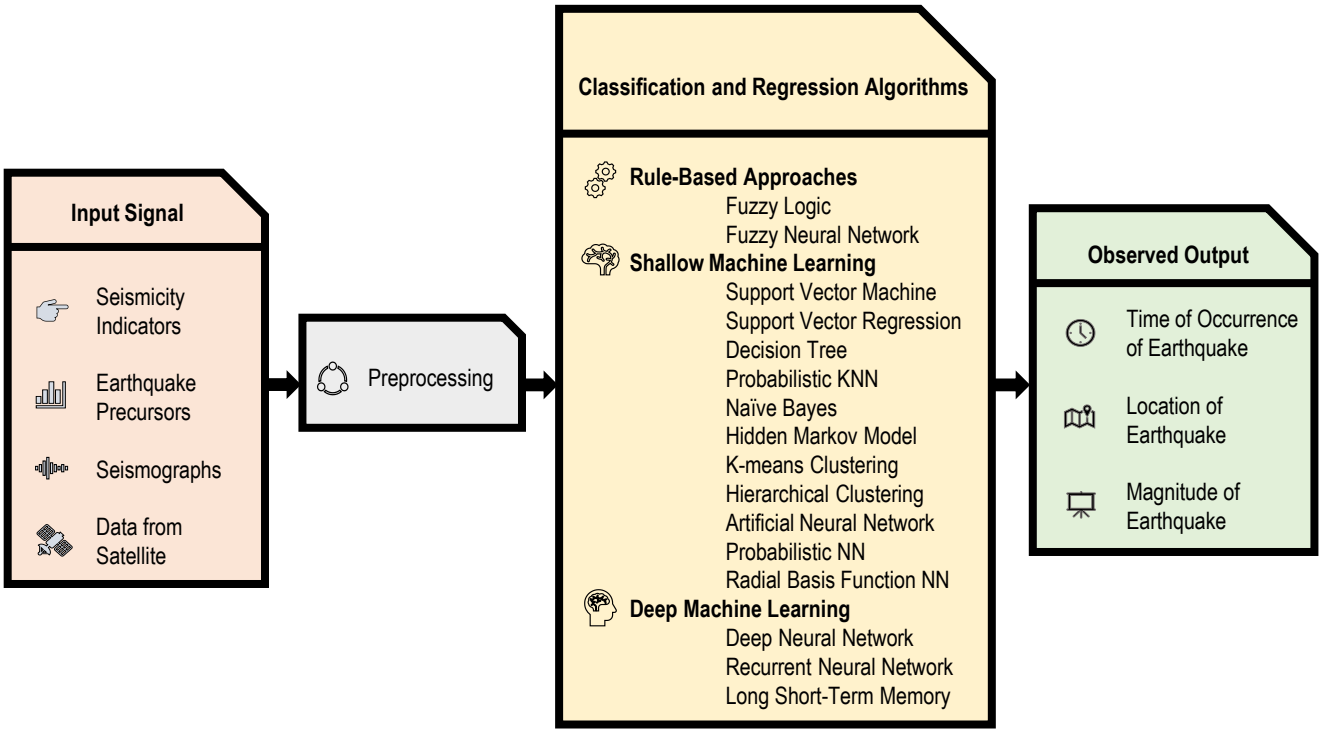

FIGURE 2. A general earthquake prediction model. Earthquakes are predicted based on some features. These features can be the seismicity indicators, which are calculated from the earthquake catalog. Some earthquake precursors can be found which happened a few days before the earthquake. But these precursors never confirm an earthquake. Radon gas concentration, soil temperature variation, and strange cloud formation are some of the earthquake precursors. From the seismograph P-wave and S-wave can be detected by which earthquake can be predicted. Some countries use dedicated satellite to monitor earthquake-related parameters which helps in finding earthquake precursors. These data are used as an input signal to the prediction model. Then the data are processed to remove missing values and converted to a form that suits the classification and regression algorithms. In this study, we have considered the AI-based algorithms only. These algorithms try to find hidden patterns in the data to classify them. In the end, these algorithms predict the time, location, and magnitude of an earthquake.

incorporated into this research. Most of the researches were from the year 2016 to the year 2019. In these four years, 36 pieces of research were done based on AI techniques. The other 48 studies were selected from the year 2005 to the year 2015. In the year 2009 and the year 2014, 8 studies were selected each year, which is the highest in the first 11 years considered for review.

This study focuses on reviewing the earthquake researches that are based on different AI techniques. It reflects the state-of-the-art historically. All the possible AI-based methods used in this regard are included with their proposed methods and findings. To the author's knowledge, the other review works in this field considered a few aspects of the earthquake and did not cover all the AI methods. We have incorporated all the studies that focus on earthquake prediction and its characteristics with their performance. This will widen the scope of further research by pointing out the most effective parameters of earthquakes and techniques with higher accuracy. Table 1 presents existing review articles published in this field. The main contributions of this paper are discussed below, which shows the uniqueness of this study:

i. This study considers the articles that include rule-based methods such as Fuzzy logic, adaptive-network-based fuzzy inference system (ANFIS); Shallow machine learning algorithms such as support vector machine (SVM), support vector regression (SVR), random Forest (RF), decision tree (DT), radial basis function neural network (RBFNN), K-nearest neighbor (KNN), probabilistic neural network (PNN), ANN, clustering; and Deep machine learning methods such as a recurrent neural network (RNN), long short-term memory (LSTM), deep neural nets (DNN) for predicting an earthquake. Not only that, the bio-inspired models are also evaluated. To the best of our knowledge, no study has been done considering all these techniques.
ii. 84 papers from renowned publishers are extensively reviewed based on the AI techniques.
iii. This study presents an in-depth description of the methodologies of these researches.
iv. Relative comparison between different techniques based on their performances are presented.
v. The databases used in the studies are also included, which can help earthquake research enthusiasts in their studies. Performance comparison based on these datasets are also provided in this article.

It is expected that this study will attract new AI researchers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

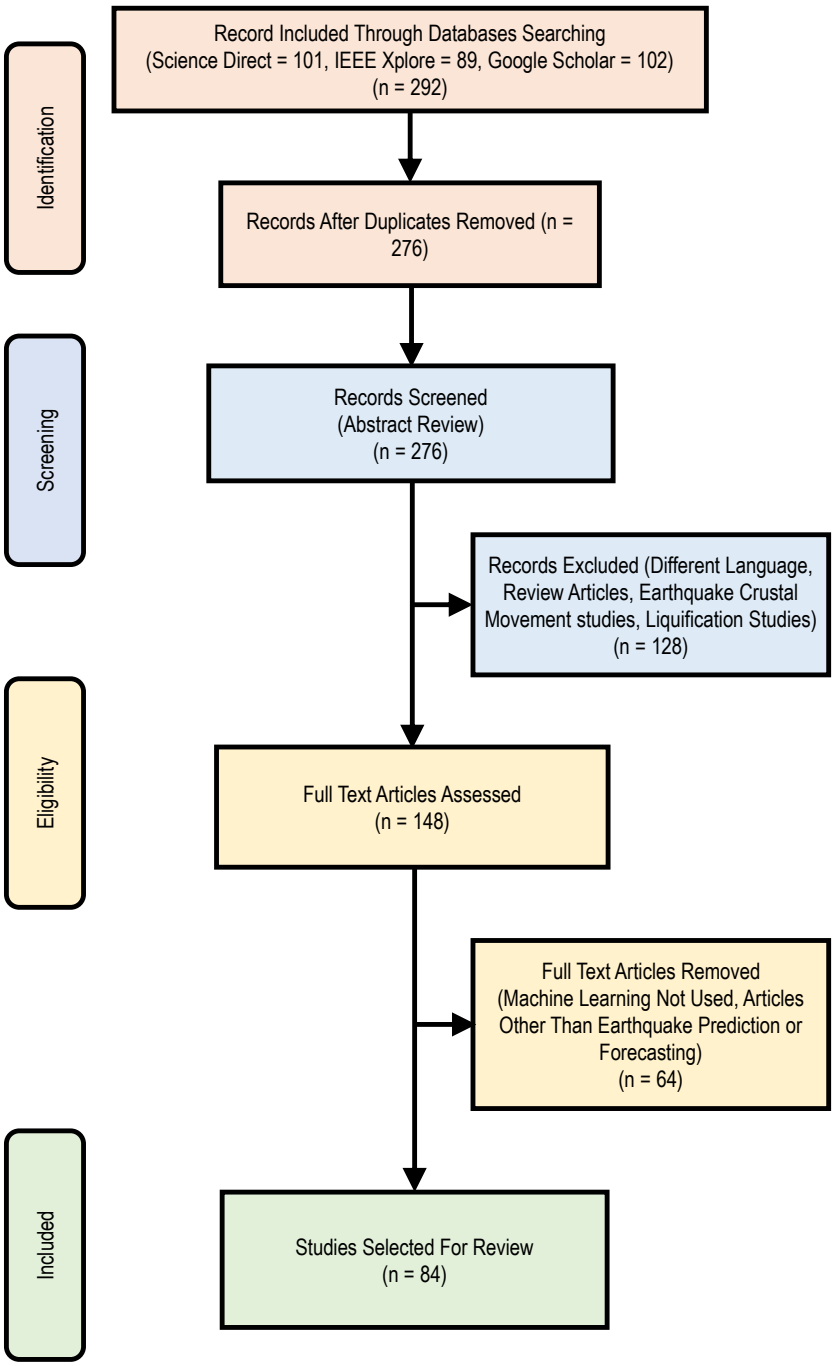

FIGURE 3. Prisma diagram of the selection process of the research articles of this review. The search string used in this study is- ("Neural Network" or "Machine Learning" or "SVM" or "RNN" or "HMM" or "Hidden Markov" or "Fuzzy" or "Deep Learning" or "Data Mining" or "SVR" or "PNN" or "LSTM" or "Clustering" or "Radial Basis" or "RBF" or "Support Vector") and ("Earthquake") and ("Prediction"). Based on this search string, we have initially found 292 research articles from Science Direct, Google Scholar, and IEEE Xplore digital library. After screening and eligibility testing, we have selected 84 research papers for this review.

to this highly demanding field of earthquake prediction.

The rest of the paper is organized as follows. In section

TABLE 1. Review researches based on earthquake prediction

Year	Reference	Topic of Review
1997	[9]	reviewed earthquake researches till 1997
2007	[10]	criticized different seismo-electromagnetic precursor-based researches for scientific value
2009	[11]	reviewed short-term earthquake prediction researches based on seismo-electromagnetic signals
2009	[12]	reviewed earthquake precursor -based researches and the satellites monitoring these precursors
2009	[13]	reviewed researches focusing animal behavior before earthquake
2011	[14]	reviewed earthquake prediction researches based on space & ground-based sensors
2012	[15]	reviewed 19 journals focusing earthquake prediction using NN & data mining approaches
2014	[16]	presented a brief review of earthquake prediction studies based on NN, Fuzzy logic, & bio-inspired algorithms
2015	[17]	reviewed radon gas concentration-based researches
2017	[18]	reviewed earthquake precursor-based researches in China
2018	[19]	reviewed data mining techniques-based studies for natural disasters
2019	[20]	reviewed ML trends in earthquake
2019	[21]	reviewed DL-based researches & future trends
2019	[22]	reviewed NN & DNN-based earthquake prediction studies

II, the works related to this study are discussed. Section III discusses the working principle of the most common AI algorithms. Section IV briefly discusses the methodologies used by the researchers of earthquake prediction. Section V describes popular evaluation metrics for performance categorization while section VI examines and discusses their performances. In section VII, some challenges of the earthquake prediction studies are mentioned, and section VIII provides the concluding remarks.

II. RELATED WORKS

Researches on earthquake prediction started in the late nineteenth century. Geller [9] reviewed earthquake researches of one hundred years and criticized their quality. He divided the researches based on different time ranges like researches before 1960, after 1960, and 1962 to 1997. He raised questions about the precursors of earthquakes and acknowledged the IASPEI guidelines for precursory researches. He recognized the works of the VAN group [23] with the earth's electric signal but doubted their research procedure. Different AI models evolved after this review. Sevgi [10] criticized different seismo-electromagnetic precursory based researches for earthquake predictions. He evaluated the researches based on their scientific content, considering whether the researches were conducted scientifically or not. He found that most of the precursory predictions were not made based on IASPEI's guidelines. He also mentioned that the earth's electromagnetic signal is very noisy and has characteristics from local permittivity and permeability, introducing background noise. In his review, though, he discussed the earth's electric signal, he did not review the earthquake prediction models with historical data.

Uyeda *et al.* [11] reviewed the short-term prediction of earthquakes based on seismo-electromagnetic signals. They first reviewed the researches that covered the history of short-term earthquake predictions. They suggested that in precursory researches, nonseismic precursors should also be considered. They also discussed different types of emissions of the earth before earthquakes like telluric current and high-frequency electromagnetic waves. They pointed out that this electric signal should not be considered as earthquake precursors. Alvan and Azad [14] reviewed earthquake prediction researches based on space-based and ground-based sensors covering most of the earthquake precursors. They divided the studies based on the different precursors like earth's crust, temperature, strange cloud formation, humidity, and Radon gas concentration. The satellite imagery and ground parameters were also discussed in this research. Woith [17]reviewed earthquake prediction techniques that used Radon gas concentration as a parameter. He pointed to the fact that though there are anomalies present in Radon concentration, in many cases, no earthquake occurs. He reviewed 105 publications and enlisted their databases and methods. He also discussed how models should differentiate between seismic disturbance of Radon concentration and human-made ones.

Huang *et al.* [18] reviewed earthquake precursory researches from 1965 to 2015 in China. In this research, the studies were clustered in different time ranges. Seismic parameters, geo-electromagnetic parameters, geodetic and gravity parameters, and ground fluids were considered as earthquake precursors. Then they discussed the ongoing projects in China for earthquake prediction. Mubarak *et al.* [12] discussed earthquake precursors like gravity variations, temperature and humidity fluctuation, Radon concentration changes, and electric field changes. Then they briefly discussed seven countries which use satellite for their precursory predictions. From the literature they reviewed, a decrease in air humidity, and an increase in Radon concentration and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

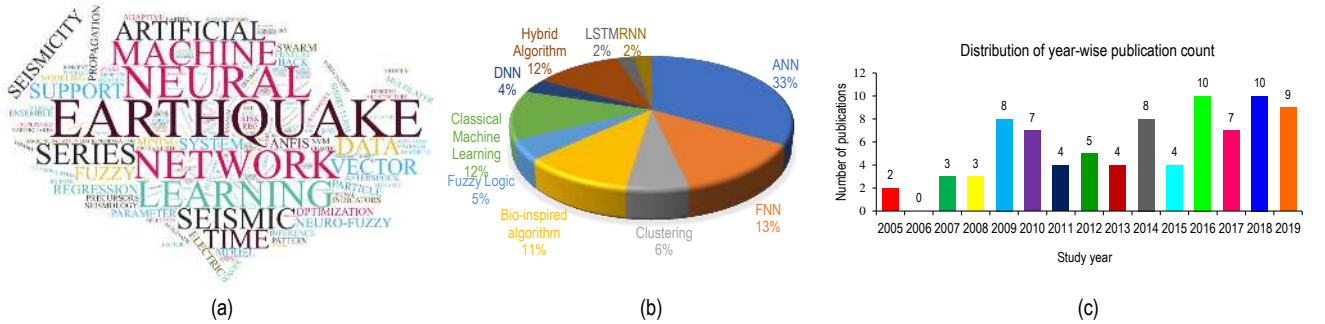

FIGURE 4. (a) The focuses of the reviewed studies depicted as retrieved keywords from the article title. This was generated using the word cloud to show what the reviewed research articles focused on their titles. (b) Algorithm wise distribution of the articles. Here using a pie diagram, the most popular algorithms in the reviewed studies are presented. The ANN was used in 33% of the cases. (c) Year wise distribution of the studies. Here, we present the number of research works that happened from the year 2005 to 2019 on earthquake prediction using AI methods, which were selected in this review. In recent years the number of researches increased a lot.

electric field can be taken as earthquake precursors. Bhargava *et al.* [13] reviewed the articles which used animals' weird behavior before an earthquake as the indicator of an earthquake and mentioned that China, Japan, and the USA have facilities for this kind of research. They did not include historical data-based researches for earthquake prediction.

Otari and Kulkarni [15] reviewed 16 journals from 1989 to 2011 and grouped them based on NN and data mining approaches. In 2018, Goswami *et al.* [19] reviewed data mining techniques to predict, detect, and develop management strategies for natural disasters like earthquakes, Tsunami, or cyclones. They proposed a twitter-based disaster management model for India. Galkina and Grafeeva [20] analyzed the ML trend in earthquake prediction research. They observed datasets, features, the magnitude of completeness, and performance measurement criteria for these studies. They noticed that these studies face difficulties in predicting rare but more important major earthquakes. Azam *et al.* [16] reviewed earthquake prediction works based on NN, Fuzzy logic, and bio-inspired optimization algorithms. However, there is a lack of detailed research in this area. Jiao and Alavi [21] reviewed the DL-based researches and predicted future trends in this area. DNN is used for this purpose as it can take unorganized data and calculate many features by itself. They presented a generalized picture of the working procedure of these systems. Mignan and Broccardo [22] analyzed 77 articles on NN from 1994 to 2019. They divided the studies into two categories- ANN and DNN. DNN is the future of the earthquake prediction model though the model is more complex and uninterpretable. As a result, overfitting becomes a problem.

All the review articles discussed either short-term earthquakes with earthquake precursors or addressed some portion of AI methods to the best of the author's knowledge. No review covers short-term earthquakes, long-term earthquakes, earth's electromagnetics, ANN-based methods, Fuzzy based studies, clustering techniques, DNN, bio-inspired algorithms, and ML techniques for prediction of earthquakes. Through this study, all these sectors were incorporated for a comprehensive review of earthquake prediction.

III. ARTIFICIAL INTELLIGENCE (AI) ALGORITHMS

A. RULE BASED APPROACHES

1) Fuzzy Logic

The decision-making process of humans is different than how a machine works. Between "yes" and "no", human considers some other options. Fuzzy-logic systems represent this way of decision making. A fuzzy logic system has some modules with whom it takes a decision. The fuzzification module uses a membership function to generate a membership degree from crisp inputs. Membership degree can be- large positive, medium positive, small, mid negative, and large negative. Then the knowledge base comes, where there are some IF-THEN rules, which are adopted from human behavior. The inference engine compares the input with the rules and provides reasoning for the input. The defuzzification module converts this reasoning to crisp output. Fuzzy logic is popular because of its ease of use, and flexibility. Fig. 5(a) shows the basic structure of Fuzzy logic systems.

2) Fuzzy Neural Network (FNN)

When Fuzzy networks are represented as ANN so that they can be optimized using backpropagation or genetic algorithm (GA), we call the system a neuro-fuzzy system (NFS). One approach to implementing this system is the Mamdani approach by Ebhasim Mamdani [24]. For this approach, both the input and output of the system must be a fuzzy quantity. It uses a simple min-max operations structure, which makes it a great model for human inference systems. This model is understandable for humans, but the complexity increases with the increase in input rules. This model uses five layers for prediction, which are enlisted as:

i. Fuzzification layer: The input vector consisting of features enters into the fuzzification layer, where its membership value is calculated. Generally, the Gaussian

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

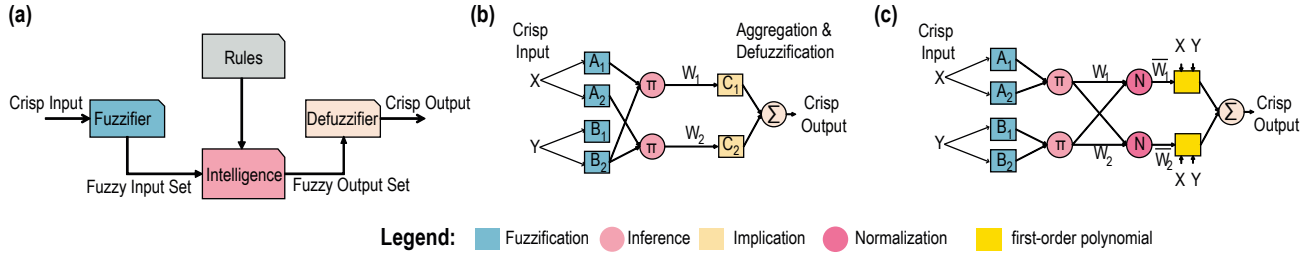

FIGURE 5. (a) Fuzzy Logic architecture. In Fuzzy logic, the crisp input is fuzzified and compared with the rules to create a crisp output. (b) Mamdani FNN architecture. It has five layers that work together to predict a value. A_1, A_2, B_1, B_2 are the input nodes which take X, Y as input. The next layer denoted by Π multiplies the values of the previous layer to generate weight W_1 and W_2. These weights are used for implication, and the result of them is summed together. This output goes to the defuzzification layer to produce an output. (c) Takagi-Sugeno ANFIS architecture. This is a five-layer architecture where A_1, A_2, B_1, B_2 are the input nodes which take X, Y as input. The next layer denoted by Π multiplies the values of the previous layer to generate weight W_1 and W_2. The layer denoted by N normalizes the value of the previous layer and outputs $\bar{W}_1$ and $\bar{W}_2$. The rules are a combination of X, Y and the input nodes. These rules are multiplied and summed together to produce an output. The square layers are adaptive as they can be changed to produce a better output.

function is selected for calculating the membership value [24].

ii. Fuzzy inference layer: In the inference layer, fuzzy rules fire based on the input vector by multiplying the membership values.

iii. Implication layer: In the implication layer, consequent membership functions are calculated based on their strength.

iv. Aggregation layer: In the aggregation layer, the multiplication of firing strength and consequent parameters are summed together.

v. Defuzzification layer: The final crisp output is achieved by defuzzification, which follows the center of the area method.

Fig. 5(b) depicts the layer structure of the Mamdani FNN.

The other approach is Takagi Sugeno neuro-fuzzy system, which is also known as ANFIS. The NN and fuzzy inference system (FIS) are combined for this model [25]. Usually, FIS does not have learning ability, and its membership function is fixed. Five layered ANFIS approach solves these problems and generates IF-THEN rules from the knowledge of an expert avoiding extensive initialization stage and making the system efficient in computation.

The first layer generates grade membership functions like Gaussian functions, triangular functions, and trapezoid functions, which are used to generate firing strength. The second layer uses the membership function's grade to calculate the firing strength. The output of each model is compared, and the product or minimum of them is selected. In the third layer, normalization is done by dividing the firing strength of a rule by the combined firing strength. Defuzzification is the next layer where the output is calculated using the weighted parameters. The sum of all the defuzzified nodes is summed together in the last stage to generate the overall ANFIS output. Fig. 5(c) depicts the architecture of an ANFIS model. The square layers are adaptive, that means, with some optimization algorithms like BP or GA, we can adjust these layers.

B. SHALLOW MACHINE LEARNING

1) Support Vector Machine (SVM)

SVM is a ML-based classification algorithm used successfully in applications like classification, pattern recognition, and prediction. It organizes the classes by constructing a hyperplane in an N-dimensional plane in a way so that the hyperplane ensures maximum margin distance between data points of the classes [26]. The data points close to the hyperplane are called support vectors, which determine the orientation and position of the hyperplane. When a linear hyperplane cannot separate the classes, a higher dimensional nonlinear hyperplane is needed. Polynomial, sigmoid, and radial basis function (RBF) kernels are some accessible kernel functions that are used for these cases. SVM is a quite computationally expensive classifier and usually takes longer time for training. It possesses regularization capability and is capable of working with linear or nonlinear data. Fig. 6(a) shows how SVM constructs hyperplane between two groups of data for classification purposes.

2) Support Vector Regression (SVR)

SVR algorithm works in an entirely different manner than most of the regression algorithms [27]. Where the other regression algorithms try to minimize the sum of squared error, SVR is concerned with the error when the error is in a particular range. This regression method works similarly to SVM, but instead of providing a class as output, it produces a real number. SVR gives flexibility in case of error to minimize coefficients (E-value) and optimizes them to improve performance. It is trained with symmetrical loss function to penalize low, and high miss estimates equally. The computation complexity of SVR is not dependent on the input shape's dimension. For nonlinear operations, it uses kernel functions like polynomial kernel, which is represented by Eq. (1), where x_i, x_j are two different observations in the dataset, r is the coefficient of the polynomial, and d is the degree of the polynomial and Gaussian RBF kernel, which is represented by Eq. (2), where x_i, x_j are two different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

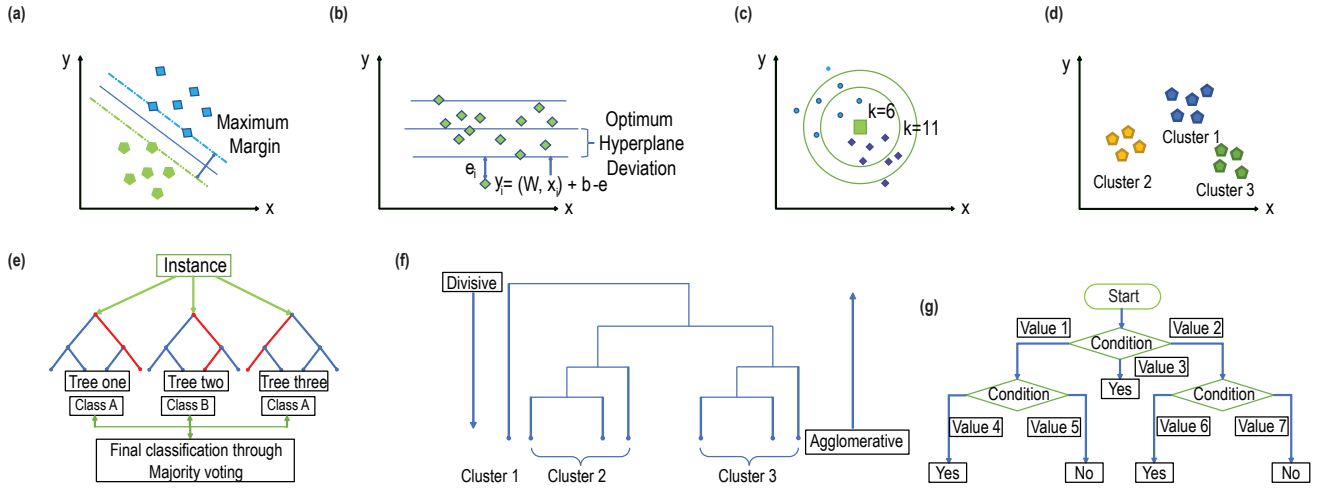

FIGURE 6. (a) Classification process of the SVM algorithm. This algorithm tries to create a hyperplane to maximize the margin between to close data points of two different classes. (b) Working procedure of the SVR algorithm. It inherits properties from the SVM algorithm but does regression operation. A regression line is drawn to cover the whole dataset. The maximum deviation is denoted by e. The data which are not in the deviation of $\pm e$ are the outliers. (c) Decision-making process of the KNN algorithm. Based on the value of k, classification can be changed. (d) The working principle of the K-means clustering algorithm. Based on Euclidean distance, the clusters are formed. The output of the clustering is represented using different colors. (e) Classification process of the RF algorithm. From the data, different sub-trees are generated, which produces different classes. The class with most occurrences are selected as the output class. (f) Dendrogram of hierarchical clustering. When data in the Dendrogram are accessed in a top-down approach, it is called divisive clustering, and when it is accessed in a bottom-up fashion, it is called agglomerative clustering. (g) Decision-making process of the DT (C4.5) algorithm. Here, based on different conditions, the algorithm reaches to different decisions.

observations in the dataset, and γ is the spread of the kernel.

$$f(x_i, x_j) = (x_i \times x_j + r)^d \tag{1}$$

$$f(x_i, x_j) = e^{-\gamma||x_i - x_j||^2} \tag{2}$$

It possesses excellent generalization capability and capable of achieving high prediction accuracy. This process is depicted in Fig. 6(b).

3) K-Nearest Neighbor (KNN) Algorithm

It is a supervised ML algorithm where data in close proximity are thought to have the same output class [28]. The value of k is determined at first, which should not be very small or massive. Then the input data's Euclidean distance is calculated considering each feature. The Euclidean distance of two-point a and b is represented by Eq. (3).

$$||a - b|| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

Where coordinates of a is (x_1, y_1), and coordinates of point b is (x_2, y_2). Based on those distances, the data are sorted in smallest to largest order. Then the labels of the first k entities are considered, and the label with the highest occurrences is selected as the class of that data. Although it is a straightforward algorithm, it is not suitable for large datasets. Fig. 6(c) shows how the change in the value of k can change the prediction process.

4) Random Forest (RF) Algorithm

This classifier is a collection of randomly selected decision trees that works in a voting method [29]. It takes votes from different decision trees to determine the final class. This method combines the output of different random decision trees to provide a classification result. Each tree of RF is constructed using different bootstrap samples. It changes the procedure of construction in the case of a regression tree. RF is quite similar to bagging, but it contains another extra layer to introduce randomness. It has the ability to acquire high accuracy and can handle massive datasets efficiently. The decision-making process of an RF classifier is shown in Fig. 6(e). This method does not need hyperparameter optimization, therefore, it is a simple but effective ML method.

5) Decision Tree C4.5 Algorithm

This is a statistical classifier that works by generating decision trees based on information gain, where the highest normalized gain is selected as the criterion of splitting [30]. It is an improved version of the Iterative Dichotomiser 3 algorithm, which can deal with continuous variables, discrete variables and missing values. It builds a classifier by analyzing the training set to classify the test set. C4.5 builds decision trees that respect Occam's Razor. In C4.5, missing values are dealt with by estimations from the dataset. This algorithm supports tree pruning to deal with overfitting. Here a subtree can be replaced by a leaf node. Meanwhile, small changes in a dataset can lead to a change in the decision tree. This algorithm is easy to implement and works well, even with a noisy dataset. Fig. 6(g) shows how DT algorithms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

come to a conclusion based on conditions in the data.

6) K-means Clustering

Clustering is an unsupervised learning technique that segments data into different sub-divisions. K-means clustering is a prevalent iterative clustering technique that finds local maxima in each iteration [31]. For this algorithm, initially, the value of *k* is fixed. The optimum value can be found by using the elbow method. The algorithm first assigns random means on each cluster and classifies the data based on distance from the mean. Usually, Euclidean or Manhattan distances are used for calculating distance. Based on the assigned clusters, the mean is again calculated for each cluster, and then the data are reclassified. This process continues until there is no change in means between successive iterations. The mean is calculated again to obtain the classified data. Fig. 6(d) shows the workflow of this algorithm.

7) Hierarchical Clustering

Hierarchical clustering is a hierarchical decomposition of data. For this, a dendrogram is built. Fig. 6(f) shows an example of a dendrogram. Initially, this algorithm considers each data points as an individual cluster [31]. Then two clusters with the lowest Euclidean distance are combined into one cluster. This distance is assigned as the height of the dendrogram. Afterwards this cluster is compared with other clusters to find two clusters with the lowest distance. This process continues until *k* number of clusters are obtained, or only one cluster is left. The value of *k* can be found from the dendrogram by a horizontal line that is not crossed by the verticle lines. Although it is a high-speed algorithm, it cannot be used for a large dataset.

8) Artificial Neural Network (ANN)

The ANN is a synthetic mimic of the human brain in response to some event. It is composed of some neurons, which are linked together with some weights and biases. By training an ANN, the biases and weights are tuned in such a way that it produces output closer to the actual result [32]. Typically, an ANN has an input layer that takes data, one or more hidden layers for feature generation, and an output layer for classification. Fig. 7(a) shows a common ANN architecture. Each layer consists of neurons that have activation functions inside. These activation functions take the inputs and biases of the previous layers, calculate the weighted sum, and scale it within some range. This way, the data move forward, and the output layer produces predicted output. After this forward pass, the error is calculated by the sum square function which is the residue of actual output and predicted output. This error function needs to be minimized. The BP algorithm is adopted to minimize the error and adjust the biases and weights accordingly. The first derivative of the error is calculated with respect to the weights. Then learning rate is multiplied with these values and is deducted from the weights to adjust them.

This method can be slow to converge to the optimal value. To resolve this, the Levenberg-Marquardt method can be used for faster convergence [33]. This method can approach second-order training speed as the Hessian matrix is not computed [33]. Since the error function is in sum squared form, the Hessian function can be estimated as $H = J^T J$, and its gradient is $g = J^T e$, where J is the Jacobian matrix, e is a vector of the network error, and T means transpose. The Jacobean matrix is calculated using BP. The LM approximates the Hessian matrix as Eq. (4).

$$X_{K+1} = X_K - [J^T J + \mu I]^{-1} J^T e \tag{4}$$

Where I is an identity matrix and μ is a control parameter. When μ is equal to zero, it works like Newton's method, and when it is big, it works like a gradient descent with a small step size. Newton's method is faster for reaching the minimization of the error. When a successful step is taken, the value of μ is decreased, and vice versa.

GA is a nonlinear optimization technique with impressive global searching ability [34]. This algorithm is often used to set the initial weights and biases of the NN so that it converges to its optimal form fast and gets out of the local minima. In GA, a set of individuals are taken, which are called population. Each individual in the population is called chromosomes, and chromones are composed of genes. There are five steps in this algorithm as follows:

i. initialization
ii. computation of fitness function
iii. selection of parent chromosomes
iv. crossover between them
v. mutation

First, the initial population is selected, and the fitness score is computed based on the ability of each chromosome to compete with others. Interconnecting weights and biases of NN can be encoded as a chromosome. Based on the fitness score, a rank is provided, and chromosomes with better fitness scores have more chances of crossover. During the crossover stage, a random point is selected from the parent gene. From here, the genes of each parent chromosomes are exchanged to produce an offspring. In the mutation stage, some bits of the offspring are flipped, which helps them better the fitness scores. Next, the fitness score of the offspring is calculated to test if they are the next generation chromosomes [35]. This process is repeated for some generations and stops if some specific number of generations are completed, or some condition is reached.

In the particle swarm optimization (PSO) algorithm, particles are used to find the best solution. If the best set of structural parameters for ANN needs to be calculated, this method can be used to optimize it. In PSO. first, the particles are initialized with some random value [36]. Then distance is measured from the goal state to the current state of each particle. There are two values which are the personal best value (PBV) and the global best value (GBV). If the PBV is higher than the particle's previous best values, it updates its PBV. The GBV is the best PBV achieved among all the particles in that iteration. Based on the GBV, velocity is

calculated for each particle, and their data are changed. This process will be terminated when all the particles reach the goal state, or some predefined conditions are met.

Any boosting classifier uses weak classifiers to build a robust classifier. In Adaptive boosting (AdaBoost), some very simple classifiers are selected, which uses specific features to classify data [37]. These classifiers can run very fast. In AdaBoost first, a classifier classifies the data and looks for the misclassified data points. Based on its performance, a weight is given to this classifier. Then the misclassified data points are given more importance, and the classifier's goal is to classify most weighted data points with more accuracy. This process continues, and eventually, a collection of weak classifiers with optimized weights is obtained. This collection of weighted weak classifiers can classify data with high accuracy.

9) Radial Basis Function Neural Network (RBFNN)

This is an NN that works with some variation in the activation function. This RBFNN has three layers that are an input layer which is not weighted and is connected to the input parameters; a hidden layer where the radial basis functions are used as activation function; and an output layer which is fully connected to the hidden layer outputs. The layer structure is represented in Fig. 7(b). In this network, some radial functions, which can be a circle, describe different classes. From the center of the circles, as we go away, the function drops off. This drop-off is commonly represented by exponentially decaying functions like e^{-bd^2}, where b controls the drop-off [38]. If b is significant, then drop-off is sharp. When some input is fed into the network, Euclidean distance is calculated between that point and the center of the radial function. The output of different hidden neurons is multiplied with weights and forwards to the output neuron. In output neurons, the linear activation function is used, and a weighted sum of the output of the previous layer is calculated to generate an output.

10) Probabilistic Neural Network (PNN)

The PNN algorithm works based on the Parzen window classifier. The parent probability density function of all the classes is approximated using a Perzon window function [3]. The Bayes rule is applied to the allocated highest posterior probabilities for each class. The network architecture has an input layer, a pattern layer, a summation layer, and an output layer. The input vector is fed into the input layer. The number of training vectors determines the number of neurons in the pattern layer. The Gaussian function is applied to the Euclidean distance of the input vector and each training vector. This process is similar to RBFNN. In the summation layer, each neuron represents a specific class and computes a weighted sum of all the pattern layer values it is related to. This way, the class, which has maximum pattern output, determines the maximum PDF. In the output layer, the class that has the highest joint PDF is assigned as one, and the rest are assigned zero. The PNN does not use general ANN concepts like learning rules. Fig. 7(c) shows the basic PNN architecture.

C. DEEP MACHINE LEARNING

1) Deep Neural Network (DNN)

This is a subclass of the ANN, which does not need hand-crafted features to be fed into the network as it has the capability of calculating complex features from the data. For unstructured data, DNN works best. A DNN model has a dense architecture of many hidden layers [39]. Each layer is composed of neurons, and the neurons are connected with weighted links and biases. The goal of the network is to optimize them so that they can produce good classification accuracy. A loss or error function like the MSE is defined for this purpose. There are lots of DL-based models such as deep belief network, convolutional neural network (CNN), RNN, and so on. If there are 2^N data points, usually an N number of hidden layers are used. For the CNN, hidden layers compute convolution operations with some fixed filter size and stride. Each neuron has an activation function that is fired when the input to that neuron is over some specified value. Because of complex patterns, DNNs can face difficulties such as overfitting. Regularization techniques like dropout of some neurons can be used for solving this problem. Learning rates and batched processing are used for computational convergence with some optimization algorithms. Batch normalization is used in some cases. Fig. 7(d) shows a DNN architecture.

2) Recurrent neural network (RNN)

Usually, NN does not have any feedback connection from the output layer, for which these algorithms are not suitable for operations where time-series data are involved. RNN works best for activities that include time-series data [40]. Normally, in an RNN, there is more than one recurrent layer. The recurrent layers have feedback connections from the model output. An RNN architecture is shown in Fig. 7(e). On every iteration, the output of the model of the previous iteration is passed through to recurrent layers and to the hidden layer outputs. These outputs are modified by the activation function of the output layer and produce a new output. This whole procedure can be illustrated by Eq. (5).

$$O_i = \sum_{j=1}^{n} f[S_i.W_j + O_{i-1}.W_r] \tag{5}$$

Where O_i is the output of the model after ith iteration, S_i is the input parameters, W_i is the weight of the input layer, and W_r is the weights of the recurrent layer. RNN uses BP to optimize the network [41].

3) Long Short-Term Memory (LSTM)

RNN can be prone to vanishing or exploding gradient problem where the gradient of the error becomes very small or very large. Consequently, the network does not learn anything. It also cannot handle long term dependencies. For solving these problems, the LSTM was introduced. It also has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

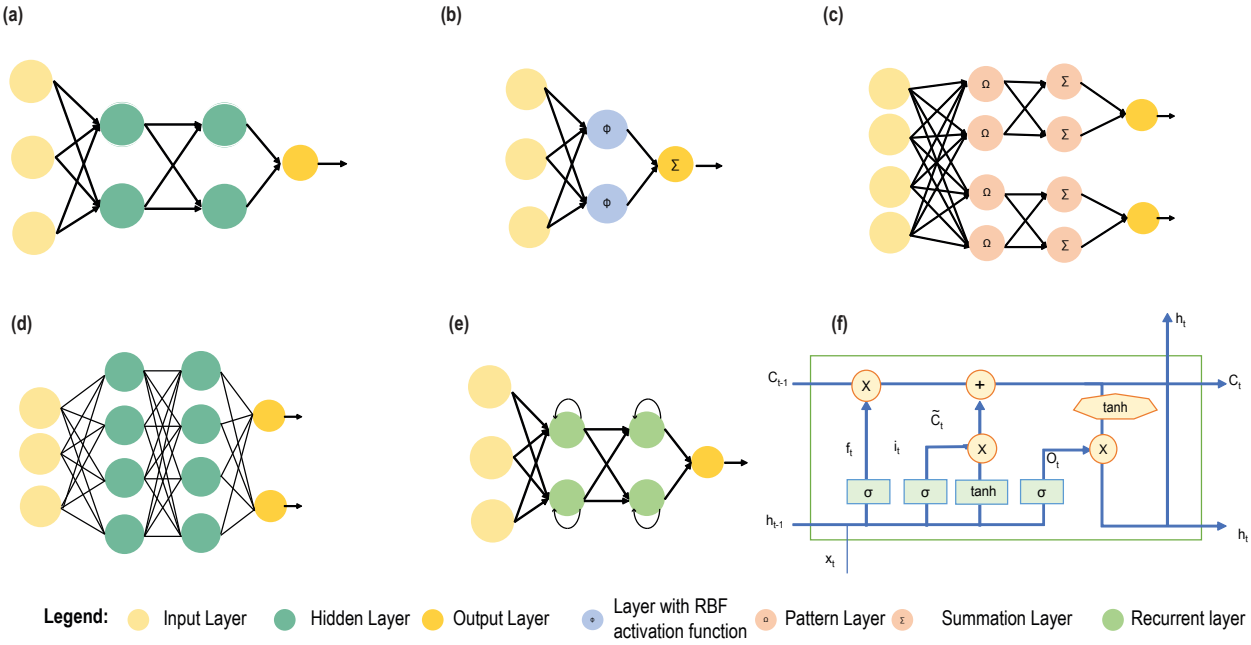

FIGURE 7. (a) An ANN architecture consisting of two hidden layers. In the input layer, there are three nodes, and there are two nodes for each hidden layer. In each node, an activation function takes the input from the previous layer. (b) An RBFNN architecture. The nodes denoted by Φ implements radial functions. (c) A PNN architecture. The pattern layer is denoted by Ω, and the summation layer is denoted by $\sum$. This algorithm uses a probability distribution function. (d) DNN architecture. This network consists of four hidden layers which are fully connected. (e) RNN architecture. There are two recurrent layers which has some feedback connection. These connections help finding historical patterns in data. (f) LSTM cell structure. It uses some parameters which are, h_t (output value of a cell), x_t (input value of a cell), f_t (forget gate value), i_t (input gate value), O_t (output gate value), C_t (cell state), $\tilde{C}_t$ (candidate value). σ, and tanh are the activation functions. The output h_t is calculated as $h_t = tanh(c_t) \times O_t$, where $C_t = f_t C_{t-1} + i_t \tilde{C}_t$.

a chain-like structure and has memory cells which consists of three gates that are the input gate, the forget gate, and the output gate [42].

Fig. 7(f) shows how an LSTM cell works. The forget gate decides how much cell state would be stored. The information that needs to be stored in the cell state happens in two segments. The input gate layer determines the values that need to be updated. A vector of new candidate values is created using a tanh function. These two are summed together and used to update the state. The output gate uses a filtered version of the output. A sigmoid layer determines which portion of the previous cell state will be shown. This portion is multiplied by the tanh function of the cell state to produce the final result. It overcomes vanishing or exploding gradient effect and can learn long term dependencies [43].

Bi-directional LSTM is an LSTM network extension that uses forward and backward pass to retain past and future knowledge. Generally, this network works better than general LSTM because bi-directional LSTM can better grasp function meaning. For this network layout, replication of the same LSTM layer is used, but the input direction is reverse for one layer. Bi-directional LSTM networks tend to do better than one-way classification problems.

IV. AI FOR EARTHQUAKE PREDICTION

A. RULE-BASED APPROACHES

In rule-based approaches of predicting earthquakes, from the knowledge base or from expert opinion, some rules are defined. The input signals are fuzzified by some membership functions so that they can be compared with the rules. The output of this comparison is defuzzified to get the actual output. This process is illustrated in Fig. 8, where the training and testing data flow in a different path to get a prediction of the earthquake. The studies are divided into two categories, which are earthquake's characteristics studies for rule-based approaches, and earthquake and aftershock prediction studies for rule-based approaches.

1) Earthquake's Characteristics Studies

In this portion, the studies that are related to earthquake prediction studies but not performed earthquake prediction are discussed. Earthquakes explanation system, usage of SES, seismic moment studies and prediction of b-value related researches are presented here.

Bofeng and Yue [44] proposed an explanation system named ESEP3.0 for providing information about earthquake prediction to people with different knowledge level. They have used a Fuzzy user model (FUM) based customized explanation system which categorizes users based on their knowledge. First of all, the user's description was given using FUM. FUM used trapezoid membership functions to convert the knowledge level to a fuzzy set. An adaptive interview

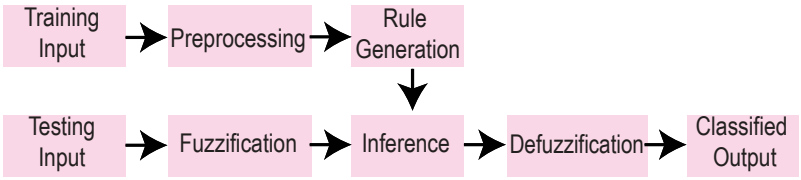

FIGURE 8. Prediction process of rule based approaches. Usually, the dataset is divided into training and testing samples. Based on the training data, rules are generated. The testing samples are fuzzified and compared with the rules to infer an output.

algorithm was used for determining the initial knowledge of the user using random sequences, which consisted of a random interview and adaptive interview. When the user's knowledge increased, their level needed to be changed, which was done by the Adaptive update algorithm. This was done based on the history of the user's dialogue. Based on this updated knowledge level, the explanation about earthquake prediction was provided.

Zhong and Zhang [45] proposed a mathematical prediction model for predicting cause of Reservoir-induced earthquakes in the three-gorge reservoir and its peripheral regions in the Yangtze River using fuzzy theory. They constructed the fuzzy evaluation system based on two main factors: water permeation-accumulations and strain energy accumulation-elimination, which corresponds to another six sub-factors (fractures, fracture angle, rock, water load, karsts, and crack). A fuzzy matrix was developed for weight calculation. These weights were used to compare the sub-factors to find the cause of an earthquake. In the upper level, factors were compared with the immediate lower level factors, and thus a reciprocal matrix was generated. The consistency ratio was showed to indicate the level of consistency of pairwise comparison. The factors for reservoir induced earthquakes were determined based on these pairwise comparisons.

Konstantaras *et al.* [46] proposed an NFM based on an adaptive filter for detecting electric earthquake precursors for the earth's seismic electric signal (SES). They have used two NFMs. In the first model, the effect on the earth's electric field by the magnetotelluric variations was predicted by using the variations of the earth's magnetic field. The magnetic field distribution and electric field distribution of 2 hours of 29 December 2005 from Greece were used for training and testing purposes. They have used 7200 samples from these distributions. Current data, along with the previous three data, were used as input to the model to get the following data. This model used subtractive clustering (SC), and the least square estimator was used for rules generation and membership functions. The second model predicted the electric signal variation by using electric signals when no seismic events were happening in the recorded signal. If there was some residual available, then that could be detected as an anomaly in the SES. Both the NFMs had six layers where, the first layer took input, the second layer assigned membership functions, the third layer guided the data using 16 rules, the fourth layer provided output membership degree, the fifth layer defuzzified the membership degree and layer six provided the crisp output.

Mirrashid *et al.* [47] investigated the capability of ANFIS to predict the next earthquake's potential seismic moment. A dataset consisting of 1480 records that occurred between 1950 and 2013 in the region of Iran was selected for this research. Two seismic indicators, the Gutenberg-Richter b-value and the period of earthquake occurrences, were chosen as input. The logarithm of the cumulative amount of seismic moment between the original event and the future earthquake was the output indicator. A Sugeno type fuzzy system containing five layers was proposed in this research. The antecedent parameters like LSE (Widrow-Hoff learning rate) and the membership functions were trained using the BP algorithm. An SC method based ANFIS was proposed in this research. This generated ANFIS with five fuzzy rules and nine membership functions for each input, considering a range of influence of 0.17.

Rahmat *et al.* [48], compared extreme learning machine and neuro-fuzzy based ANFIS model in prediction of b-value in the Andaman-Nicobar area. The ANFIS had four inputs and one output with five layers. For fuzzification, the bell membership function was used. To adjust the weights, the model used gradient descent with the BP algorithm. The ELM was a single, hidden layer feed-forward network that randomly selected weights to make the model faster and more generalized. It used auto-correlation and normalization before training the model. The ELM parameters were chosen randomly for increasing generalization capabilities. Gradient descent algorithm was used for error correction.

2) Earthquake and Aftershock Prediction Studies

In this portion, the studies that performed earthquake prediction are studied. Earthquake magnitude prediction, time of occurrence prediction, location prediction, aftershock prediction, epicentral prediction-based studies are discussed in this portion.

Ikram and Qamar [49] designed an expert system that could predict earthquakes at least twelve hours before its occurrence. The data were selected from the USGS repository, and the necessary features were selected. Data from all around the world were considered for this system. Based on the location of the epicenter, the world was divided into 12 portions. Then they used frequency pattern mining with the help of a frequency pattern growth algorithm. This

algorithm used a divide-and-conquer method and generated a frequency pattern tree to find fifty-six frequent items. From that, eighty rules were derived. The rules were then converted to inferential rules and then to predicate logic. Then they found only seventeen distinct rules. The rule-based expert system was composed of a user interface, a knowledge base, and an inference engine. The inference engine matched the input from the user with each rule in the knowledge base and predicted the magnitude, location, and depth of the next earthquake.

Konstantaras *et al.* [50] have proposed a hybrid NFM to predict the interval between two high magnitude seismic events in the southern Hellenic arc. They tried to draw a relation between the seismic frequency and the occurrence of high magnitude seismic events. The smaller seismic events accumulate energy in the earth's crust, and a series of these events leads to a high magnitude seismic event. There were four inputs to the fuzzy system, which were related to the mean seismic rates and duration between two seismic events having a magnitude greater than 5.9. In a fuzzy network, the number of neurons depends on the input and number of membership functions. The proposed fuzzy network worked in a similar way of the feed-forward network where weight was updated based on the difference of actual output and expected output. They trained the NFM for 20 epochs, where the error was ideally zero. There was only one output neuron that provided the date of the next big seismic event.

Dehbozorgi and Farokhi [51] proposed a neuro-fuzzy classifier for predicting short-term earthquakes using seismogram data five minutes before the earthquake. The equal number of seismogram signals were selected, which have and do not have an earthquake after five minutes. The selected data were from Iranian region and to give them as an input to the model, they were sliced. The baseline drift of the signals was removed by the fourth-order Butterworth high pass filter, which normalized the data. Fifty-four features were calculated by statistical analysis, wavelet transformation by Daubechies-two methods, fast Fourier transform, entropy calculation, and power spectral density calculation. There were sixty rules for the NFM, which was compared with a multi-layer perceptron (MLP) with two hidden layers having thirty neurons. After training, the feature selection was performed using the UTA algorithm which replaces a feature with the mean value of that feature and measures the performance. If the performance decreases, then the feature is considered as an important one. This feature selection procedure improved the base model.

ANFIS is a very popular model among the earthquake prediction researchers as many models were developed based on them [52]- [57]. Zeng *et al.* [52] proposed an adaptive fuzzy inference model to predict epicentral intensity. They used the magnitude and depth of hypocenter as input to the model, and the model provided the intensity of hypocenter as output. The data from the Sichuan province of China from the year 2004 to the year 2015 were used for this study. They also calculated the mean of the magnitudes and variance of the substantial magnitudes. The membership function was ridge shaped, and the membership degree was one as the mean was in the center, and the variance was the same as the width. The samples were classified according to the magnitude A_i and the depth of hypocenter B_j. Then they calculated the mean of magnitude, depth, and epicentral intensity. When there were less than three samples, the mean of epicentral intensity was adjusted by using the growth rate.

Andalib *et al.* [53] came up with the idea of using a Fuzzy expert system for solving the problem of predicting time between two earthquakes and their distance of occurrence. Sugeno type ANFIS was used for this purpose. The data were collected from the Zagros earthquake catalog and used four parameters as input to the inference system. The input parameters were the magnitude of two earthquakes, time distance, and geographical distance. The knowledge of the human experts were used to generate the rules for the ANFIS as this model tries to replicate the performance of the human experts. Expert opinion was used to fuzzify the crisp inputs as well. Based on the rules of the FIS the crisp outputs were generated which was the prediction of an earthquake before 6 months. This model considered the most powerful earthquake in the area and compared it with a magnitude threshold M and distance N miles. The value of N and M were optimized for the prediction.

Shodiq *et al.* [54] came up with the idea of using a combination of automatic clustering and ANFIS for earthquake prediction. The proposed model involves pre-processing, automatic clustering, and ANFIS. Automatic clustering used hill climbing and valley tracing for finding the optimum number of clusters, and for the clustering between zones, the K-means algorithm was used. The data were clustered in seven zones. They selected the magnitude of completeness as 5.1 Mw. In the ANFIS portion, seismic indicators were calculated and normalized within a range of 0.1 to 0.9. The ANFIS model used a Sugeno model having five layers consisting of adaptive nodes and fixed nodes. They have used data of Indonesia from the year 2010 to the year 2017. The ANFIS used 2 Gaussian membership function to get the membership degree. The model was trained with 100 epochs to predict the occurrence and non-occurrence of an earthquake.

Kamath and Kamat [55] tried to predict the magnitude of the next earthquake using ANFIS. Here the ANFIS had five layers, and the Takagi Sugeno type FIS was used. Data from Andaman and Nicobar Island was used for training and testing purposes. While clustering, they checked SC and grid partitioning (GP) for building the initial FIS. They chose Triangular and Gaussian shapes for input membership function. The parameters of the membership functions were optimized using BP and hybrid algorithms. The SC performed best with 8 fuzzy rules and 70 neurons, which varied the influence and squash factor. The model was trained for 50 epochs, and the squash factor was selected as 1.25 for the training process. Pandit and Biswal [56] proposed ANFIS with GP and SC for predicting the magnitude of an earthquake. Ground motions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

TABLE 2. A summary of used algorithms and features by rule-based earthquake prediction approaches

Ref.	Used Algorithm	Used Features
[44]	FUMCE	pk., sk., cpk., knowledge learned by fam.
[45]	HDFT	seaef., wpaf.
[49]	Rule-based Expert System	lo., la., dpt., mg., intensity
[46]	NFM	SES
[50]		Mean seismicity rates and time intervals
[51]	NFC	Statistical features, FFT, PSD, entropy
[52]	ANFIS	mg. and dpt. of hypocenter
[53]		td., m2e., gd.
[54]		Dynamic GR's law, increments of b value, OU's law
[55]		dpt., lo., la.
[56]		Epicentral distance, time, pga, mg.
[47]		Time, logarithm of mean annual rate of exceedance
[57]		Location and logarithm of seismic moment
[58]	GFCV	Seismograph
[48]	ANFIS and ELM	b-value parameters

Legends: Ref.- References; GR– Gutenberg Richter; pk.– precursory knowledge; sk.– seismology knowledge; cpk.– comprehensive prediction knowledge; fam.– Fuzzy association model; seaf.– strain energy accumulation-elimination factor; wpaf.– water permeation-accumulation factor; lo.– longitude; la.– latitude; dpt.– depth; mg.– magnitude; SES– seismic electric signal; FFT– Fast Fourier Transform; PSD– Power Spectral Density; td.– time distance; m2e.– magnitude of two earthquakes; gd.– geographical distance; pga– peak ground acceleration.

of 45 earthquakes from the USA, Canada, Japan, Mexico, and Yugoslavia were selected as a dataset. In GP, a uniformly portioned grid with defined membership functions and parameters were generated to produce an initial FIS. In SC, each data point was selected as a cluster center according to the surrounding data point's density and found out the form and an optimum number of fuzzy rules. Triangular membership function was used to generate the membership degree. Hybrid optimization and BP were used for training using MATLAB GUI.

Mirrashid [57] proposed ANFIS for the prediction of the earthquake over the magnitude of 5.4. Earthquake catalog from the year 1950 to 2013 in the Iranian region was used as a dataset. The dataset had different magnitude scales which were converted to the moment magnitude scale. They used seismicity indicators as input to the model and normalized them between 0.1 and 0.9. Generally, the seismicity indicators are elapsed time (t_e), mean magnitude ($\bar{A}_e$), earthquake energy (E_e), slope of magnitude (dB) ($\frac{d\bar{A}_e}{dt}(dB)$), mean square deviation (Δ_e), meantime ($\bar{t_e}$), magnitude deficit (δA_e), and coefficient of variation (ρ). The ANFIS that they proposed was the Sugeno type five-layer model, which is composed of BP and least square estimates. GP, SC, and fuzzy c-mean algorithms were used along with ANFIS. The GP model used a predefined number of membership functions to divide the data spaces into rectangular sub-spaces. The SC algorithm calculated the potential of each data point for being the cluster center. The unsupervised FCM learning algorithm considered each data to be part of all classes.

Bahrami and Shafiee [58] proposed a fuzzy descriptor model with a generalized fuzzy clustering variety (GFCV) algorithm to forecast earthquakes in Iran. Linear descriptor systems and fuzzy validity functions were used to divide the input space of a fuzzy descriptor into linear sub-spaces. The normalized Gaussian type functions were used as validity functions. This model is an extension of the Takagi-Sugeno fuzzy model. The linear system descriptors and validity functions were adjusted using the GFCV algorithm, which calculated MSE and stopped the system's training when it started to increase. They have used 560 seconds of seismograms signal of an earthquake sampled at 50 Hz.The model used 7 neurons and was trained for 7 epochs only to produce a better result than other fuzzy descriptor algorithms.

Table 2 summarizes the used algorithms and the features of the studies that used rule-based prediction approaches.

B. SHALLOW MACHINE LEARNING

In shallow ML, there are classical ML approaches, clustering approaches, and NN-based approaches. The classical ML algorithms such as SVM, SVR, KNN, RF, DT, etc., use handcrafted features for prediction of an earthquake. As they cannot generate feature themselves, feature selection is an essential aspect of this prediction process. Fig. 9 shows a basic diagram of these algorithms classifying earthquake events. This section will be divided into two categories to keep similar studies under the same hood. The categories are- earthquake and aftershock prediction studies for shallow machine learning-based techniques, and earthquake's characteristics studies for shallow machine learning-based studies.

1) Earthquake and Aftershock Prediction Studies

In this portion, the studies that performed earthquake and aftershock prediction are studied. Earthquake magnitude prediction, time of occurrence prediction, location prediction, earthquake detection, aftershock prediction, and energy prediction-based studies are discussed in this portion.

Jiang *et al.* [59] tried to predict the most significant annual magnitude of the earthquake in China synthetically.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

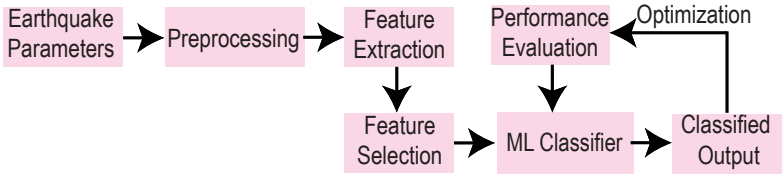

FIGURE 9. Earthquake prediction process of classical ML approaches. First, the earthquake parameters are preprocessed to remove missing values. Then features are calculated from them. Selected features are fed to the ML algorithms to provide an output. Based on the performance, the hyperparameters of the algorithms are changed.

Different seismic precursors such as stress, water level, hydro-chemistry, gravity were collected for the north region of China, and Beijing. Their choice of the algorithm was SVM, and they have selected twelve seismic precursors as features. The SVM algorithm maps the sample space into a high dimensional Eigenspace with the use of nonlinear functions. Since seismic events are very nonlinear, SVM helps in predicting them accurately. The SVM model used the polynomial kernel function and tried to optimize the value C, which is the punishment of samples for which the error is more than ϵ. The suitable value for ϵ was found to be 0.6707.

Astuti *et al.* [60] proposed a method to predict earthquake location, magnitude, and time using singular value decomposition (SVD) and SVM. They have used the earth's electric field signal as input to the model. They have used data from Greece for the year 2003 to the year 2010. In the preprocessing stage, the E-W and N-S pole field values were squared and summed, and then their root was calculated. The Gfdiff was calculated from the input signal, which was the difference between the nth sample and $(n\text{-}1)$th sample electric field. The peak from the Gfdiff was calculated, and the slope to the next day's peak was captured. For feature extraction, first SVD was applied for orthogonal transformation, and segments of Gfdiffs of 180 samples were separated. The LPC coefficients were found using the Levinson-Durbin algorithm. Then the features were used as input to the SVM classifier, where a hyperplane was determined to separate the data into different classes. The optimization was done using the Lagrange multiplier, and for nonlinearity, kernel functions were used.

Hajikhodaverdikhan *et al.* [61], proposed an SVR model optimized by particle filter to predict mean magnitude and number of earthquakes in the next month in Iran. They have evaluated 30 precursors for this study. In SVR, the model searches for a hyperplane that can separate the dataset into different portions based on classes. Particle filters estimate the state of a linear system and convert it to have some randomness in the presence of noise. SVR has some parameters like C, ϵ, and kernel scale. When C value increases, the generalization of that model decreases, but error performance increases. ϵ represents loss function whose lower value is desired, but if it is zero, than there may be some overfitting present. In this model, Gaussian RBF was used as a kernel filter. These three parameters were selected using the particle filter by calculating probability density function with particle weights. The kernel width, C, and ϵ are the parameters that were optimized by the PSO to improve performance of SVR.

Asim *et al.* [62] proposed a hybrid algorithm of SVR and NNs to predict earthquake over magnitude five in Hindukush, Chile, and South California. The cutoff magnitude for Hindukush, Chile, and South California are 2.6M, 3.4M, and 4.0M, respectively, which are calculated from the GR-curve. They have calculated 60 parametric and non-parametric features. The maximum-relevance-and-minimum-redundancy feature selection technique was used for each region, and separate features were selected for a different region. Based on the features, the input vector was given to the SVR model. The output was used as the input of the LM-NN. The weights were passed to the Quasi newton network and from that to the Bayesian regularization NN. With each NN to escape local minima, an enhanced PSO was used. MCC was chosen as optimization criteria for PSO. It optimized the hyper-parameters of SVR to increase its efficiency.

Li and Kang [63] proposed a hybrid algorithm of KNN and Polynomial regression (PR) to predict the aftershock of an earthquake. The time intervals of aftershocks were the conditional attribute, which was converted to seconds, and aftershock magnitude was the decision attribute. They have collected the time intervals of the earthquake aftershocks of the Wenchuan region of China. The shortest distance of a sample to other samples was calculated using Euclidean distance, and these values were sorted to find K neighbors. The decision attributes were modeled by PR, calculating the least square estimation of the coefficient vector. Then the model was compared with the regular KNN and distance weighted KNN based on absolute error (AE) and RE.

Prasad *et al.* [64] proposed a seismic wave-based earthquake detection method that used Haar Wavelet transformation (HWT) for denoising purposes. They collected seismic signal of 140 earthquakes from different sources. These data were de-noised using HWT. The next step was to apply a fast Fourier transformation spectrum analysis to calculate the energy and frequency of the concerned signal. Using this energy E of the signal, the magnitude M was calculated with the formula Eq. (6) [65].

$$M = |(\log E - 11.8)/1.5| \tag{6}$$

If the magnitude was greater than three, then it was called an earthquake. These data were then selected as a dataset for different ML algorithms, such as- RF, Naïve Bayes (NB), j48, REP tree, and BP.

Sikder and Munakata [66] tried two algorithms that are rough set and DT for identifying their performance in predicting an earthquake. They have used fifteen attributes related to Radon concentration and climatic factors. For the Rough set, the decision table was generated with these fifteen attributes. They used 155 records of weekly geo-climatic conditions regarding earthquake. Then approximation of each decision class was constructed considering all the conditional attributes. They have used the discernibility matrix algorithm to find the smallest subset of attributes that represents the data. 440 reducts were found with whom fifteen decision rules were made using the learning from example, module 2 algorithm. For the DT algorithm, tree-building was done using the C4.5 learning scheme. For both cases, the model was validated using leave one out cross-validation method.

Marisa *et al.* [67] tried to predict the probability of the number of earthquakes in 15 days period in Sumatra. First, the earthquakes with a magnitude greater than 4.9 and depth less than 70 km were selected. They collected earthquake catalog of Sumatra island from the year 2008 to the year 2018. Then the variance value and average value were calculated. The variance value was 3.463, which was higher than the average value of 1.676. This indicated that there was oversampling. Hence, they should have used a Poisson mixture model, but the earthquake data showed auto-correlation. So, they selected Poisson Hidden Markov Model (PHMM). The model parameters were selected using the Expectation-Maximization (EM) model and parameters with two, three, and four hidden models were selected. With the model, they predicted the earthquakes for 2017 and 2018, compared them with the actual prediction, and used the chi-square test for validation.

In clustering-based studies, first, the study area is divided into different clusters. The data which are of the same quality and characteristics are clustered in a group. Based on individual clusters, the decision making process changes. Fig. 10 shows this process of clustering, leading to earthquake prediction.

Florido *et al.* [68] proposed a clustering-based model for the prediction of an earthquake in Chile. First of all, in the preprocessing stage, they removed all the foreshock and aftershocks from the dataset. This is done to get only the mainshocks. They collected data for Talca, Pichilemu, Santiago, and Valparaiso region for this study. Then they calculated the b-value, magnitude, and time of occurrence to get the dataset prepared. For calculating the b-value, 50 earthquakes with a magnitude of more than three were used. Then they used the clustering algorithm to make clusters that need the optimum value of the number of clusters. Silhouette index was used to get the value. Then using Euclidean distance for the K-means, the clusters were defined. For earthquakes with a magnitude of more than 4.4, patterns were searched, which can be used for predicting future earthquakes.

Florido *et al.* [69] have used a tree-based algorithm that used exhaustive search for predicting an earthquake in seven different datasets. They used an earthquake catalog to generate a propositional dataset which has Gutenberg-Richter parameters to predict the earthquake with the highest magnitude. They have collected earthquake catalog of Chile, Iberian Peninsula, and Japan for this study. From this dataset, for training purposes, clustering and grouping were done. For this grouping, the number of clusters (K) and the length of the pattern (A) were used. With exhaustive searching, the best combination of A and K was obtained. Then a tree of the precursors was extracted, and patterns were captured using the best precursors from each sub-tree. Based on these procedures, an earthquake of the next day was predicted.

Hierarchical K-means algorithm was used by Shodiq *et al.* for researches [70]- [71]. Shodiq *et al.* [70] proposed a spatial analysis and automatic clustering-based technique to predict earthquake's time period. They collected earthquake data from the year 2004 to the year 2014 for the Indonesian region. At first, they collected data, normalized, and vectorized it. Then they used the valley tracing technique. They calculated the accuracy of clustering and found that six clusters can optimally separate the data. Then hierarchical K-means clustering is done where centroid K-means outperformed the other K-means. Then the probability is calculated using Eq. (7).

$$P(M,T) = (1 - e^{-N(M)T}) \tag{7}$$

Where P is the probability of an earthquake with magnitude M and time period T, and $N(M)$ is index seismicity. Then they divided the dataset into learning samples and testing samples.

Shodiq *et al.* [72] also proposed an automatic clustering and ANN-based method to predict an earthquake of magnitude higher than 5.5 within 5 days of the occurrence of an earthquake. They have collected information about 82,580 in the Indonesian region from the year 1910 to the year 2017. At first, they preprocessed the dataset and prepared the dataset for clustering. The optimum number of clusters was obtained using valley tracing and hill-climbing algorithms and then clustered the data using the K-means clustering algorithm. Then they sub-clustered the seismic parameters and calculated the input parameters. There were seven input parameters, and five of them were obtained using the Gutenberg-Richter law. Then a NN consisting of seven input neurons, two hidden layers where each layer had thirty-two neurons, and one output was used to predict the earthquake. They performed 100,000 training iterations with a learning rate of 0.01 to the BP algorithm.

Shodiq *et al.* [71] proposed a hierarchical K-means clustering algorithm for predicting aftershocks of earthquakes within five days. The earthquakes of magnitude greater or equal to 5.1 were selected for calculating seismicity parameters. This research used earthquake catalog of 1910 to 2017 for Indonesia by the BMKG and USGS repositories.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

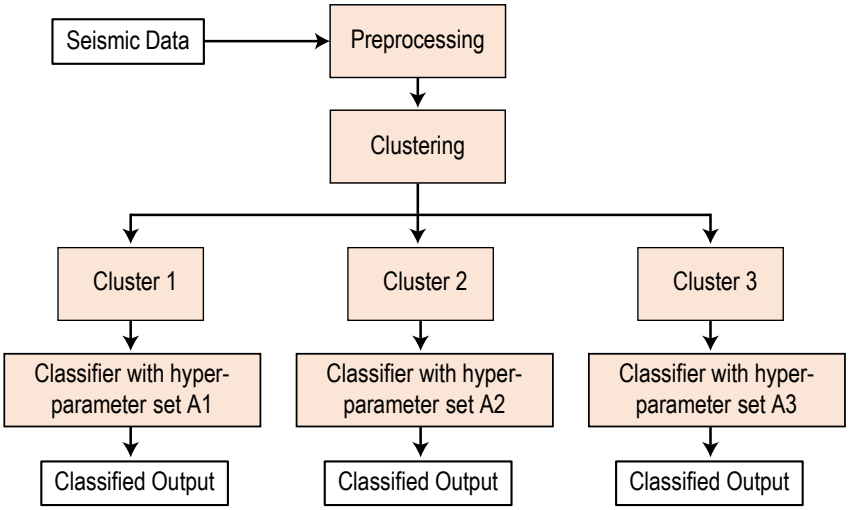

FIGURE 10. Earthquake prediction process using clustering approaches. The processed seismic data are given as input to the clustering layer, which divides them into different clusters. For example, here, Cluster 1, Cluster 2, and Cluster 3 are three clusters calculated from the seismic data. Each cluster is classified using different versions of the classification algorithm based on their characteristics, which produce the desired output.

For clustering the data based on epicenter parameters, the first valley tracing and the hill-climbing algorithm were used to find the optimal number of clusters which analyzed the moving variance of the clusters and used it to find local and global minima. The optimal cluster number was six. Then hierarchical clustering was used for clustering the data into six clusters. An NN was used to predict aftershocks. The number of hidden layers and neurons was found by trial and error. The best combination had two hidden layers, with thirty-two neurons in each, having a learning rate of 0.1. This model performed better for earthquakes with a higher magnitude than 6.

Mejia *et al.* [73] used the Dobrovolsky clustering and Kannan-Mathematical model for predicting earthquakes with a magnitude of more than 3.9 in the Philippines. The data for earthquakes in the year 2011 to the year 2013 were collected for this research. Since there are many fault lines in this area, the earth's electric field was used to cluster the area. To find the radial area, Dobrovolsky-Megathrust equation was used. The epicenter of the megathrust was found using the latitudinal cluster's geometrical center. Then Kannan-Mathematical model was applied, which is based on the spatial connection theory. According to this theory, the earthquakes in the same fault line are connected historically. So, they calculated the Poisson range identifier (PRI) in this work. Poisson distribution was used to calculate distance factors. The distance factor and recursive Poisson distribution were used to predict an earthquake.

Tan and Cai [74] proposed an agglomerative hierarchical clustering (AHC) based earthquake prediction algorithm that can reduce the effect of an imbalanced dataset. They analyzed the earthquake events in the Yunnan Region between 1970 and 1998, and they only took the events whose magnitude was greater than 3.5. They found that positive classes are in very fewer numbers compared to negative classes, which make the dataset imbalanced. They imposed the AHC method, which was proposed by Cohen to oversample positive class and then applied Biased SVM for classification purposes. AHC constructed the dendrogram using single linkage, group average or multiple linkage algorithm and calculated centroids from all the clusters. The centroids were joined with the minority samples and dealt with the data imbalance.

Lu *et al.* [75] proposed a GA-based clustering algorithm to predict the location and timing of earthquake aftershocks. They have used seismographs of Southern California and Japan, which was used to create a directed graph. The correlation between each node was calculated, which represents relation tremors and time-stamps of two points. The connection weights were converted to one or zero based on whether there was a connection between the two points or not. After doing clustering on this directed graph using a GA, a central point was found, which was the epicenter of the main earthquake. From here, some sub-clusters can be found, which reoccurred in different places of the graph. A regression algorithm was used to predict future re-occurrence of the aftershocks.

Shao *et al.* [76] proposed an ant-colony clustering (ACC)-based method for earthquake prediction. At first, they selected the duration of 14 different anomalies and classified them into seismological and precursory anomalies. The number of anomalies was also considered as input. They classified the magnitudes into three groups and the ACC was done. In the ACC, an ant calculates its similarity among its neighborhood and decides whether to make a cluster with it or not. The agent can change the similarity and exert its influence in clustering. The parameters they used were α = 0.8, k +=0.8, and k-= 0.2 for the optimized model.

TABLE 3. A summary of used methodologies and features by the earthquake prediction researches that included classical ML and clustering techniques

Ref.	Used Algorithm	Used Features
[59]	SVM	wl., ems., hc., tlt., sls., adsg., b-value, etc.
[60]	SVD and SVM	Time, East-West and North-South polarity
[61]	SVR	mmt., awsp., mg., mean temperature
[62]	SVR and HNN	GR law, ser., ff., trt., src.
[63]	PR-KNN	Time interval between aftershocks
[64]	HWT with RF, NB, J48, REP Tree, and BP	Seismic signal
[66]	RST and DT (C4.5)	Location and climatic factors
[67]	PHMM	Initial probability, transition probability
[68]	KMC	b-value, date of occurrence, mg.
[69]	KMC and TBS	Seismicity indicators, increments of b-value
[71]	HKMC with ANN	Seven seismicity indicators
[72]		b-value, Bath's law, and OU's law parameters
[70]	HKMC	lo., la.
[73]	Dobrovolsky-based Clustering	SES
[74]	AHC Over Sampling	Seismicity indicators
[75]	GA based Clustering	Time, lo., la., tremor mg., dpt. of seismograph
[76]	ACC algorithm	Belt, water level, b-value, tilt, frequency etc.
[77]	PSO Clustering Algorithm	Seismometry index and precursory index

Legends: Ref.– Reference; TBS– Tree-based search; GR– Gutenberg Richter; wl.– water level; ems.– electromagnetism; hc.– hydro chemistry; tlt.– Tilt; sls.– Short leveling surveys; adsg.– abnormal distribution of seismic gap; mmt.– Maximum and minimum temperature; awsp.– average wind speed and precipitation; mg.– magnitude; ser.– seismic energy release; ff.– foreshock frequency; trt.– total recurrence time; src.– Seismic rate change; lo.– longitude; la.– latitude; SES– seismic electric signal; dpt.– depth.

Zhang *et al.* [77] proposed a PSO based clustering technique to predict earthquake and compared the proposed model with a model using the K-means clustering technique. PSO has fewer parameters and can solve multi-domain and nonlinear problems. In the research work, 14 different abnormal indices were considered for earthquake prediction. This research tried to find a connection between the earthquake magnitude and precursors such as short levelling, belt, and seismic gap. At first, they normalized the data to remove the dimensional effect. Then parameters were initialized automatically, and the PSO algorithm was applied to it. Then an evaluation function was designed. They set the number of particle swarms to 500, α = 0.8, k+= 0.8, k-= 0.2 for finding optimized output. Table 3 shows the research methods and used parameters by the classical ML-based and clustering-based researches.

The NN-based methods for prediction of earthquakes use an architecture that consists of weights, biases, connections, and neurons. In each neuron, there is an activation function that is connected to links from the previous layer. This activation function fires when the value of the input is higher than the threshold of the activation function. An error is calculated using the predicted output and actual output. Some optimization function try to minimize the error value by updating the weights and biases of the network to achieve the final output. Fig. 11 depicts this process of output generation by a NN-based model.

ANN-based researches are used in many cases for prediction of earthquake [78], [79]. Abdulrahman *et al.* [78] tried to find out the best feed-forward NN for prediction of an earthquake. After collecting data from the earthquake catalog of the northern Red Sea area, first, they filtered the noise from the data and then reformatted the location parameter with 16×16 tallies. This reduced the possibility of different locations by 99.84%. They calculated different statistical features on the different features. They calculated earthquake sequence number, tile location, earthquake magnitude, and the depth of source to use them as features. Then they tried different delays, the number of hidden layer neurons, and activation functions and calculated MSE and MAE to find the best combination. The best combination was found with a delay of three. The number of hidden layers was two. The first hidden layer had three neurons, the second had two, and the activation function is Tansig.

Reyes *et al.* [80] proposed a NN-based model that could predict earthquakes more than a specific threshold five days before the earthquake in Chile. Since Chile is a country with a large area and the seismicity of different areas does not match, they divided the country into six different regions based on Kohonen-self organized map. Among them, four cities (Talca, Pichilemu, Santiago, and Valparaiso) were selected for analysis using four different ANNs. All four ANNs had the same structure, but the weights were different. The input layer of the ANN had seven inputs, and the hidden layer had fifteen neurons. The learning was done using BP with a sigmoid activation function. The model was then compared with SVM, KNN, and K-means algorithm based on zero level hit, one level hit, Sp, and Sn.

Morales-Esteban *et al.* [81] proposed an ANN for prediction of earthquake magnitude greater than a threshold or between some range in the area of the Alboran Sea and Azores-Gibraltar Fault. For training the model of the Alboran Sea, data from December 2004 to May 2005 were used whereas for Azores-Gibraltar earthquakes. data from July 2003 to June 2005 were considered. For this model, as the dataset behaves like episodes of events, no preprocessing was done. They used a three-layer feed-forward network with a BP learning algorithm. The hidden layer has seven inputs and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

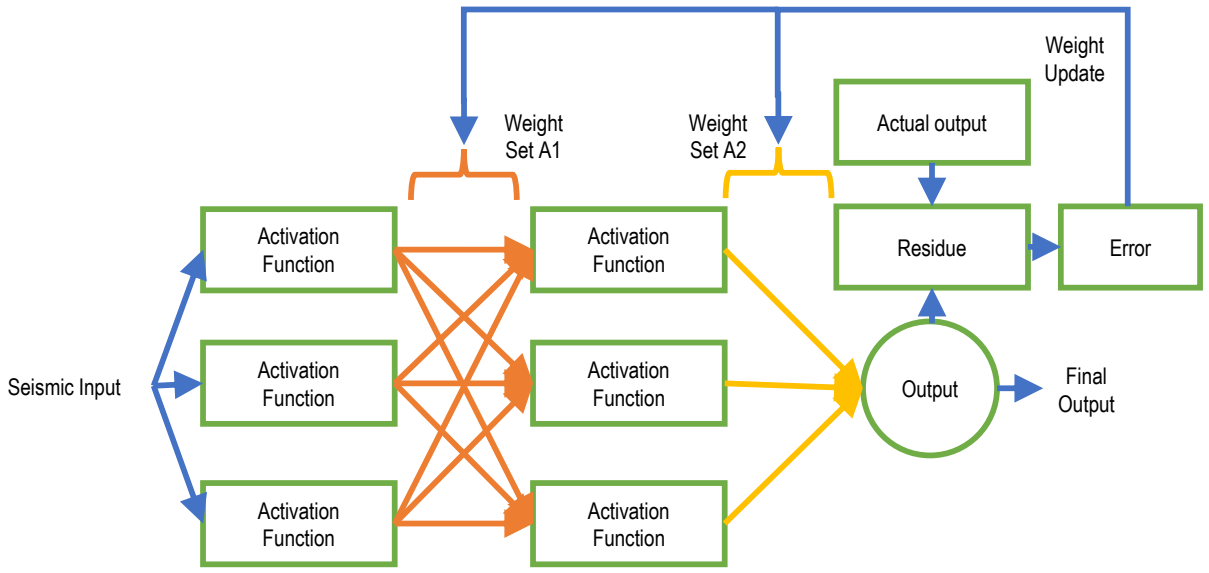

FIGURE 11. Earthquake prediction process of a NN-based model. The seismic inputs of the network go through the activation functions to produce some output which is multiplied by weights. The difference between the final output and the actual output is the error which is used to adjust the weights to predict earthquake more accurately.

fifteen neurons. Hence, the number of neuron connections was 122. The model was trained with 300 epochs, and the activation function was sigmoid shaped.

Moustra *et al.* [82] used time-series magnitude and SES for prediction of the most significant seismic event in the next day. They used data of Greece for the year 1980 to the year 2001. First, they used a three-layer perceptron model with a learning rate of 0.4, a momentum of 0.8, and the optimal width of time moving window before the concerned day was 30 days. The outliers of the data were the seismic events, which had a magnitude greater than 5.2. Next, they used only the outliers for training all the data. Afterwards, they used seismic signals calculated by the VAN team, which had only 29 instances. These seismic signals were collected between the year 1983 and the year 1997. They randomly selected the values for other 157 events, and the model did not work out. Then they used the NN model to work reversely to generate the SES for the 157 events, which led to excellent results.

Xu *et al.* [83] proposed a method for predicting earthquakes, which are greater than a magnitude of 5, using DEMETER satellite data and seismic belt information. They first converted the DEMETER data to raster data and considered a lithospheric abnormal electrical signal of ($30km \times 30km$) area around the epicenter. The variations of electron and ion densities within 30 days of the earthquakes were used. For seismic belt information, if the epicenter of an earthquake was within 60 km of any seismic belt, then the parameter was set to one. They also calculated global background signal based on seasons when there were no seismic events available. They used a three-layer BPNN with a linear activation function. The number of hidden neurons was 20, and the output activation function was sigmoid.

Narayanakumar and Raja [84] have proposed a BPNN model for the prediction of an earthquake in the Himalayan region of India. They have used eight seismicity indicators, which have been calculated from the earthquake catalog using Gutenberg-Richter law. The earthquakes over 2.5 magnitudes were considered for this study, which was collected from the Global Hypocenter Database. The eight input seismicity parameters were calculated using data of the Himalayas from the year 1987 to the year 2015. Then they have used a NN with input neurons having nine neurons, a hidden layer with twelve neurons, and an output layer with a single neuron. The choice of activation function was Tan-Sigmoid and Purelin. They trained the model with the LM algorithm with 10,000 epochs. The initial learning rate was 0.01, and the momentum was set to 0.94.

Cheraghi and Ghanbari [85], predicted the timing of an earthquake and its energy using an NN. They have used earthquake timing, depth, magnitude, and distance to the fault line from the epicenter as input. At first, all the data were normalized at first. The NN used a BP algorithm as a learning algorithm, and MSE was used for error calculation. The NN had two hidden layers with three and two neurons, respectively. A sigmoid function was used as an activation function that brings nonlinearity to the network. For energy prediction, they used the predicted magnitude. The initial learning rate was set to 0.001 and the momentum of the first, second, and output layer was 0.02, 0.01 and 0 respectively.

Xie *et al.* [86] proposed a NN model for predicting an earthquake in the East Chinese region. For this, at first, they collected the data from CENC and removed the aftershocks by using *K-K* rule. Then two seismicity rate variations V_1 ($3.5 < M < 4$) and V_2 ($M < 4.5$) were calculated. These

variation rates and time intervals were the input to the BPNN. The model provided the time of occurrence of the earthquake.

Hu *et al.* [79] proposed an integrated NN and LR-based model to predict the magnitude of an earthquake. They collected earthquake catalog of north and south China from the year 1970 to the year 2000. They calculated seismic indicator for the range of six months leading to 61 samples. They selected 51 samples for training the models and 10 samples for testing. They evaluated the performance of LR and the performance of three-layered BPNN where the parameters were the input. They calculated the regression prediction value (ML_R). After that, they found the difference between regression value and actual magnitude (ΔML_R). Next, the BPNN took 13 parameters as input: six seismic parameters, six cosmological parameters, and the difference between actual magnitude and regression magnitude. They set the number of neurons in hidden layers to 25, and the output layer provided the predicted deviation (ΔML_N). The final estimated output was calculated using Eq. (8).

$$ML_{RN} = ML_R + \Delta ML_N \tag{8}$$

Some of the earthquake researches are based on LM algorithm [87], [32], [88]. For instance, Zhou *et al.* [87] proposed an LM-BP based model to predict the magnitude of an earthquake. Usually, BP models face problems such as slow learning and can not find the global minima. LM algorithm can solve this problem. This model converges to the goal at a faster speed and significantly reduces the problem of not reaching global minima. They proposed this model to have one input layer with seven inputs, one hidden layer with sixteen neurons, and one output layer. They tried different combinations of hidden layer neurons for achieving the goal. The S-tangent transfer function was used as an activation function for the hidden layer, and S-logarithmic was used for output layer.

Kulahci *et al.* [32] proposed a three-layer feed-forward NN algorithm for earthquake prediction using the monitoring of Radon gas concentration. Data have been collected from the Turkish-Arabian fault boundary, where Radon started to concentrate from 2004. They have used eight parameters that are related to Radon concentration and soil temperature. By using statistical analysis, they saw that two-third of the attributes are essential, and they must be used in every case. The model uses gradient-descent for weight update and LM algorithm for BP procedure as it converges fast. The transfer function used for the input layer to the hidden layer was tangent sigmoid function, and the hidden layer to the output layer is the Purelin function. To avoid overfitting, automated training-testing was used while training where the training process halts at some random time to check error.

Ma *et al.* [88] proposed a BP-based model using the parameters from the DEMETER satellite for prediction earthquake before thirty days. The DEMETER satellite provides data of seismic electromagnetic radiations like ion and electron densities, and temperatures. When there is no earthquake distribution of wave intensity, they called it background, which gets affected by environmental parameters. To predict an earthquake, they removed the background from the signal. Then they used it as input to the three-layer perceptron model. The LM algorithm was used for learning as it works fast. The MAE was used for error calculation, and the activation function was sigmoid to add nonlinearity. The validation of the model was performed based on the seismic events happened in the year 2008.

Hu *et al.* [89] have used a hybrid algorithm of linear regression (LR) and BP to predict earthquakes in three months of range. The study area for this research was Beijing and they collected earthquake catalog of the year 1990 to the year 2003. They converted the different earthquake magnitude scales to the Richter scale value. They have used five seismic parameters and two-time varying parameters for training the model. First, LR was used to calculate M_R (regression prediction value) based on the seismic parameters. Then ΔM_R was calculated as $\Delta M_R = M_A - M_R$, where M_A is the actual magnitude of the earthquake. After that, BP was used to predict the magnitude of the earthquake, which is ΔM_N. Then they integrated both the models, and the final output of the integrated model (M_I) was calculated using Eq. (9).

$$M_I = M_R + \Delta M_R \tag{9}$$

Zhou *et al.* [95] proposed a NN and an SVM model to predict the magnitude of seismic events. They also showed that the combination of these two algorithms performed better than they performed independently. They collected data around the world from the year 1999 to the year 2016 and selected latitude, longitude, and focal depth as training factors and magnitude as testing factor in achieving data reduction. They trained the SVM model with a linear kernel, and the cost function of the violation was set to 10. They also trained an error inverse propagation-based NN with 5000 iterations. They combined the result using Eq. (10).

$$\gamma = \alpha y_1 + (1 - \alpha) y_2 \tag{10}$$

Where y_1 and y_2 is the prediction result of SVM and NN, γ was the final result, and α was set to 0.5.

Suratgar *et al.* [96] proposed a NN-based solution for prediction of earthquake magnitude. They used data of Iran from the year 1970 to the year 1976 for simulation purposes. They considered the absolute value of declination and horizontal geomagnetic field in this prediction. 5 parameters of 9 days were used as input to the two-layered nonlinear auto-regressive networks with exogenous input (NARX) algorithm. NARX is a recurrent dynamic network. For error correction of the weights, the LM algorithm was used, which was modified by the Gauss-Newton method.

Rafiei and Adeli [97] predicted earthquake magnitude using neural dynamic classification algorithm and the location of it using an optimization algorithm named neural dynamic optimization of Adeli and Park (NDAP). Based on a time resolution of one week, the time lag number was calculated with a magnitude threshold of 4.5, 5.0, 5.5, 6.0, and 6.5. 8 seismicity indicators of an earthquake were calculated using

TABLE 4. A summary of used methodologies and features used in NN based researches for earthquake prediction

Ref.	Used Algorithm	Used Features
[78]	ANN	dpt., esn., mg., Tile location
[80]		Increments of b-value, Bath's law, OU's law
[81]		Seven seismicity indicators
[90]		OU's law, Dynamic GR's law, seismicity indicators
[82]		Daily maximum mg., SES
[83]		et., oid., heid., hid., it., ed.
[84]		t_e, $\bar{A_e}$, E_e, $\frac{dA_e}{dt}(dB)$, Δ_e, $\bar{t_e}$, δA_e, ρ
[85]		mg., dbeecf., timing, focal dpt.
[91]		15 channels from ocean bottom pressure gauge
[86]		Seismicity variation rates
[79]		Six seismic focus parameters from MapSIS
[92]	Embedded BPNN	Occurrence time, slip rate, epicenter, dpt.
[87]	LM with BP	b-value, cumulative frequency, energy, etc.
[32]		dpt., tos., lo., la., pressure, rc.
[88]		et., oid., heid., hid., it., ed.
[89]	LR and BP	Seismic and time varying parameters
[93]	NLFA and ANN	b-value, scde., noe., risk, sei., if.
[94]	HC and ANN	srga., wbt., La., Lo., dbt., stp.
[95]	ANN and SVM	lo., la., dpt.
[37]	BP and AdaBoost	Statistical features, Wavelet features
[96]	NARX	vgfd., hrh., gt., horizontal component, rainy rate
[97]	NDC and NDAP	t_e, $\bar{A_e}$, E_e, $\frac{dA_e}{dt}(dB)$, Δ_e, $\bar{t_e}$, δA_e, ρ
[98]	GMDHNN	Time Series data
[99]	MFOFLANN	Six seismicity indicators
[35]	GA	Geo-magnetic and solar indices
[100]	ANN with GA	b-value, accumulation frequency of eq., energy, etc.
[101]	BP and BP-GA	Moment magnitude, year
[36]	BP and PSO	re., ef., cr., b-value, los., S-value
[102]	IABC with MLP	Time series data
[103]	Meta-Learning and TL	Inter-event distance, time, magnitude
[104]	Ensemble of RBFNN	rst., b-time, ef., s-level, Vp/Vs, etc.
[38]	RBFNN	Energy, mg., b-value, etc.
[105]		Time series data
[106]		dpt., ssd., mg., la., lo., Mc
[107]		Month, day and the F-E seismic regions
[3]	PNN	
[108]		t_e, $\bar{A_e}$, E_e, $\frac{dA_e}{dt}(dB)$, Δ_e, $\bar{t_e}$, δA_e, ρ
[109]	KNN, NB, SVM, DT, ANN	Seismicity indicators, increments of b-value
[110]	DT, SVM, RF, LR	Maximum amplitude, sd., start and end time, etc.
[111]	RF, ANN, SVM	Sixty different seismic features.
[112]	DT, RF, ROT Boost, ROT Forest	51 features from GR law, and energy release
[113]	RF, NB, KNN, SVM	cscsf., onaf., slip distribution
[114]	LR, AR, DT, Rep Tree, MLP	dpt., local mg., lo., la., local time, date

Legends: Ref.– Reference; GR– Gutenberg Richter; dpt.– depth; esn.-– Earthquake Sequence Number; mg.– magnitude; et.– electron temperature; oid.– Oxygen ion density; heid.– Helium ion density; hid.– Hydrogen ion density; it.– ion temperature; ed.– electron density; dbecf.– distance between earthquake epicenter and the causative fault; et.– electron temperature; oid.– Oxygen ion density; heid.– Helium ion density; hid.– Hydrogen ion density; it.– ion temperature; ed.– electron density; scde.– spatial concentration degree of earthquakes; noe.– number of earthquakes; sei.– seismicity evolution index; if.– intensity factor; tos.– temperature of soil; rc.– Radon concentration; lo.– longitude; la.– latitude; srga.– soil Radon gas amount; wbt.– Wet bulb temperature; dbt.– dry bulb temperature; stp.– steam pressure; vgfd.– variation of geomagnetic field declination; hrh.– hourly relative humidity; gt.– ground temperature; re.– releasing energy; ef.– earthquake frequency; cr.– creep; los.– lack of shock; rst.– resistance; ssd.– seismic sequence duration; Mc– magnitude of completeness; sd.– standard deviation; cscsf.– Coulomb stress change on the surface fault; onaf.– orientation of neighboring active faults; t_e– elapsed time, $\bar{A_e}$– mean magnitude, E_e– earthquake energy, $\frac{dA_e}{dt}(dB)$– slope of magnitude (dB), Δ_e– mean square deviation, $\bar{t_e}$– mean time, δA_e– magnitude deficit, and ρ– coefficient of variation.

the earthquake catalog. They collected earthquake catalog of 1932 to 2016 for the southern Californian region and calculated features for a range of half months. Therefore, the number of feature vectors for each year was 24. The NDC network had five layers that are an input layer, feature vector representation layer, pattern layer, summation layer, and output layer. The input feature vector was transformed into a 3-dimensional feature vector twice in the second layer of the network. The transformation parameters are optimized using the NDAP algorithm. In this network, if the prediction of magnitude is larger than the threshold value, only then the location is predicted using the NDAP algorithm.

Majhi *et al.* [99] proposed a functional link ANN optimized by the moth flame optimization (MFO) algorithm to predict the magnitude of an earthquake. The FLANN has no hidden layers where some nonlinear function obtains nonlinearity. They have used regular BP, least-square optimization, Gradient descent, and LM-BP and MFO algorithm as a learning algorithm to see which one works better. These algorithms were used to find the optimal weights of the model. First, earthquake data with a magnitude greater or equal to 5.5 was selected. After that the time and date attributes were combined as one attribute. All the attributes were normalized and then expanded. Five seismicity indicators were calculated and then expanded using 7 different nonlinear functions. Finally, the model was trained with them.

Zhang and Wang [100] proposed a combination of a GA and BPNN model for earthquake prediction. They imposed a GA to obtain better convergence and to avoid local minima. They considered 17 groups of input-output data for this study, where 14 groups were used for training the model. They used 7 features that were calculated from the dataset. These features were the input to the BPNN. The architecture of the BPNN had a hidden layer with 15 neurons and an output layer with 1 neuron. There were around 136 connections; among them, 105 connections were between input and hidden, and 15 connections were between the hidden and output layer. The weights were updated using the GA. A colony composed of two hundred individuals and twenty sub-colonies was used to update the weights. The generation gap was 0.8, the intercrossing rate was 1, the mutating rate was 0.01, the max generation was 500, the rate of migration was 0.2, and the insertion rate was 0.9. After a set of selection, mutation, and intercrossing, GA erased the lower adapted chromosomes and set the optimal weight for the BPNN connection.

Tao [101] tried to predict long-term earthquakes with high magnitudes with the use of BP and BP with GA. He considered data from the Himalayan region and Nepal for this study. The data was calculated for 2150 B.C. to 2014 A.D. A three-layer feed-forward network was used where the hidden layer had 6 neurons, and the activation function was S-tangent function. The output layer used the S-algorithm activation function. The steepest descent algorithm trained the model. The weights and thresholds of the BP network were then obtained using the GA where the maximum generation was 2000, and the size of the population was 10. The mutation rates and cross probability were 0.1 and 0.5, respectively.

Li and Liu [36] proposed a hybrid algorithm of PSO and a BPNN for predicting the magnitude of an earthquake. They calculated six features from the earthquake catalog of longitude 117 degrees to 120 degrees east and latitude 22 degrees to 26 degrees east. These data were divided into 29 groups of which first 20 groups were taken for training and rest were used for testing purposes. They have used an improved PSO algorithm as it is strong in global search at first and becomes good at local search eventually because of inertia weight (ω). To improve the PSO, the nonlinear algorithm function was used as inertia weight, which made it capable of being strong in local search in initial stages as well. This PSO algorithm was used in optimizing the weights and thresholds of a three-layer BPNN to make it faster and more effective. The NN had 6 input neurons and 15 hidden layer neurons, which used the Tan sigmoid activation function.

Shah and Ghazali [102] proposed the improved artificial bee colony (IABC) algorithm for training MLP in time series earthquake data. They selected the earthquake data of California of the year 2011 for this study. This dataset contained 6000 instances, and 70% of the data were used for training the model and rest were used for testing. IABC algorithm reduced the weight areas and used 3 control parameters. In this algorithm, agents divided the food area, and the bees searched for the best weights. Every bee suggested a solution area where the best among them were selected using a greedy algorithm. In this work, the learning rate was 0.6, and the momentum was 0.5. The MLP model had 2 hidden layers and an output layer. The weights of the model were initialized randomly, and the IABC algorithm was used to find the optimized configuration.

Maya and Yu [103] proposed the use of transfer learning and meta-learning in a NN for improving the learning process to predict an earthquake. Transfer learning and meta-learning are capable of increasing the speed of convergence in the NN. They at first trained a simple MLP consisting of a hidden layer with 10 neurons and one output layer where the learning rate was 0.155. This method was proved to be inconvenient. Then they made the same prediction by using the fusion of MLP and SVM with only 100 iterations. This time, the performance was improved than the previous MLP, but the delay elements were more significant than expected. They introduced transfer learning by choosing the last weights as initial weights of the new model, and the combination of MLP, meta-learning, and transfer learning improved the performance of MLP with a less amount of time.

Liu *et al.* [104], proposed an ensemble procedure of RBFNN called CERNN for earthquake prediction. Here, the number of neurons in hidden layers and the number of training epochs were determined automatically. They have used 14 earthquake precursors as input to the network and generated training subsets using a Bootstrap algorithm. The model was built by training the RBFNN with 23 earthquakes. A group of 11 earthquakes were used for validation, and a separate group of 11 earthquakes were used for testing

the model. The RBFNN was designed using nearest neighbor clustering. They have trained the RBFNN for a certain amount of time then they checked if it produces error less than a definite value. When the error was minimum than the threshold, it is added in the CERNN.

RBFNN was successfully used in earthquake prediction researches [38], [105]–[107]. For example, Wang *et al.* [38] proposed an RBFNN to predict the earthquake magnitude. They experimented with data of south-west Yunnan province. They selected 7 different factors such as accumulation of energy, b-value, number of earthquakes, activity cycle, magnitude, and so on. as input and value of magnitude were normalized to prevent over-saturation of neurons. They proposed a model containing an input layer, a hidden layer, and an output layer. The network output was counter normalized to achieve actual prediction. For training purposes, 10 samples were used, whereas 7 samples of the earthquake were used for testing purposes. The result of the RBFNN was compared with the performance of BPNN.

Amar *et al.* [107] proposed a BPNN and an RBFNN in predicting an earthquake. The RBFNN had 3 layers where output is generated based on the linear combination of the hidden layer outputs. Each hidden layer used the RBF function as the activation function. RBFNN showed impressive generalization ability, and lengthy calculation was avoided. Parzen windows classification was used to divide the dataset into different magnitude groups. It divided the dataset into four classes. Then they trained the model with data from around the world of the year 2000 to the year 2011. The data of the year 2011 was used for testing purposes of the two models. The RBF model performed better than the vanilla BP model.

PNN was used for earthquake prediction in some studies [3], [108]. For example, Adeli *et al.* [3] predicted earthquake with a PNN, which can predict class ranges but not specific values. Therefore, this model can be used for magnitude range prediction but not for time or place of occurrence of an earthquake. The historical data of 1950 to 1990 of the southern California region was used for this study. The seismicity indicators were calculated for every 15 days of the concerned data range. They have used 8 seismicity indicators as an input vector, which are normalized with RMS values. The model consisted of one input layer, a pattern hidden layer, a summation layer, and one output layer. The pattern layer computed the Gaussian function of the Euclidean distance between the input vector and the training vector. It did not use the learning rate or predefined convergence criteria. The Parzen window classifier was used for PDF calculation and classification. Sheng-Zhong [108] proposed a PNN model for predicting the magnitude of earthquakes. At first, he calculated 8 different seismic indicators, which was given as input to the PNN model. The PNN model consisted of an input layer having 8 neurons, 2 hidden layers named pattern layer and summation layer, and an output layer. There were 7 different input classes which were based on the magnitude of earthquakes and 7 output classes according to different range of magnitude. They have used data from Northern China, and 997 feature vectors were used for training this model. The performance of each class range was evaluated separately to have a profound realization of the performance of this model.

Li *et al.* [110], used an ML approach for distinguishing the earthquakes from seismic data and predicting the arrival time of primary (*P*) wave and secondary (*S*) wave. They collected data from 16 seismological stations. The data were in SAC format. At first, they performed some data preprocessing tasks and extracted 6 features from the file header. As each data file was around 100 MB in size, data clipping was done using a trigger picker algorithm, such as recursive short time average or long time average, to capture the seismic event. The signals were also processed using a band-pass filter and detrend algorithm to remove noise. They used the autoregressive (AR) Picker algorithm for estimating the arrival time of *P* and *S* waves. Then different ML algorithms such as SVM, DT, RF, and LR were used to detect earthquakes and predicted the arrival time of *P* and *S* wave. They also calculated the epicenter using a triangulation technique.

Asim *et al.* [111] proposed a seismicity analysis for extracting features and ML algorithms for prediction of earthquakes in Cyprus where thresholding magnitudes were 3, 3.5, 4, and 4.5. They collected the data from Euro-Mediterranean Seismological Center, which was comprised of around 10000 events from the year 1924 to 2018. The aftershocks were removed from the data using de-clustering methodologies. All the earthquake events below the corresponding threshold were removed as well. Sixty different seismic features such as energy release, seismic rate, change of seismic rate, and so on were calculated from the dataset. Then ANN, RF, and SVM were used for prediction purposes. The proposed ANN was three-layered, and there were 20 neurons in the hidden layer where the weights were updated using BP. Radial kernel-based SVM and an RF comprised of 20 decision trees were also used for prediction purposes. This model could predict an earthquake for different magnitudes and period thresholds.

Asim *et al.* [112], used seismic indicators and tree-based ML algorithms for predicting an earthquake in Hindukush. The proposed model was capable of predicting earthquake whose magnitude was greater than 5.0 and which would possibly occur within 15 days. They considered the classification task as a binary classification task. At first, they calculated 51 different seismic features from seismic data of the previous 50 events. They mainly calculated Gutenberg-Richters law, energy release, and seismic rate change using different procedures that extracted the maximum possible features. DT, J48, RF with 20 decision trees and depth of 10, Rotation forest with 8 subsets, RotBoost with 10 iterations were compared for binary classification from extracted features. They employed 10-fold cross-validation scheme for evaluating the performance of the models.

Karimzadeh *et al.* [113] used ML classifiers along with Coulomb stress change, stress distribution, and fault map to predict the aftershock of the Kermanshah earthquake. They

treated this problem as a binary classification problem and predicted the location of aftershock with magnitude greater than 2.5. The slip distribution was calculated from a nonlinear and linear inversion process. The Coulomb stress change, which defines the faults and the surrounding faults caused by the earthquake, was calculated using the slip distribution, friction coefficient, and Skemton's coefficient. A binary grid map was generated from the geographical coordinates of the aftershocks and NB, KNN, SVM, and RBF classifiers predicted whether any aftershock occurred on a specific grid or not. These models simply predicted zero or one for a point for which Coulomb stress change, slip distribution, and slip distribution was represented.

Celik *et al.* [114] proposed an earthquake magnitude prediction method based on analyzing the time series and implementing ML algorithms. The date, time, depth, location, and local magnitude of 3000 earthquake events in Turkey were used for this study. The magnitude range of the earthquakes was between 1 and 5. Auto-correlation, and partial correlation of the delay were also considered along with earthquake catalog parameters. LR, DT, Additive Regression, Rap Tree, and MLP were used for comparison of prediction performance. For forecasting, Weka application was used. The longitude, latitude, depth, and magnitude were predicted by turns. Table 4 shows the summary of the publication years, used methods, and parameters for studies that involve NN-based researches.

2) Earthquake's Characteristics Studies

In this portion, the studies that are related to earthquake prediction studies but not performed earthquake prediction are discussed. Earthquakes best set of features, precursors, model behavior, explosions, forecasting error reduction, anomaly detection, and earthquake modelling related researches are presented here.

Martínez-Álvarez *et al.* [90] tried to use feature selection in predicting earthquakes. They selected the region of Chile and the Iberian Peninsula for seismic data. The basis for feature selection was information gain. First of all, they converted the problem to a prediction problem to a classification problem. Then information gain for each set of features was calculated. They have calculated 16 seismicity indicators based on GR-Law and OU's Law. Based on that, a rank was given to the samples. Then seven best features were selected and used as input to an ANN model. The model's statistical performance was calculated next using non-parametric analysis of variance, which is also called the Friedman test. This feature selection technique improved the performance of the ANN model a lot.

Okada and Kaneda [91] proposed a NN model for finding indicators of an earthquake. They collected data from the dense oceanfloor network system for earthquakes and tsunamis, which is a multi-channel observatory. Fifteen channels of ocean bottom pressure gauge were taken as input data. The data were normalized between -1 and 1. The NN has three layers with a hidden layer having three neurons and one bias. They have used gradient descent for optimization and BP for learning. For nonlinearity, the hyperbolic tangent *tanh* activation function was used. From the training data, an output curve was generated, and its trajectory was found. They assumed that this trajectory could be used as a precursor for an earthquake as it moves back to the normal position after an earthquake.

Lin *et al.* [92] tried to find out the optimal number of neurons in hidden layers of a BPNN to predict earthquakes. They used the data from Philippines to construct the model. At first, they created the initial magnitude prediction BPNN model using two hidden layers with data from the year 2000 to 2010. The best prediction was achieved using 10 neurons for hidden layers and a learning rate of 0.83. Then they used an embedded BPNN model with initial weights same as the previous model and also used the slip rate of the plate so that it can predict data in the future. It was trained with data from the year 1990 to 1999 and year 2011 to 2014. Then the initial BPNN was trained with data from the year 1990 to 2014 and learning rate 0.33, and had a similar result as the previous model. They calculated correlation coefficient, standard deviation and MSE for the predicted and the actual magnitudes on a yearly basis to evaluate the methods.

SriLakshmi and Tiwari [93] proposed a nonlinear forecasting technique and ANN-based model for earthquake prediction. They used data from the year 1960 to the year 2003 of the North-East Indian region. Data with magnitude greater or equal to 4.0 were selected for this study. Initially, they reconstructed the data into a two-dimensional phase plane to use in nonlinear forecasting. They considered each event as m-dimensional points, and a time delay was also calculated comparing with previous events. A neighborhood was chosen from the m-dimensional points for which the prediction got the smallest simplex. The optimum simplex had the minimum phase volume and vector displacement. They also proposed an ANN approach with one hidden layer and an input layer with 2 to 10 neurons. They used the LM algorithm for this analysis. The output of ANN was the frequency value of the next month. They set the number of neurons in the hidden layer to 10, learning rate to 0.01, and momentum to 0.9 after the trial and error process. They also used LR to evaluate the model where they used the output of the model and the target output.

Niksarlioglu and Kulahci [94] tried to estimate earthquake magnitude using the change of concentration of Radon gas, the relation of this gas with seismic events, and other environmental changes. An ANN and clustering were used for achieving this goal. The input to the ANN model was data related to Radon gas volume and temperature by which the magnitude of the earthquake is estimated. Next, clustering was applied using non-hierarchical clustering, which uses Euclidean distance. The seismic data from the East Anatolian fault was collected for the year 2005 to the year 2010. They recorded 69 earthquakes with magnitude greater than 3. The ANN used LM algorithm for BP operation. For the 3 clusters, this ANN model predicted the magnitude of earthquakes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

Zhao *et al.* [37] came up with an ensemble algorithm of BP and AdaBoost to differentiate between natural and explosion-based earthquakes. They used data from the China earthquake science data center, which had 656 samples. Among the samples, 138 of them were earthquakes, and the rest were explosion data. This method utilizes some weak BP classifiers to make a strong one. They have used seismic wave signals as input and calculated 27 features from them. The features were then normalized. The weak classifiers were made of only one hidden layer with three neurons, which helps in having excellent generalization capabilities. 10 weak classifiers were used to create the proposed classifier, and a comprehensive decision-maker predicted the output class based on them. The classifier was then compared with principal component analysis (PCA)-SVM and BPNN models.

Fong *et al.* [98] proposed an extension of the existing group method of data handling (GMDH) algorithm to reduce errors in forecasting rare events. The proposed method consisted of three stages. At first, they converted the time series data to multivariate format from univariate format. Then a feature selection was made to find out the relevant variables. The selected variables were the input of GMDH modeling, which was done using the models, inference and algorithms or combinational (COMBI) algorithm. Lagged time series data and residuals were also the input of the first layer of GMDH. The coefficients of each neuron were calculated using the least square regression method. The fittest nodes were the input to the next layer of the NN. This process continues until finding out the best neurons for a layer. For testing their theory, they collected earthquake data from 1973 to 2012 around the world and evaluated the performance. There were 663,852 events on which the model was built. Their proposed model worked well on the rare earthquake events.

Akhoondzadeh [35] proposed using GA to detect anomaly of total electron content (TEC), which can help in the prediction of an earthquake. TEC is the integrated number of electrons between two satellites or between a satellite and a receiver. The TEC related to the 8.0 magnitude Solomon earthquake was considered for this study. The bio-inspired GA algorithm was used, which has three major states such as crossover, mutation, and reproduction. In the crossover stage of the GA, two chromosomes were taken to create a new one. In the mutation stage, some of the gene values were changed, and in the reproduction part, the new chromosome's fitness function was calculated to verify if it is a new generation chromosome. The higher bound of TEC can be calculated using Eq. (11).

$$\mu + 1.5\sigma \tag{11}$$

Where μ is the mean and σ is the SD of the differences between the actual and predicted TEC value. Different other models were compared with this model to detect anomalies.

Chen *et al.* [105] proposed a seismic time series prediction algorithm using chaos theory and RBFNN. They used the seismic event that occurred in Guanxi as their dataset. The earthquakes with magnitude less than 0 and greater than 9 were trimmed from the dataset. At first, they calculated the parameter from a time series dataset and reconstructed phase space using them. Then they established the RBFNN. The momentum term was calculated by adding an adaptive factor into gradient descent, and a Gaussian function was used as a transfer function. The number of the neurons of the hidden layers was selected using mutual authentication, which reduced the RMSE. A linear activation function was used for the output layer. Iteration of one step was used to carry a multi-step prediction. The time series analysis of earthquakes revealed chaotic behavior which could be formulated as a deterministic process.

D'Amico *et al.* [106], compared the RBFNN and SVR as a heuristic method to overcome the contemporary anomaly problem of the delta-sigma method to forecast the number of aftershocks. They used the aftershocks which occurred in the Australian-Pacific plate boundary with main shocks having a magnitude greater or equal to 7. The ratio of differences between the observed temporal trend and the theoretical trend was used for finding anomalies in the delta-sigma method. Using this approach, if $\frac{\Delta}{\sigma} > 2.5$, then an anomaly was detected. First, the RBFNN was created using two hidden layers, the first layer having RBF transfer function and the second one having linear transfer function. The weighted input function was calculated by $|dist|$ function, and element by element product for the first and second layers, respectively. The SVR method used Vapnik ϵ-insensitive loss function with linear kernel and the libsvm optimizer. The performance of the two methods combined with the delta-sigma algorithm was tested using RMSE and Willmott's index of agreement (WIA).

Asencio-Cortés *et al.* [109] tried to determine the parameters which work best predicting an earthquake. A set of earthquake parameters were calculated from an earthquake catalog of Chile. These parameters could have different parameterizations. Based on the parameters, 5 different studies were done using a different subset of the parameters and different parameterization values. Their Sn was observed, and optimized values for them are selected. Then some popular supervised ML classifiers such as KNN, NB, SVM, C45, and ANN were used to see the classification accuracies. Different studies provided different feature sets, which means that the best set of features may vary based on the data used for building the model. But this process can help to find the best set of features.

C. DEEP LEARNING (DL)

DL is the hot topic of AI-based researches. This ML process does not need handcrafted features and can generate thousands of very sophisticated features, which is very difficult for a human to find by hand. These DL-based models use multiple hidden layers, that can be time-consuming. Because of sophisticated features, these models can face the overfitting problem. Therefore, dropout and regularization concepts are used. The process of earthquake prediction

with DL-based approaches is presented in Fig. 12. Here, the multiple hidden layers are used to find the features which are used in classification using a fully connected layer. We will categories the studies into two groups such as earthquake's characteristics studies for deep learning-based approaches, and earthquake and aftershock prediction studies for deep learning-based approaches.

1) Earthquake's Characteristics Studies

In this portion, the studies that are related to earthquake prediction studies but not performed earthquake prediction are discussed. Earthquake' SES anomaly prediction studies and earthquake trend prediction related researches are presented here. DNN is getting popular in earthquake prediction researches [39], [115]. Kanarachos *et al.* [115] proposed a hybrid algorithm comprising of the wavelet transform, DNN, and Hilbert transformation for detecting anomalies in time series data. They used this model to predict the anomaly in SES during an earthquake. They collected SES of the 18th and 19th of December 1995 after the Kozani-Grevena earthquake. These SES were used for anomaly prediction. Their transferable algorithm was used for wavelet decomposition for de-noising purposes and used the Daubechies basis function. The output from the wavelet decomposition was used as input to the DNN, which first learned hidden patterns in the data and then generalized them. The hidden layer was comprised of an ensemble of different stacked NN, where for activation, logistic sigmoid functions were used. The next layer was used to learn the temporal structure of the signal. The output of this model was then compared with the wavelet decomposition output for error calculation. The Hilbert transformation was used for the detection of anomalies in the signal. For seismic application, they used SES's long dipoles L's - I, L', L.

Kishore *et al.* [116] compared the feed-forward NN and LSTM to predict the trend of an earthquake. From the Afghanistan and Tajikistan area, 5000 samples were taken for this study. The data from Thailand, North-East India, and Lakshadweep were also collected. The feed-forward NN had 2 hidden layers with 20 and 40 neurons, each having a sigmoid activation function. The learning rule they have used was RMSprop. The LSTM model finds the dynamics of the sequence using cycles in the structure. LSTM removed the problem of vanishing effect from an RNN. The LSTM model they have used has 2 hidden layers with 40 hidden units in each LSTM cell, and BP was limited to 15 steps. A drop out layer was placed between the 2 hidden layers. They reduced the RMS loss using the Adagrad algorithm.

2) Earthquake and Aftershock Prediction Studies

The earthquake magnitude prediction-based studies are discussed in this portion.

Wang *et al.* [39] proposed a DL-based magnitude prediction model for earthquakes using the *P*-wave to give forecasting information. They have used 30,756 accelerograms from the year 1997 to the year 2019 for this study. The data were collected with the Kiban Kyoshin network for magnitude range of 4 to 9. This model did not need to adjust the hyperparameters as they were adjusted adaptively. The model took the first 2.56 seconds (28 samples) of the p-wave of an earthquake as input to the model and predicted the logarithm of the maximum displacement in the horizontal components. They have used MSE as a loss function. For each filter, the kernel size was 2, and the stride was 1. The choice activation function was a rectified linear unit, in short ReLU. For the pooling layer, max-pooling was chosen where the dropout rate was 0.5 to remove overfitting. The Adam optimizer was used along with a learning rate of 0.0001.

LSTM is good method for finding pattern in data. Therefore, researchers are using it for earthquake prediction [42], [116]. Wang *et al.* [42] proposed to use LSTM for earthquake prediction, which used spatio-temporal correlation. LSTM uses memory-cells, which helps in predicting long-term events. They have used data of China and divided the region into equal-sized sub-divisions. Earthquakes of magnitude higher than 4.5 were collected for this study from the year 1966 to the year 2016. They have used a 2-dimensional matrix to represent earthquake data having the same timestamp but of different locations to achieve spatial and temporal correlation. They used this matrix as input to the LSTM layer. The output of this layer went to the drop out layer to overcome the overfitting problem. Then it was passed to a fully connected layer whose output was passed through a softmax activation function. The error function they have used was cross-entropy, and to optimize it, RMSprop optimizer was used.

Asim *et al.* [117] tried to evaluate the performances of pattern recognition NN (PRNN), RF, RNN, and Ensemble of linear programming boost (LPBoost) algorithm in predicting earthquake greater than 5.4 before 1 month. They used earthquake data of Hindukush region from the year 1976 to the year 2013. A total of 11,137 events were used for this study. The PRNN used LM-BP for training as it works faster than the normal BP. It was made with 2 hidden layers, each having 12 neurons, and the transfer function was tan Sigmoid. The RNN algorithm could save the internal state as there were directed cycles present between the units. It was composed of 2 hidden layers with 6 and 7 neurons, respectively. The RF algorithm was an ensemble of many decision trees that were merged through Bootstrap aggregating or bagging with 50 trees. The ensemble of LPBoost maximized a margin between training data of different classes and linearly added many tree-based classifiers.

Panakkat and Adeli *et al.* [40] proposed 2 different approaches for calculating the seismic indicator and showed the effectiveness of RNN in earthquake location and time prediction. The earthquake catalog of the southern California was used for this research from which the 2 different sets of seismicity features were calculated. In the first approach, they divided the whole region into some sub-regions and recorded the earthquakes, which were also divided into some equal time intervals. From each sub-region and each in-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

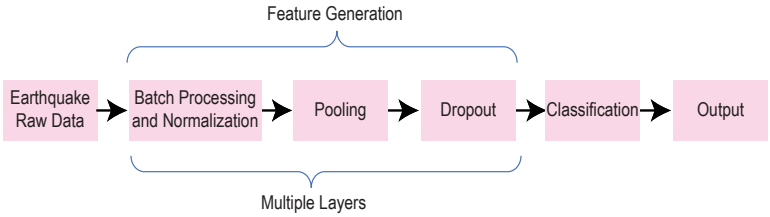

FIGURE 12. Prediction process of an earthquake using DL-based approaches. The DNN uses multiple pooling, batch processing, dropout layers to produce features. It can generate sophisticated features which are very difficult to calculate by hand. Based on these features, an earthquake is predicted.

TABLE 5. A summary of DL-based articles used in this review which include used methodologies and features

Ref.	Used Algorithm	Features
[115]	DNN	long dipoles L's - I, L' and L, SES
[39]		First 2.56s (28 samples) of P-waves
[42]	LSTM	Spatio temporal parameters
[116]		Seismicity indicators
[117]	RNN, PRNN, RF, LP Boost	$t_e, \bar{A_e}, E_e, \frac{dA_e}{dt}(dB), \Delta_e, \bar{t_e}, \delta A_e, \rho$
[40]	RNN	
[65]	RNN, BP, RBF	
[118]	DNN, GLM, RF, GBM	Seismicity indicators, increments of b-value

Legends: SES– Seismic Electric Signal; t_e– elapsed time, $\bar{A_e}$ –mean magnitude, E_e– earthquake energy, $\frac{dA_e}{dt}(dB)$– slope of magnitude (dB), Δ_e– mean square deviation, $\bar{t_e}$– mean time, δA_e– magnitude deficit, and ρ– coefficient of variation.

terval, seismicity indicators were calculated. In the second approach, the records were not divided into equal periods. Instead, it was divided according to the period between each large earthquakes and seismicity indicators. Here, 8 seismic indicators were given as input to RNN, which has 1 recurrent layer and 1 output layer. The output layer contained 2 nodes in case of location prediction and 1 node in case of time prediction. LM training algorithm was used for minimizing the MSE between the desired output and network output. The number of iterations was set to 1000, and the goal MSE was set to 0.001 as convergence criteria. The proposed methodology predicted time and location with an error of 56 days and 15.7 miles when the threshold of the earthquake was 6.5.

Panakkat and Adeli [65] have introduced 8 seismicity indicators and compared RNN, LM-BPNN, and RBFNN in predicting an earthquake. The best structures for each of the model were determined. They used Southern California and San-Francisco Bay region's earthquake catalog for this study. They provided an in-depth description of the calculation of seismicity indicators in this research. The best BP network had 2 hidden layers, each having 8 nodes. The RNN had 2 hidden layers. One of the hidden layers was a general dense layer having 8 nodes, and another hidden layer was a recurrent layer with 4 nodes, which used tan Sigmoid activation function. For the south Californian region, the RBFNN used 1 hidden RBF layer with 8 nodes. An architecture of 2 hidden layers with 8 nodes each were used for the San Francisco Bay region. The performance of these algorithms was then compared based on the evaluation metrics.

Asencio–Cortés *et al.* [118] proposed a cloud-based big data analysis of earthquake magnitude prediction within 7 days of occurrence, using 5 regression algorithms. They have considered DL, generalized linear models (GLM), RF, gradient boosting machines, and stacking ensemble of these algorithms for this purpose. They collected about 1 GigaByte of earthquake catalog of California and divided the data into different cells of size 0.5 longitude $\times$ 0.5 latitude. From there, they selected 27 cells where the number of events greater than magnitude 5 was more than 500. Then they calculated 16 features from them. Amazon web services was used as a cloud-based platform for big data analysis. The cells were stored in tables using Amazon redshift. For the implementation of the 5 regressors, the H2O library was used. The summary of the methods and features used in DL-based researches are shown in Table 5.

V. EVALUATION METRICS AND PERFORMANCE CATEGORIZATION

A. POPULAR EVALUATION METRICS

When a model is built, the next important thing is to check its performance. Because a low performing model is of no use and can lead to some unfavorable conditions. Here, we will briefly describe some of the most used evaluation techniques in the reviewed articles. Initially, we will see the basic concept of true positive, true negative, false positive, and false negative values. Afterwards, we will explore some of the complex evaluation metrics.

a: True Positive Values (TP)

In the prediction of an earthquake, the number of cases in which the model predicted earthquake matches the actual recorded earthquakes are called TP.

b: True Negative Values (TN)

In the number of cases, when the model predicts no earthquake, and no earthquake happened indeed is called the TN.

c: False Positive Values (FP)

This value states how many times the model predicted earthquake but did not happen in real-life.

d: False Negative Values (FN)

The number of times the model did not predict an earthquake, but in the real case, there was an earthquake, which is called FN.

From these metrics, we can calculate complex matrix such as Sn, Sp, P_0, P_1, $R-score$, and Matthews correlation coefficient (MCC). These metrics are described below:

e: Sensitivity (Sn)

This indicates the positive predictions that are predicted correctly by a model. Mathematically, Sn can be calculated by Eq. (12).

$$Sn = \frac{TP}{TP + FN} \tag{12}$$

f: Specificity (Sp)

This metric indicates the negative values that were predicted correctly by the model. Mathematically, Sp can be calculated by Eq. (13).

$$Sp = \frac{TN}{TN + FP} \tag{13}$$

g: Positive Predictive Value (P_1)

This denotes the proportion of the correctly predicted positive values out of all the positive predicted values. The mathematical equation of P_1 is calculated by Eq. (14).

$$P_1 = \frac{TP}{TP + FP} \tag{14}$$

h: Negative Predictive Value (P_0)

The portion of the correctly predicted negative values out of all the negative predicted values are called P_0. This can be calculated by Eq. (15).

$$P_0 = \frac{TN}{TN + FN} \tag{15}$$

i: False Alarm Rate

It is the probability of detection of false values. It is denoted as FAR and calculated by Eq. (16).

$$FAR = 1 - P_1 \tag{16}$$

j: Accuracy, R-score, and Matthews Correlation Coefficients

Accuracy ($\mathcal{A}$), R-score ($\mathcal{R}$), and Matthews correlation coefficient ($\mathcal{M}$) are obtained from Eqs. (17) - (19).

$$\mathcal{A} = \frac{TP + TN}{TP + TN + FN + FP} \tag{17}$$

$$\mathcal{R} = \frac{(TP \times TN) - (FP \times FN)}{(TP + FN) \times (TN + FP)} \tag{18}$$

and

$$\mathcal{M} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{[TP + FP][TP + FN][TN + FP][TN + FN]}} \tag{19}$$

k: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error ($RMSE$)

The measure of the closeness of the predicted value to the actual value is called MAE. The mean of the square of the residue of the predicted and the actual value is called the MSE value. They are obtained from Eqs. (20) and (21), respectively.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |V_{predicted_i} - V_{actual_i}| \tag{20}$$

and

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (V_{predicted_i} - V_{actual_i})^2 \tag{21}$$

The $RMSE$ is defined by Eq. (22).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (V_{predicted_i} - V_{actual_i})^2} \tag{22}$$

l: Absolute Error (AE) and Relative Error (RE)

AE is defined mathematically by Eq. (23).

$$AE = |V_{predicted} - V_{actual}| \tag{23}$$

And the relative error (RE) is defined by Eq. (24).

$$RE = \frac{V_{predicted} - V_{actual}}{V_{actual}} \tag{24}$$

These are the common evaluation metrics used by most of the researches reviewed in this paper. We can compare the studies based on these metrics.

B. CATEGORIZATION OF PERFORMANCE

Here, we will introduce a new performance measure for the different methods based on their relative accuracy, relative RMSE, and relative average error. The formula for calculating relative performance is mathematically represented as Eq. (25),

$$new_{value} = \frac{(Value_{old} - Min_{old})(Max_{new} - Min_{new})}{(Max_{old} - Min_{old}) + Min_{new}} \tag{25}$$

where $Value_{old}$ is the value we want to convert to the new scale, Min_{old} is the minimum value of the previous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

performance, Max_{old} is the maximum value of the previous performance, Min_{new} is the minimum range of the relative performance, and Max_{new} is the maximum range of the relative performance. For all cases, Min_{new} is 0% and Max_{new} is 100%. For the relative accuracy, if the new_{value} is between 0% and 25%, then it is considered low performance, and if the new_{value} is between 75% and 100%, then it is considered as high performance. The rest are considered as medium performing methods. This range division is mathematically represented in Eq. (26), where $\mathcal{P}$ is the relative performance class.

$$\mathcal{P} = \begin{cases} high, & \text{if } 75 < new_{value} \leq 100 \\ medium, & \text{if } 25 < new_{value} \leq 75 \\ low, & \text{otherwise} \end{cases} \tag{26}$$

For RMSE and average error, when the new_{value} is between 0% and 25%, it is considered high performance, and it is regarded as low performance when the new_{value} is between 75% and 100%. The rest are considered as medium performance. The range classification is presented mathematically in Eq. (27), where $\mathcal{P}$ is the relative performance class.

$$\mathcal{P} = \begin{cases} high, & \text{if } 0 < new_{value} \leq 25 \\ medium, & \text{if } 25 < new_{value} \leq 75 \\ low, & \text{otherwise} \end{cases} \tag{27}$$

VI. PERFORMANCE ANALYSIS AND DISCUSSION

A. RULE-BASED APPROACHES

Bofeng and Yue [44] used the FUMCE model to increase the understandability of an earthquake prediction expert system by giving different levels of explanations to a user who has a different knowledge level. They categorized the users as a novice, common user, advanced user, domain expert, and knowledge engineer so that an expert system can give an explanation of the earthquake prediction based on their ability to understand.

Zhong and Zhang [45] tried to predict earthquakes which are formed because of reservoirs using fuzzy logic. They considered the "Three George" reservoir and divided it into 1300 units. They divided the magnitude scale of earthquakes into 4 classes as v_1, v_2, v_3, and v_4 that represents micro, sensible, devastating and strong earthquakes respectively. They found that their study area produced the biggest size of membership function for the class v_3, which was 3.155. Therefore, they expected a devastating earthquake in that area.

Konstantaras *et al.* [46] used NFS to detect anomalies in electrotelluric potentials, which can be used as a precursor of an earthquake. They found that the ratio of their recorded magnetic field and residue of predicted and recorded magnetic field varied from 57.14 dB to 75.28 dB. For the electric field, this value varied from 41.86 to 52.23 dB.

Mirrashid *et al.* [47], in his ANFIS model, tried to predict the seismic moment of the next earthquake. They found that there is a correlation between the origin of an earthquake and the cumulative seismic moment. The R^2 value was 98%, and they achieved an MAE of 0.15%.

Rahmat *et al.* [48] tried to predict b-value using ANFIS and ELM algorithm. For training, the ANFIS achieved RMSE of 0.003, and ELM achieved RMSE of 6.33×10^{-7}. During testing, the ELM algorithm performed similarly, but the ANFIS produce RMSE of 1.507, which led to an inferior success rate of 21.84%. Table 6 shows the algorithms, outcomes, used dataset, and the evaluation metrics by the researches involving rule-based methods.

Iqram and Qamar [49] used predicate logic to predict earthquakes before 1/2 hour, 1 hour, 2 hours, 6 hours, and 12 hours. They found that their system can predict all the earthquakes within 12 hours margin.

The NFS of Shodiq *et al.* [50] revealed the relation between time intervals and large earthquakes ($\geq$ 5.9). They tested 11 earthquake instances and successfully predicted 8 of them. The other 3 events happened 12 days after, 5 months before, and 44 days after the original prediction.

Dehbozorgi and Farokhi *et al.* [51], in his NFM, tried to predict earthquake 5 minutes before its occurrence and used feature selection to see if the accuracy increases. They compared the model with an MLP model. Without feature selection, the NFM achieved 82.85% accuracy, where the MLP model was 71.42% accurate. After the feature selection, the accuracy reduced by nearly 2%, but the network complexity reduced a lot.

Zeng *et al.* [52] used an ANFIS inference model and compared it with a fitting model for prediction of epicentral intensity. Their model performed better than the fitting model with a mean of error of 0.36, a standard deviation of 0.47, and a correlation coefficient of 0.9362.

Andalib *et al.* [53] had the goal of reproducing the performance of human experts for earthquake prediction. Their model produced 36 false alarms for the 147 tested cases, which had a magnitude of 5.5 and above. For earthquakes greater than magnitude 6, this model false alarm was 40% as there was a lack of data for higher magnitudes.

Shodiq *et al.* [54], used ANFIS for spatial analysis of magnitude distribution of aftershocks and found that the non-occurrence was able to be predicted by the model with 70% accuracy. But the prediction of the occurrence of the earthquake was 60% accurate.

Kamath and Kamat [55] achieved RMSE of 0.059 for training and 0.408 for testing in predicting the magnitude of an earthquake using the ANFIS model.

Pandit and Biswal [56] used ANFIS with SC to achieve correlation factor (R^2) of 0.8699 for forecasting earthquakes.

Mirrashid [57] used ANFIS with fuzzy means clustering to achieve RMSE of 0.172 for earthquakes with magnitude greater or equals to 5.5. With 12 clusters and 12 rules, he achieved an R^2 value of 0.95 and an MAE of 0.150. Bahrami and Shafiee [58] forecasted an earthquake with an MSE of 0.0395 using their ANFIS model, which had seven neurons.

For the rule-based approaches, accuracy is converted to relative performance with Min_{old} equals to 70% and

TABLE 6. A summary of used algorithms, datasets, evaluation metrics, and obtained outcomes of rule-based earthquake prediction researches

Ref.	Used Algorithm	Dataset	Evaluation Metrics	Performance
[44]	FUMCE			Classified a user into a defined group
[45]	HDFT		Size of membership degree	Largest membership degree was 0.3155
[49]	Rule-based ES	USGS	Acc.	Acc. was 100% (12 hrs. Before eq.)
[46]	NFM		REF/(PEF - REF)	REF/(PEF - REF) varied between 41.86 and 52.23
[50]		GI-NOA catalog	Prediction plot	Provide 8 correct predictions
[51]	NFC	IRIS	Acc.	Acc. was 82.8571%
[52]	ANFIS		Mean and sd. of error	Mean of error was 0.3614
[53]		ZAC	Prediction plot, false alarm	36 false alarms out of the 147 testing data
[54]		BMKG and USGS	Acc., P_0, P_1, Sn, Sp	Average Acc. was 70%
[55]		EMSC	RMSE	RMSE was 0.108
[56]		SMVDC	Residual	SC was most effective
[47]		IIEES	RMSE, MAE, R^2	RMSE was 0.15%
[57]			R^2, MAE, RMSE	RMSE was 0.172
[58]	GFCV		MSE	MSE was 0.0395
[48]	ANFIS and ELM	ISC's catalog	RMSE, MAPE	RMSE for ELM was 6.25×10^{-7}

Legends: Ref.- References; Acc.– accuracy; ES– Expert System REF– recorded electric field; PEF– predicted electric field; RMSE– root mean square error; MSE– mean squared error; USGS– United States Geological Survey; GI-NOA– Geodynamic Institute of the National Observatory of Athens; IRIS– Incorporated Research Institutes for Seismology; IIEES– International Institute of Earthquake Engineering and Seismology; ZAC– Zagros Earthquake Catalog; BMKG– Indonesian Agency for meteorological, Climatological, and Geophysics; EMSC– Euro-Mediterranean Seismological Center; SMVDC– Strong Motion Virtual Data Center; MAE– Mean Absolute Error; sd.– standard deviation; P_0– negative predictive value; P_1– positive predictive value; Sn– sensitivity; Sp– Specificity; MAPE– Mean Absolute Percentage Error.

Max_{old} equals to 100%. For RMSE conversion, Min_{old} is 6.25×10^{-7} and Max_{old} is 0.601. The relative performance classes are evaluated using Eqs. (26) and (27).

a: The Works Reporting the High Performance for Rule-Based Approaches

The rule-based expert system method achieved relative accuracy of 100% [49]. The ANFIS based methods [55] and [47] provided relative RMSE of 17.96% and 24.95%. The ANFIS with ELM method [58] achieved relative RMSE of 19.8% and GFCV method [48] achieved relative RMSE of 0%. These methods are considered as high performing methods according to the criteria described previously.

b: The Works Reporting the Medium Performance for Rule-Based Approaches

The NFC-based method [51] got relative accuracy of 42.86% and the ANFIS-based method [57] provided relative RMSE of 28.62%. These two methods are considered medium performing methods.

c: The Works Reporting the Low Performance for Rule-Based Approaches

The ANFIS-based method [54] achieved relative accuracy of 0% and the ANFIS-based method [52] had relative RMSE of 0%. These two models are reported as low performing methods. We have presented the categorization of the studies based on relative performance in Fig. 15.

B. SHALLOW MACHINE LEARNING APPROACHES

Jiang *et al.* [59], with his SVM based model, predicted the most significant magnitude in the following year with training accuracy of 69%. With ± 0.5 error, it increased to 75%. The testing MSE they achieved was 0.19.

Astuti *et al.* [60] used the SVD for feature extraction and SVM for classification of location, time, and magnitude of an earthquake. For location and magnitude, this method achieved 77% and 67.67% accuracy on average, respectively. It could predict earthquakes 2 to 7 days ahead of its occurrence.

Hajikhondaverdikhan *et al.* [61] proposed a PSO based SVR model, which achieved 96% accuracy for magnitude prediction and 78% accuracy for predicting the number of earthquakes in the next month. For magnitude prediction, the MSE was 0.0015, and 0.017 for the number of earthquake prediction. This method worked better than the regular MLP method.

Asim *et al.* [62], used the SVR with the hybrid PSONN for prediction of an earthquake and achieved an accuracy of 82.7%, 84.9%, and 90.6% for Hindukush, Chile, and Sothern California region respectively.

Li and Kang [63] have used polynomial regression with KNN for the prediction of aftershocks and achieved an RE of 22.756, which is 6.012% better than that of the conventional KNN, and 7.75% better than the distance weighted KNN.

Prasad *et al.* [64] tried to predict the magnitude of the seismic signal with Haar wavelet transformation and ML classifiers. For classification, they used J48 algorithm, RF, REP tree, LMT, NB, and BP. Among these the RF algorithm was 100% successful in detecting the magnitude.

Sikder and Munakata [66] used rough set theory and DT to predict an earthquake. The rough set theory achieved 88.39% accuracy, and the computed receiver operating characteristics (ROC) curve was 0.60. The DT performed better with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

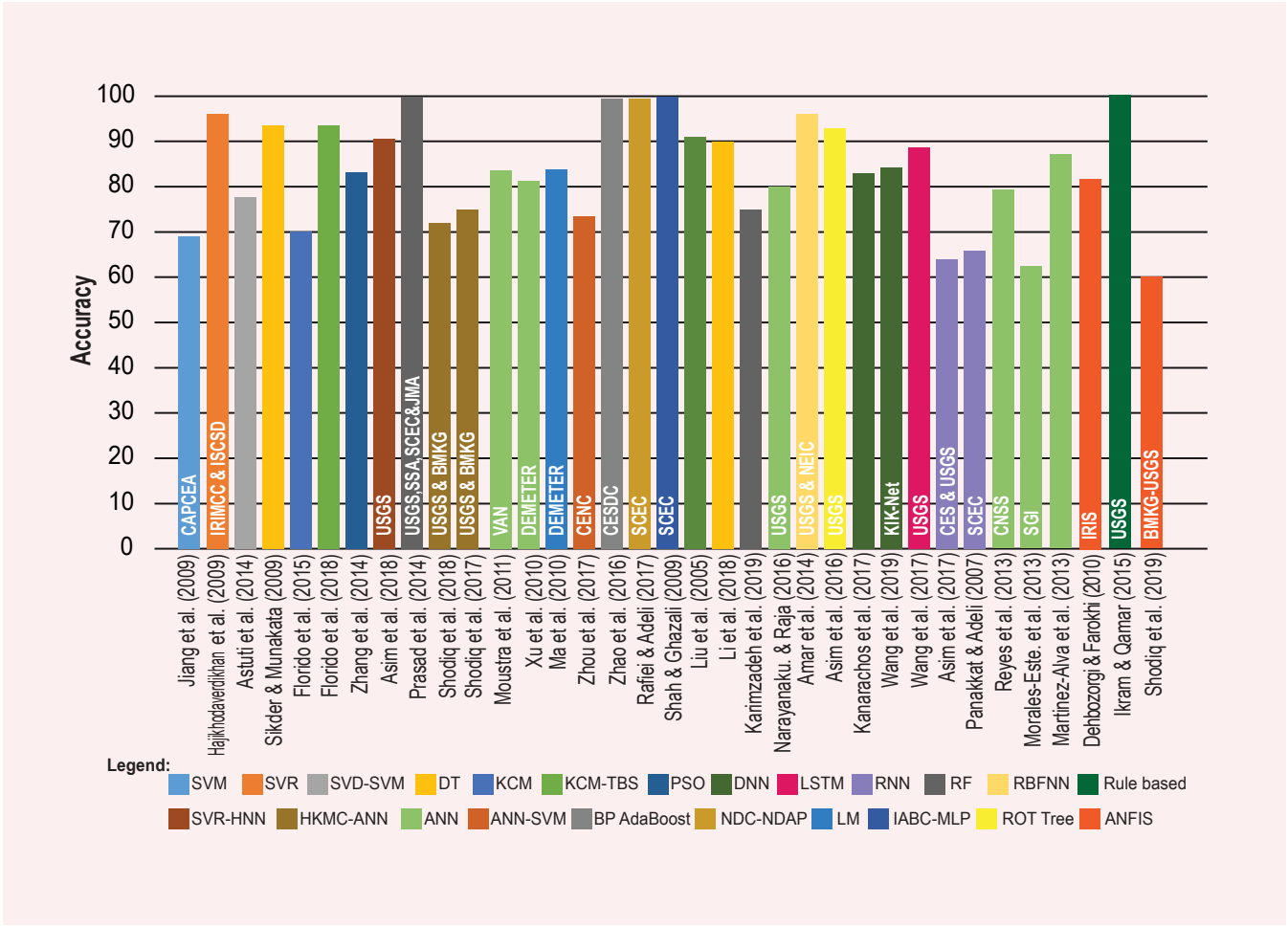

FIGURE 13. Performance comparison of the articles based on accuracy. The different algorithms are presented using different colors. The height of each bar represents its accuracy. The dataset used by the researches is written over the bar. The rule-based expert system was able to achieve 100% accuracy in earthquake prediction within 12 days. The NDC-NDAP, BP-AdaBoost, and IABC-MLP algorithms produced accuracy more than 99% for predicting earthquakes using their datasets. The RF algorithm was 100% successful for detecting earthquakes. All the algorithms are at least 60% successful in predicting earthquakes.

93.55% accuracy and computed ROC of 0.626.

Marisa *et al.* [67] predicted the probability of an earthquake with Poisson HMM. The model with 3 hidden states achieved Akaike information criteria of 763.074, which was smaller than other models. Based on this model, they predicted the number of earthquakes in 1, 2, 10, 30, and 40 years range. Till 10 years range, the model performed well, which was validated by the chi-squared test.

Florido *et al.* [68] have used K-means clustering for mainshock prediction and achieved 70% accuracy on average. The Sp for Talca, Pichilemu, Santiago, and Valparaiso was 0.82, 0.86, 0.87, and 0.89, respectively. They have used the tree-based method along with K-means clustering [69] and achieved an average accuracy of 91% for the Iberian Peninsula, 88% in Chile, and 83.70% in Japan.

Shodiq *et al.* [70], used Hierarchical K-means clustering to find the clustering of earthquake affected zones. The centroid clustering method performed best with the sum squared error of 0.98347 and took only 6.11 seconds. They predicted that there would be no earthquake till 2020 greater than magnitude 7. They performed a spatial analysis of the magnitude distribution and achieved Sp of 100% for earthquakes of magnitude 5.5 to 6 [72]. For earthquakes more significant than 6, the Sp reduced to 84%. They used ANN with Hierarchical K-means clustering to predict an earthquake in the next 5 days and achieved an Sp of 92.73% for a magnitude range of 5.5 to 6 [71]. For earthquakes with a magnitude greater than 6, the Sp was 99.16%.

Mejia *et al.* [73] used Debrovolsky based clustering to create a probability distribution plot of an earthquake. With a distance of 10 kilometers, an angle of 10 degrees, and a time lag of 10 days, they were able to give correct predictions for testing samples.

Tan and Cai [74] proposed a method based on AHC and SVM and achieved an accuracy of 75% for positive samples and 68% for negative samples. The achieved G-value was 0.67.

Lu *et al.* [75] have used GA based clustering for predicting the time and location of aftershocks. They created a weighted graph where the correlated aftershocks were clustered to-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

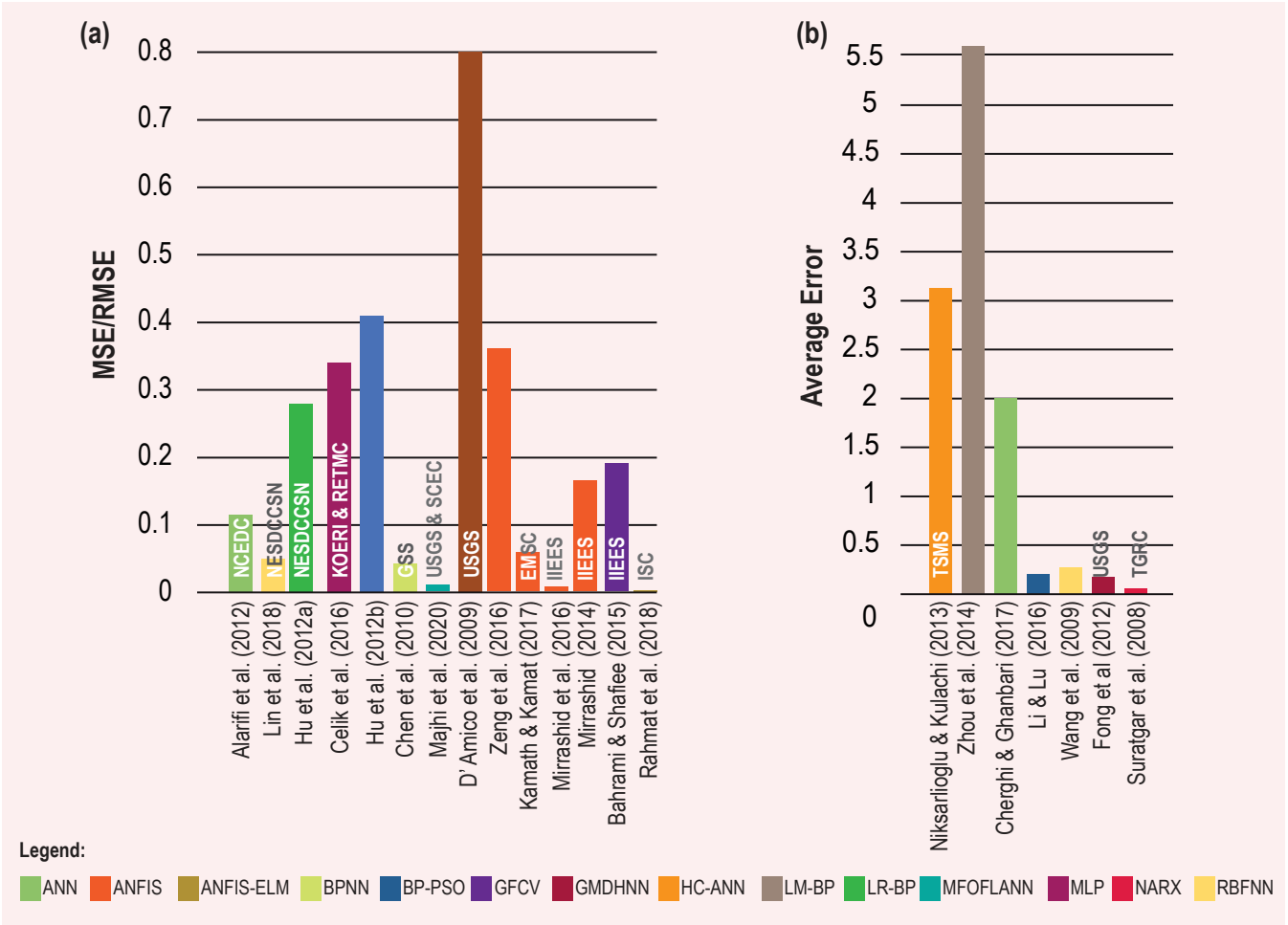

FIGURE 14. (a) Performance comparison of earthquake prediction methods based on MSE and RMSE. The model with the lowest MSE or RMSE values is mostly appreciated. The different techniques are represented with different colored bars, and their height represents their value. The respective datasets are mentioned over the bar. The GMDHNN algorithm produced the highest amount of error. The ANFIS-ELM method performed best, and the bio-inspired algorithm MFOFLANN also produced excellent results. (b) Performance comparison of earthquake prediction methods based on average error. The LM-BP method produced the worst performance, with an average error of 5.5%. The NARX algorithm produced the least average error.

gether.

Shao *et al.* [76], used ACC and compared it with the K-means clustering. The average distance the K-means clustering moved was 44.267, and for the ACC model, it was 41.488.

Zhang *et al.* [77] used PSO based clustering, and it achieved 83.3% accuracy. The distance it covered was the same as the ACC method proposed by Shao *et al.* In Table 7, the summary of the algorithms, outcomes, used dataset, and the evaluation metrics used by the classical ML-based researches are shown.

For the classical shallow ML methods, Min_{old} is equal to 69% and Max_{old} is equal to 100% for relative accuracy calculation. Based on this range, the relative performance conversion is done using Eq. (25). The relative performance classes are evaluated using Eq. (26).

a: The Works Reporting the High Performance for Classical Machine Learning-based Approaches

The SVR-based method [61] was a high performing method with a relative accuracy of 87.9%. The HWT-based method [64] for the feature transformation was found to be successful as the relative accuracy was 100%. The DT-based method [66] and the KMC-TBS method [69] were also high performing as their relative accuracy were 79.03% and 79.32% respectively.

b: The Works reporting the Medium Performance for Classical Machine Learning-based Approaches

The bio-inspired PSO-based clustering method [77] achieved relative accuracy of 46.77%, which makes it a medium performing method. The SVM-SVD method [60] produced relative accuracy of 29.32%. Therefore, it is a medium performing model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

c: The Works reporting the Low Performance for Classical Machine Learning-based Approaches

The SVM based method [59] low performing as the relative accuracy of these models is 0%. The KMC method [68] and the HKMC with ANN-based methods [71], [72] were also low performings with a relative accuracy of 3.23%, 9.68% and 19.35% respectively. These categorizations are depicted in Fig. 15.

Alarifi *et al.* [78] proposed an ANN-based magnitude prediction model of the earthquake which used Tan Sigmoid as an activation function. The model achieved MSE of 0.1153 and MAE of 0.26371. It works at least 32% better than other configurations of this model. Rayes *et al.* [80], used ANN in the Chile region and found the Sp of 88.5%, 77.8%, 97.9%, and 93.9% for Talca, Pichilemu, Santiago, and Valparaiso respectively. The average Sn is a bit small, which was 40.9%. Morales-Esteban *et al.* [81], tried to predict earthquakes within 7 days that are greater than some threshold. For the Alboran sea region, the training and testing specificities were 77.1% and 25%. For Azores-Gilbraltar fault, the training Sp was 80.3%, and the testing Sp was 92.8%. Moustra *et al.* [82] used ANN to find the magnitude of the next day using SES. Using time-series data, they achieved an accuracy of 80.55%. When they used the NN recursively to generate missing SES, that model achieved 84.01% accuracy for magnitude prediction and 92.96% accuracy for time lag prediction. Xu *et al.* [83] used DEMETER satellite data and predicted 95 positive earthquake instances of the 117 instances correctly. It could predict only 57.5% non-seismic events with their ANN-based method. Narayanakumar and Raja [84] provided an earthquake magnitude prediction model using BPNN, which predicted small earthquakes with 66.66% accuracy. The moderate earthquakes were predicted with 75% accuracy. Since there was a minimal amount of training samples for earthquakes with a magnitude greater than 5.8, the model could not predict major earthquakes. Cheraghi and Ghanbari [85] have used ANN to predict the time and magnitude of an earthquake. For magnitude prediction, the average error was 0.5%, and the maximum error was 3.5%. They could predict the timing of an earthquake with 10 days of the error. Xie *et al.* [86] predicted the actual time of the earthquake within 2 months range using BPNN. At 221 epochs, they managed to reduce MSE at the level of 10^{-3}. Hu *et al.* [79] predicted the future magnitude of 6 months using the LR-BPNN method. The LR alone can achieve RMSE of ± 0.78 ML, and the integrated LR-BPNN model can achieve RMSE of ± 0.41 ML.

Zhou *et al.* [87] compared the traditional BP model with the BP-LM model and found that the BP-LM model works better than BP. This model achieved an error level of 0.0006, and the traditional BP model reached the error level of 0.4159. This model showed a 5.6% error for testing samples. Kulachi *et al.* [32] used the LM-BPNN model for prediction of the magnitude of an earthquake. The AE they achieved was 2.3%, and the RE was between 0% and 12%. They used factor analysis and showed that 80.18% of the data were covered by four factors only. Ma *et al.* [88] have used an LM-BPNN model on the DEMETER satellite data. This model correctly predicted 78 out of 93 seismic samples achieving 83.9% accuracy. For non-seismic events, the accuracy falls to 46.6%.

Hu *et al.* [89], compared an LR and LR-BPNN model for magnitude prediction. The training RMSE for the LR model was ± 0.66, and testing was ± 0.62. With the combined LR-BPNN model, they achieved training RMSE of ± 0.22, and for testing, it was ± 0.28.

Zhou *et al.* [95] used an ANN-SVM model for earthquake magnitude prediction. They found that if the error does not exceed ± 0.5, then the SVM model can predict 9 out of 15, and ANN could predict 10 out of 15 testing samples. The combination of these two could predict 11 out of 15 samples with 73.33% accuracy.

Suratgar *et al.* [96] presented a simulation-based study of earthquake magnitude prediction. The norm of error they achieved was 0.047 to 0.067.

Rafiei and Adeli [97] proposed an earthquake magnitude prediction model using the NDAP-NDC model. This model achieved an accuracy of 99.4% and an R-score of 0.94. They also compared it with the EPNN model, which achieved an accuracy of 99.3%.

Majhi *et al.* [99] used their MFOFLANN model for earthquake magnitude prediction. This model achieved an RMSE of 0.0565. They have compared it with the IPSO-BPNN model, which achieved similar RMSE, which is 0.0590.

Zhang and Wang [100] predicted the probability of earthquakes using the BP-GA method. The maximum averaged RE was 0.1585, and its minimum value was 0.0632. For normal BP, it varied between 0.2495 and 0.5330.

Tao [101] proposed a hybrid algorithm of BPNN and GA for predicting earthquakes with a magnitude greater than 6.5. For the Himalayan region, the error was -0.06 to +0.05. Using only BPNN, it was -0.0062 to -0.0028.

Li and Lu [36] have used the IPSO-BPNN method for the magnitude prediction of an earthquake. This model achieved AE between 0.021 and 0.375. Shah and Ghazali [102] proposed an IABC-MLP model for magnitude prediction, which achieved an accuracy of 99.89%. The MSE varied between 0.001702 and 0.001910.

Maya and Yu [103] proposed a model for short-term earthquake prediction with meta-learning and transfer learning. With MLP-meta-learning, it achieved MSE of 0 to 0.09. With the MLP-meta learning-transfer learning model, the MSE was between 0 and 0.07.

Lui *et al.* [104] predicted the magnitude of an earthquake using an ensemble of RBFNN. When the magnitude difference is not more than 0.5, this model achieved an accuracy of 91%. Wang *et al.* [38] also used RBFNN for the magnitude prediction of an earthquake. The maximum prediction error was 0.8152, and the average prediction error was 0.2722. Amar *et al.* [107] predicted the magnitude of an earthquake before 1 day of its occurrence using RBFNN. They achieved

TABLE 7. A summary of used algorithms, datasets, evaluation metrics, and obtained outcomes of classical ML and clustering-based earthquakes prediction studies

Ref.	Used Algorithm	Dataset	Evaluation Metrics	Performance
[59]	SVM	CAPCEA	Acc.	Acc. was 69%
[60]	SVD and SVM	earthquakeprediction.gr data		Acc. was 77.78% (location estimation)
[61]	SVR	IRIMCC and ISCSD	Acc., R-score, MSE, RMSE	Acc. was 96% (mean magnitude)
[62]	SVR and HNN	USGS	Acc., P_0, P_1, Sn, Sp	Acc. was 90.6% (South California)
[63]	PR-KNN	CSNDMC database	AE, AAE, RE, ARE	Average RE was 8.317
[64]	HWT with ML algorithms	USGS, SSA, SCEC, and JMA	Acc., Recall, Precision	Detection Acc. of RF was 100%
[66]	RST and DT (C4.5)		Acc., Sn., Sp., F-measure	Acc. was 93.5% (DT)
[67]	PHMM	USGS	Chi-squared test	The predicted and actual values were similar
[68]	KMC		Sn, Sp	Acc. was 70%
[69]	KMC and TBS		Acc., P_0, P_1, Sn, Sp	Acc. was 93.59% (Chile)
[71]	HKMC and ANN	BMKG and USGS		Acc. was between 56% and 72% ($M \geq 6$)
[72]				Acc. was 75%
[70]	HKMC	BMKG	Probability	Predicted no eq. with $M > 7$ within 10 years
[73]	Dobrovolsky-based Clustering	PHIVOLCS	Prediction validation	Clustered the zones successfully
[74]	AHC Over Sampling	CEDC	G-value	Improved the precision and prediction hit rate
[75]	GA based Clustering			Clustered eq. prone areas
[76]	ACC algorithm		Average distance, time	Average distance covered was 41.488
[77]	PSO Clustering Algorithm		Acc., Average distance, time	Acc. was 83.3%

Legends: Ref.- Reference; TBS– Tree-based search; Acc.– accuracy; eq.– earthquake; CAPCEA– Center for Analysis and Prediction of China Earthquake Administration; IRIMCC– Islamic Republic of Iran Meteorology Center for Climatic Data; ISCSD– Iranian Seismological Center for Seismic Data; USGS– United States Geological Survey; CSNDMC– China Seismic National Data Management Center; SSA– Seismological Society of America; SCEC– South California Earthquake Data Center; JMA– Japan Meteorological Agency; BMKG– Indonesian Agency for meteorological, Climatological, and Geophysics; PHIVOLCS– Philippine Institute of Volcanology and Seismology; CEDC– China Earthquake Data Center; MSE– mean squared error; RMSE– root mean square; P_0– negative predictive value; P_1– positive predictive value; Sn– Sensitivity; Sp– Specificity; MCC– Matthews correlation coefficient; AE– absolute error; AAE– average absolute error; RE– relative error; ARE– average relative error.

an average R-score of 0.75. For the prediction of 7 days, this model achieved an average R-score of 0.88.

Adeli and Panakkat [3] introduced PNN for earthquake magnitude prediction. For magnitude 4.5 to 6, the average R-score was 0.68, and the average POD score was 0.84. Sheng-Zhong [108] also used PNN for earthquake magnitude prediction. For an earthquake magnitude of 3.5 to 6 in the Chinese region, the average R-score was 0.59, and the average POD score was 0.69.

Li *et al.* [110] compared RF, SVM, and DT algorithms for prediction of earthquake and arrival time of P-wave and S-wave. The accuracy of RF, SVM, and DT models were 85%, 83%, and 88%, respectively. Asim *et al.* [111] compared different ML algorithms for the magnitude of earthquake prediction within 15 days. For a magnitude range of 3 to 4, RF performed best and achieved an MCC of 0.81. For 4 to 4.5 magnitude range, SVM achieved MCC of 0.86, and for the earthquake over 4.5 magnitude range, RF performed best with MCC of 0.724. They also compared the ML algorithms for short-term earthquake prediction [112]. The rotation forest algorithm worked best with an F-measure of 0.928. Karimzadeh *et al.* [113] tried to predict the location of earthquakes with different ML algorithms. Without the fault information and with information of fault 1, NB algorithm performed best with 78% accuracy. With the information of fault 2, the RF algorithm achieved the best accuracy of 75%. Celik *et al.* [114] compared LR additive regression, REP tree, MLP, and DT for predicting different aspects of earthquakes. For longitude and latitude prediction, LR performed best with MSE of 0.84 and 0.96. For depth and magnitude prediction MLP performed best with MSE of 1.88, and 0.34. Table 8 shows the algorithms, outcomes, used dataset, and the evaluation metrics by the NN-based researches.

Martinez-Alvarez *et al.* [90] used feature selection with ANN to find out if there would be an earthquake greater than a threshold magnitude. This model achieved Sp of 81.08%, 97.22%, 96.77%, 94.62%, 85%, and 91.30% for Talca, Santiago, Valparaiso, Pichilemu, Alboran sea, and western Azores-Gilbraltar fault region respectively.

Okada and Kaneda [91], using ANN, learned a waveform which replicates waveform of ocean bottom pressure gage. It exhibits a cyclic trajectory which returns to its initial position after an earthquake. They said that this could be used in predicting aftershocks.

Lin *et al.* [92] first figured out the best number of neurons in the hidden layer for an ANN model and used their EEMPBPNN model for magnitude prediction. The average standard deviation was 0.21, and the MSE varied from 0.01 to 0.09.

Lakshmi and Tiwari [93] tried to evaluate the dynamics of an earthquake in the northeast Indian region. They have used nonlinear forecasting and ANN. The highest R-value they have found is 0.4014 for nonlinear forecasting and 0.520 for ANN. Because of this low value, they evaluated NEI as a chaotic plane.

Niksarlioglu and Kulachi [94] used ANN and clustering for the magnitude prediction of earthquakes and achieved RE between 0% to 6.25%. They divided the data into three clusters.

Zhao *et al.* [37] used different hybrid algorithms to differentiate earthquakes and explosions. The BPNN model achieved 98.21% accuracy, and PCA-SVM achieved 99.23% accuracy. The best model was the BP-AdaBoost model, which was 99.49% accurate.

TABLE 8. A summary of used algorithms, used datasets, evaluation metrics, and obtained outcomes of NN-based earthquake researches

Ref.	Used Algorithm	Dataset	Evaluation Metrics	Performance
[78]	ANN	NCEDC	MAE, MSE	MSE was 0.115351
[80]		CNSS	P_0, P_1, Sn, Sp	Sp was 87.2% and Sn was 46.9%
[81]		Earthquake catalog of SGI		Sn was 87.2% (Alboran sea)
[90]				Sn was 80% and Sp was 91.30%
[82]		SES dataset of VAN group	MAE	Acc. was 83.56% (Magnitude prediction)
[83]		DEMETER satellite data	Acc.	Acc. was 81.2%
[84]		USGS		Acc. was 80% (mid-level eq.)
[85]			AVE, maximum error	Average error was 0.5% to 3.5% (eq. energy)
[91]		Data from JAMSTEC		Successfully predicted aftershocks
[86]		CENC	Error curve	Predicted time within 2 months of error
[79]			RMSE	Reduced the RMSE to 0.41ML
[92]	Embedded BPNN		PCC, RMSE, MAE	MSE was between 0.01 and 0.09
[87]	LM and BP		Error	With LM-BPNN error was 5.6
[32]		EAFS	RE	The highest relative error was 12%
[88]		DEMETER satellite	Acc.	Acc. was 83.90% (magnitude > 5.9)
[89]	LR and BP	NESDCCSN	RMSE	RMSE was ± 0.28 ML
[93]	NLFA and ANN	NOAA and USGS	R-value	The R-value was 0.525
[94]	HC and ANN	TSMS	RE	The highest prediction error was 6.25%
[95]	ANN and SVM	CENC	Acc.	Acc. was 73.37%
[37]	BP and AdaBoost	CESDC	Acc., Precision, Recall	Acc. was 99.49%
[96]	NARX	TGRC	Norm of error	The highest norm of error was 0.067
[97]	NDC and NDAP	SCEC	Acc., POD, FAR	Acc. was 99.4%
[98]	GMDHNN	USGS	MAE, RMSE, NMAE	MAE was 0.1734
[99]	MFOFLANN	USGS and SCEC	RMSE	RMSE was 0.0041
[35]	GA	GIM data by NASA JPL		DTEC value exceeded 54.66% (before 8 days)
[100]	ANN with GA		AE, RE	Averaged relative errors decreased to 0.4698
[101]	BP and BP-GA		AE	14 eq. will happen till 2044 (M > 6.5)
[36]	BP and PSO			The AE was 0.021 to 0.375
[102]	IABC with MLP	SCEC	Acc., MSE, NMSE	Acc. was 99.89%
[103]	Meta-Learning and TL		MSE	MSE 0.07
[104]	Ensemble of RBFNN		Acc.	Acc. was 91% (Magnitude)
[38]	RBFNN		AVE, maximum error	Average prediction error was 0.2722
[105]		GSS	MAE, RMSE	RMSE was 0.04217
[106]		USGS	RMSE, WIA	RMSE was 0.80
[107]		USGS and NEIC	Acc., P_0, POD, FAR	Acc. was 96%
[3]	PNN	SCEC	POD, FAR, R	POD score varies from 0 to 0.88
[108]			P_0, POD, FAR	Successfully predicted small magnitude events
[109]	KNN, NB, SVM, DT	CNSS	P_0, P_1, Sn, Sp	r, s, t, u, b_A/b_M, σ_1, σ_2 and n; but not ϕ worked best
[110]	DT, SVM, RF, LR		Acc.	Acc. was 90% (decision tree regressor)
[111]	RF, ANN, SVM	EMSC and ISC-GEM	P_0, P_1, Sn, Sp, MCC	RF and SVM achieved the best result
[112]	DT, RF, ROT Forest	USGS	F-Measure, Sn, Sp	F-Measure was 92.8% (ROT tree)
[113]	RF, NB, KNN, SVM		Acc.	Acc. was 75% (RF)
[114]	LR, AR, DT, MLP	KOERI and RETMC	Acc., MSE	LR provided the best Acc.

Legends: Ref.– Reference; MSE– mean squared error, MAE– mean absolute error; Acc.– accuracy; eq.– earthquake; RMSE– root mean square error; NMAE– normalized mean absolute error; POD– probability of detection; NCEDC– Northern California Earthquake Data Center; CNSS– Chile's National Seismological Service; DEMETER– Direction of Electro-Magnetic Emission Transmitted from Earthquake Regions; CENC– China Earthquake Network Center; SGI– Spanish Geographical Institute; USGS– United States Geological Survey; JAMSTEC– Japan Agency for Marine-Earth Science and Technology; NCEI– National Centers for Environmental Information; IMD– India Meteorological Department; NESDCCSN– National Earthquake Science Data Center of China Seismic Network; NOAA– National Oceanic and Atmospheric Administration; EAFS East Anatolian Fault System; TSMS– Turkish State Meteorological Service; CESDC– China Earthquake Science Data Center; TGRC– Tehran Geophysics Research Center; SCEC– South California Earthquake Data Center; GIM– Global Ionospheric Map; NASA– National Aeronautics and Space Administration; JPL– Jet Propulsion Laboratory; GSS– Guangxi Seismic Station; NEIC– National Earthquake Information Center; EMSC– Euro-Mediterranean Seismological Center; ISC– International Seismological Center; GEM– Global Earthquake Model; KOERI– Kandilli Observatory and Earthquake Research Institute; RETMC– Regional Earthquake Tsunami Monitoring Center; ML– Local magnitude; P_0– negative predictive error; P_1– positive predictive error; Sn– sensitivity; Sp– specificity; AVE– average error; PCC– percent correction classification; RE– relative error; FAR– false alert ratio; AE– absolute error; WIA– Willmott's Index of Agreement ; MCC– Matthews correlation coefficient.

Fong *et al.* [98] forecasted earthquake magnitude using the GMDH method. The worst residual error they observed was 0.2367, and the best was 0.1733. The standard deviation was between 0.11 and 0.19, which indicates that it is a stable model.

Akhoondzadeh [35] used GA for detecting anomalies in TEC before and after an earthquake. His model found that 8 days and 7 days before the earthquake, the TEC was 54.66% and 21.75% higher than the threshold, respectively. 10 hours after the earthquake, it was 29.8% higher than the higher bound.

For earthquake prediction, Chen *et al.* [105] proposed a chaos theory based RBFNN. This model achieved training RMSE of 0.0107 to 0.0109. The prediction RMSE varied between 0.042 and 0.049. D'Amico *et al.* [106] compared RBFNN with SVR for aftershock prediction with global RMSE values 0.80 and 0.88 for RBFNN and SVR. The RBFNN achieved a global WIA of 0.94, which was 0.02 better than that of the SVR model.

Asencio-Cortes *et al.* [109] tried to find out the best set of parameters for earthquake prediction. They tried KNN, SVM, NB, DT, and ANN algorithm in combination with the parameters to find the best possible set. They found that the start and end of the training and testing catalog, calculated b-value, lower and upper bound of magnitude threshold are the essential criteria for earthquake prediction.

For the NN-based methods, Min_{old} is equal to 71.5% and Max_{old} is equal to 99.89% for relative accuracy calculation. The relative RMSE is calculated with a lower bound of 0.0041 and upper bound of 1.19 to convert it to the range of 0 to 100. For average error, the Min_{old} and the Max_{old} are 0.0625 and 5.6 respectively. Based on these ranges, the relative performance conversion is done using Eq. (25). The relative performance classes are evaluated using Eqs. (26) and (27).

d: The Works Reporting the High Performance for Neural Network-based Approaches

The BP-AdaBoost [37] and the NDC-NDAP method [97] provided relative accuracy of 98.59% and 98.27%. The bio-inspired IABC-MLP method achieved a relative accuracy of 100%. The RBFNN model [107] also performed well with a relative accuracy of 86.30%. These models are high performing methods based on relative accuracy.

The embedded BPNN [92] and the LR-BP [89] methods achieved relative RMSE of 24.95% and 23.27% respectively. The GMDHNN method [98], Meta-learning with transfer learning method [103], and RBFNN-based method [105] performed well with relative RMSE of 16.98%, 21.97%, and 3.4% respectively. The MFOFLANN method [99] performed best based on the relative RMSE. These methods are high performing methods based on relative RMSE.

The LM-BP method [32] and the HC-ANN method [94] provided relative average error of 1.04% and 0%. The bio-inspired ANN-GA method [100] and the BP-PSO model [36] provided relative average error of 7.36% and 2.45% respectively. These models, along with the RBFNN method [38] are considered as high performing methods.

e: The Works Reporting the Medium Performance for Neural Network-based Approaches

The ANN-based models [82]–[84], [90] were all medium performing based on accuracy as their relative accuracies were 44.21%, 42.48%, 34.17%, and 29.68% respectively. The LM-BP model [88] and the ensemble of RBFNN technique [104] provided relative accuracy of 43.69% and 68.69% respectively. The PNN method [3], and the DT-regressor method [110] provided relative accuracy of 58.12% and 65.16% respectively. These models are reported as medium performing based on relative accuracy.

The ANN-based methods [78], [79] provided relative RMSE of 28.29% and 34.23% respectively. The RBFNN method [106] achieved relative RMSE of 67.11%. These are medium performing methods based on relative RMSE. The ANN-based method [85] achieved a relative average error of 34.99% and reported as medium performing.

f: The Works Reporting the Low Performance for Neural Network-based Approaches

The ANN-based methods [80], [81] achieved relative accuracy of 16.24% and 0% respectively. The ANN-SVM model [95] and the RF-based method [113] provided 6.59% and 12.33% relative accuracy, respectively. These models are reported as low performing methods based on relative accuracy.

The LR-based method [114] achieved relative RMSE of 100% and classified as a low performing model. The LM-BP model [87] achieved a relative average error of 100% for which it was also enlisted as a low performing model. The categorization of the studies are depicted in Fig. 15.

C. DEEP LEARNING APPROACHES

Here we will discuss the outcome of DL-based researches.

Kanarachos *et al.* [115] have used DNN with wavelet transform and Hilbert transform form-finding long-term patterns in the earthquake. They achieved an average TP rate of 83% on the 18th of April 1995. On the 19th of April, they found a TP rate of 86% before the Kozani Grevena earthquake.

Vardaan *et al.* [116] used LSTM for predicting the future trend of earthquakes. The overall R^2 score of the model was -0.252.

Wang *et al.* [39] used DL model to predict the magnitude of P-wave during an earthquake. For the error range of ± 0.5, this model predicted 84.21% samples correctly.

Wang *et al.* [42] proposed an LSTM based model for the prediction of earthquakes. With one-dimensional input, this model achieved an accuracy of 63.50%. With two-dimensional input, this model achieved an accuracy of 86% for (5×5) sub-region.

Asim *et al.* [117], compared PRNN, RNN, RF, and LP Boost algorithm for magnitude prediction of an earthquake. For unseen data, the LP Boost algorithm worked best with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

TABLE 9. A summary of used algorithms, datasets, evaluation metrics, and obtained outcomes of DL-based earthquake prediction studies

Ref.	Used Algorithm	Dataset	Evaluation Metrics	Performance
[115]	DNN	SES of 18th, 19th April 1995	Acc., ROC curve	Acc. was 83% (Anomaly detection)
[39]		KiK-net	Acc.	Acc. was 84.21% (Magnitude prediction)
[42]	LSTM	USGS		Acc. was 88.57% with 2D input
[116]			R^2	Achieved R^2 value of -0.252
[117]	RNN, PRNN, RF, LP Boost	CES and USGS	Acc., Sn,Sp, P_0, P_1	Acc. was 79% (PRNN and LPBoost)
[40]	RNN	SCEC	POD, FAR, R-score	Predicted location with 15-39 miles error
[65]	RNN, BP, RBF			POD score was 0.658 (RNN)
[118]	DNN, GLM, RF, GBM	NCEDC	MAE, RE	MAE was 0.74 (RF)

Legends: Ref.- Reference; Acc.- accuracy; MAE- mean absolute error; POD- probability of detection; SES- seismic electric signal; KiK-net- Kiban Kyoshin network; SCEC- South California Earthquake Data Center; USGS- United States Geological Survey; CES- Center for Earthquake Studies; NCEDC- National California Earthquake Data Center; ROC- receiver operation characteristics; FAR- false alarm ratio; Sn- sensitivity; Sp- specificity; P_0- negative predictive value; P_1- positive predictive value; RE- relative error.

65% accuracy, and RNN achieved 64% accuracy. PRNN produced the least amount of false alarms. Panakkat and Adeli [40] predicted the time and location of an earthquake using RNN. This model could predict the location of an earthquake with an error of 15 to 39 kilometers. The mainshock was predicted with an error of 75 to 94 days, and aftershocks were predicted with an error of 5 to 16 days. Panakkat and Adeli [65] compared BP, RNN, and RBFNN for predicting the most massive earthquake in the next month. The RNN based model performed best with an average R-score of 0.562 for different magnitude ranges. Asencio-Cortes *et al.* [118] compared DL, GLM, GBM, and RF for magnitude prediction of an earthquake. For the magnitude of 3 to 7, the RF algorithm performed best with a MAE of 0.6. Table 9 shows the algorithms, outcomes, used dataset, and the evaluation metrics by the DL-based researches.

For the DL-based methods, Min_{old} is set to 65.8% and Max_{old} is set to 88.57% for relative accuracy calculation. This conversion is done based on Eq. (25) so that the models can be categorized in three levels. The relative performance classes are evaluated using Eq. (26).

a: The Works Reporting the High Performance for Deep Learning-based Approaches

The DNN based models [39], [115] showed high performance with a relative accuracy of 75.53% and 80.85%. The LSTM model [42] performed best with a relative accuracy of 100%

b: The Works Reporting the Medium Performance for Deep Learning-based Approaches

The PRNN method [117] achieved relative accuracy of 57.97%. Therefore, this method was labelled as medium performing.

c: The Works Reporting the Low Performance

The vanilla RNN-based method [65] performed worst among the DL-based methods. This method provided a relative accuracy of 0% and reported as a low performing model. The categorization of the studies based on relative performance are presented in Fig. 15.

In Fig. 13, the performance of the models was compared based on accuracy of prediction. For the prediction of an earthquake, all the AI-based algorithms achieved accuracy by more than 60%. The rule-based expert system was able to predict earthquakes within 12 hours before an earthquake with 100% accuracy. The IABC-MLP method was also successful, with 99.89% accuracy. The NDC-NDAP and BP-AdaBoost algorithm also performed very well. For the detection of P-wave, RF was 100% accurate. For the sake of comparison, the studies that only used Sp and Sn as evaluation metrics were converted to accuracy using MatCal software. In Fig. 14(a), the performance of the models was compared based on MSE and RMSE. The ANFIS-ELM algorithm produced the least amount of error. The MFOFLANN model also produced the least amount of error. These algorithms did not exceed the MSE value of 0.8. Therefore, all the AI-based algorithms work well. In Fig. 14(b), the methods that used average error as evaluation metrics were compared. The NARX based method was most successful. We have observed that the performance of the NN optimized by the bio-inspired algorithms, performed well. More researches should be done based on these methods.

D. DISCUSSION ON PERFORMANCE BASED ON STUDY AREA

Different earthquake-related researches are discussed in this study. However, the performance of different methods on the same database needs to be addressed. Here, we will discuss the performance of different models on the different datasets based on relative performance discussed earlier. The methods are discussed based on high, medium and low performance. Some studies were done considering data from the whole world. The rule-based expert system [49] predicted earthquake with high relative accuracy. For reducing errors in forecasting GMDHNN [98] showed high performance on world data. For magnitude prediction based on the data around the world, RBFNN [107] and MFOFLANN based model [99] showed high performance. The ANN-SVM model [95] provided low performance.

For the study area of Iran, ANFIS [47] showed high performance for seismic moment prediction. For forecasting earth-

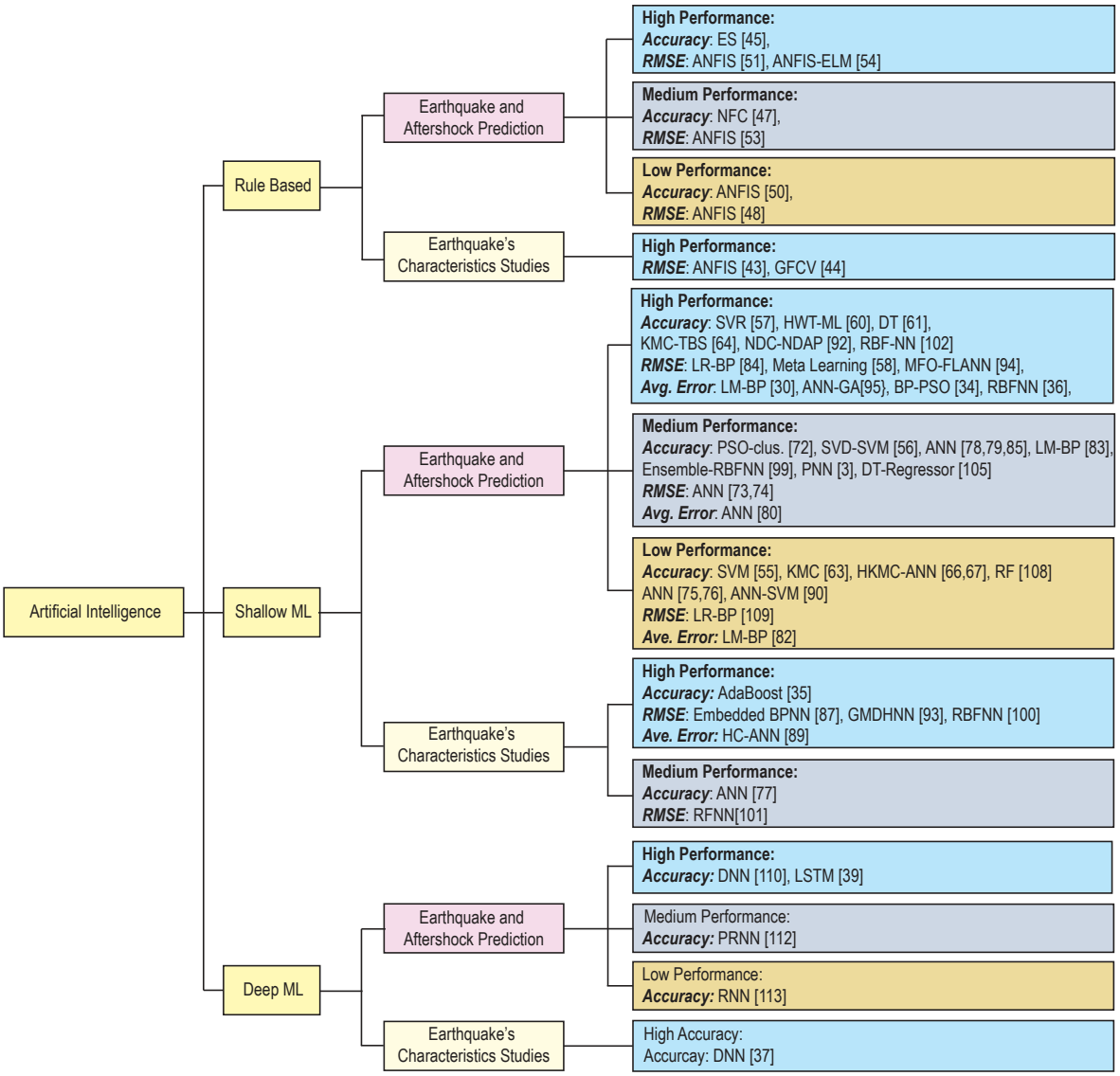

FIGURE 15. Categorization of performances for various rule-based, shallow/deep machine learning approaches. The studies are compared on their relative performance calculated by Eq. (25). Based on the relative performance, the studies are categorized in high, medium, and low classes.

quake ANFIS-ELM model [58] and the SVR-based model, [61] achieved high performance for Iranian region. For short-term earthquake prediction, NFC model [51] showed medium performance. For magnitude prediction, ANFIS-based model [57] provided medium performance, but the RF-based model [113] showed low performance on data of the Iranian region.

Some researches took China as their study area. For earthquake occurrence, LR-BP [89] and LSTM [42] model achieved high performance. For earthquake magnitude prediction, RBFNN [38] achieved high relative accuracy, ANN [79] and ensemble of RBFNN [104] produced medium relative performance and SVM-based model [59] showed low performance. Intensity near the epicenter in China was predicted by ANFIS [52] with low relative performance. Aftershock in China was predicted by PRKNN [63] with low performance.

For the study area of Indonesia, the magnitude of the earthquake was predicted by HKMC-ANN-based model [71], [72] with low performance and with ANFIS [55] it was predicted with high relative performance. ANFIS [54] predicted the occurrence and non-occurrence with low relative performance. GFCV-based method [48] showed high performance for b-value prediction for the Indonesian region.

In southern California, researches were done for earthquake magnitude prediction. The NDC-NDAP method [97] and the IABC-MLP method showed high performance for this area. The PNN [3] and the SVR-HNN [62] achieved medium performance. The RNN-based model [65] provided low performance for the south Californian region.

The KMC-TBS model [69] provided high accuracy for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029859, IEEE Access

magnitude prediction, but ANN-based [80] model achieved low performance for data of Chile. The best set of parameters were calculated by ANN [90] model with a relative medium performance for this area. The occurrence of the earthquake was predicted by KMC [68] with a low performance for the data of Chile.

For magnitude prediction in Greece, both the ANN-based model [82] and the SVM-SVD model achieved medium performance. The DNN-based model [115] showed high performance for SES anomaly prediction in the Greek area.

For the Turkish region, LR-based model [114] provided low accuracy for earthquake magnitude prediction, but HC-ANN model [94] achieved high relative performance for the occurrence of earthquake prediction.

In the Iberian Peninsula area, ANN model [81] achieve low performance for earthquake magnitude prediction. For the Himalayan region, ANN [84] provided medium performance for magnitude prediction. The optimized neural number was predicted with high performance by the embedded BPNN model [92] for the Philippines region. In the Hindukush region, PRNN [117] provided high relative performance for magnitude prediction.

VII. CHALLENGES AND FUTURE RESEARCH

Earthquake researches face great difficulties due to the rarity of data, quality of data collection, lack of pattern, and variability of performance of the same model in different geological locations. Most of these problems significantly affect the performance of the model. Here we will discuss the difficulties faced in earthquake researches, and some methods of overcoming these situations are also presented, which can be adopted in future studies.

A. PREDICTION ACCURACY LIMITED TO MAGNITUDE

All the earthquake prediction models reviewed in this paper performed very well in predicting magnitudes between 3 and 5. Naturally, an earthquake with a magnitude greater than magnitude 6 is rare events. When an earthquake with a magnitude higher than 6 happens, the AI-based models show poor performance because of data scarcity. Training the model separately with earthquake events greater than magnitude 6 can help in these scenarios. Generation of artificial data using models like generative adversarial networks can be used as well.

B. TIME PREDICTION ERROR

Since the occurrence of the earthquake does not follow some specific patterns, the earthquake time prediction usually faces significant errors. For long-term earthquake time prediction, 20 days to 5 months of error are observed in most cases. Prediction of earthquakes can be treated as a time-series analysis problem. Therefore, the best performing models in time-series analysis can be used for it. The earthquake prediction researchers should explore Attention-based architectures.

C. RESOLUTION

The epicenter is the point on the earth's surface vertically above the earthquake's position of generation. The quake's intensity is highest on the epicenter, which leads to the highest amount of damage. But epicentral location is usually predicted with 70 miles of error. Therefore, the perfect occurrence of an earthquake is very tough to predict. Clustering the earthquake events based on their location keeps the location-based patterns intact. Thus, clustering the earthquake events can help a lot for more accurate prediction.

D. NO BENCHMARK DATASETS

The earthquake prediction datasets typically contain earthquake catalog, SES, seismic waves, precursory parameters, or animal behavior. The probability of earthquake occurrence is small. Consequently, for a specific area, data is not sufficient. SES for the historical earthquake catalog is not available. VAN team managed to record SES for 29 events in quest of creating an SES dataset. Though this effort is not enough for machine learning or deep learning researches. The duration of the *P*-wave is minimal, which makes earthquake prediction very difficult. There is no benchmark dataset based on which model can be evaluated. The DL-based approach cannot be used due to the scarcity of large benchmark dataset. An earthquake dataset should be created on which every earthquake model can be tested for easing the comparison process of the different models.

E. EFFECT OF ENVIRONMENTAL FACTORS

The physics behind the occurrence of an earthquake is not easy to understand. Precursors such as radon concentration, soil temperature, strange waves are very troublesome to monitor. Monitoring animal behavior is not easy as well. So, the prediction studies of an earthquake are not built on strong in-depth knowledge of earthquake science. This leads to under-performing models. In-depth research on the location-dependent behavior of an earthquake is necessary for better understanding of this phenomenon.

F. INSUFFICIENT EFFECTIVE PARAMETERS

For earthquake prediction studies, the best set of earthquake parameters are not defined. Different studies use different parameter sets. The magnitude of completeness is different for different datasets as well. Based on these, b-value parameters are calculated. In a single dataset, different earthquake's magnitude techniques (Local-scale, Richter scale) are evaluated by different techniques. These kinds of variance make earthquake prediction studies more difficult. In most cases, the same method performs very differently, just because the dataset is different. The best set of earthquake features need to be defined for different geological locations. While recording the earthquake data, a specific magnitude scale should be adopted all over the world.

VIII. CONCLUSION

Of all the natural disasters, earthquake is one of the most devastating ones as it occurs suddenly, damages a significant number of infrastructures, and takes away lives. Many of the existing prediction techniques provide high false alarm, therefore, lack of accurate prediction process is a contributor to this catastrophic consequence of earthquake. AI-based methods have created a new scope for improving this prediction process due to their high accuracy when compared to other techniques. Such methods can significantly reduce damages as the concerned area can be evacuated based on the forecasting. To facilitate the prediction process, this paper reviewed the existing techniques that involve AI-based earthquake prediction. From the academic research databases 84 papers were selected between 2005 and 2019. The reported methods were summarized in tables and extensively discussed. Then these techniques were compared based on their performances. The reported results of the methods, used datasets, and applied evaluation metrics were also summarized in tables. This work aims to highlight the impact of AI-based techniques in earthquake prediction which will help the researchers to develop more accurate methods.

ACKNOWLEDGMENT

The authors would like to thank Bangladesh University of Professionals for supporting this research.

COMPLIANCE WITH ETHICAL STANDARDS

Funding: This research was supported by the research funds of Jeonbuk National University in 2020 and by the Information and Communication Technology division of the Government of the People's Republic of Bangladesh in 2019 - 2020.

Conflicts of Interest: All authors declare that they have no conflict of interest.

Ethical Approval: No ethical approval required for this study.

Informed Consent: This study used secondary data, therefore, the informed consent does not apply.

Authors and Contributors: This work was carried out in close collaboration between all co-authors.

REFERENCES

[1] "Search earthquake catalog," https://earthquake.usgs.gov/earthquakes/search/, 2020, accessed on May 11, 2020.
[2] D. E. Alexander, "The l'aquila earthquake of 6 april 2009 and italian government policy on disaster response," J. Nat. Resour. Policy Res., vol. 2, no. 4, pp. 325–342, 2010.
[3] H. Adeli and A. Panakkat, "A probabilistic neural network for earthquake magnitude prediction," Neural Netw., vol. 22, no. 7, pp. 1018–24, 2009.
[4] J. Patterson and A. Gibson, Deep learning: A practitioner's approach. O'Reilly Media, 2017.
[5] Y. Ma et al., "Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis," Cogn. Comput., vol. 10, no. 4, pp. 639–650, 2018.
[6] X. Yang, K. Huang, R. Zhang, and J. Y. Goulermas, "A novel deep density model for unsupervised learning," Cogn. Comput., vol. 11, no. 6, pp. 778–788, 2019.
[7] R. D. Cicerone, J. E. Ebel, and J. Britton, "A systematic compilation of earthquake precursors," Tectonophysics, vol. 476, pp. 371–96, 2009.
[8] M. Wyss and D. C. Booth, "The iaspei procedure for the evaluation of earthquake precursors," Geophys. J. Int., vol. 131, no. 3, pp. 423–424, 1997.
[9] R. J. Geller, "Earthquake prediction: a critical review," Geophys. J. Int., vol. 131, no. 3, pp. 425–450, 1997.
[10] L. Sevgi, "A critical review on electromagnetic precursors and earthquake prediction," Turk. J. Electr. Eng. Comput. Sci., vol. 15, no. 1, pp. 1–15, 2007.
[11] S. Uyeda, T. Nagao, and M. Kamogawa, "Short-term earthquake prediction: Current status of seismo-electromagnetics," Tectonophysics, vol. 470, no. 3-4, pp. 205–213, 2009.
[12] M. Mubarak et al., "Earthquake prediction: a global review and local research," Proc. Pakistan Acad. Sci, vol. 46, no. 4, pp. 233–246, 2009.
[13] N. Bhargava, V. Katiyar, M. Sharma, and P. Pradhan, "Earthquake prediction through animal behavior: A review," Indian J. Biomech, vol. 78, pp. 159–165, 2009.
[14] H. V. Alvan and F. H. Azad, "Satellite remote sensing in earthquake prediction. a review," in Proc. NatPC, 2011, pp. 1–5.
[15] G. Otari and R. Kulkarni, "A review of application of data mining in earthquake prediction," Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 2, pp. 3570–3574, 2012.
[16] F. Azam, M. Sharif, M. Yasmin, and S. Mohsin, "Artificial intelligence based techniques for earthquake prediction: a review," Sci Int, vol. 26, no. 4, pp. 1495–1502, 2014.
[17] H. Woith, "Radon earthquake precursor: A short review," Eur. Phys. J. Spec. Top., vol. 224, no. 4, pp. 611–627, 2015.
[18] F. Huang et al., "Studies on earthquake precursors in china: A review for recent 50 years," Geod. Geodyn., vol. 8, no. 1, pp. 1–12, 2017.
[19] S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, and B. Chakraborty, "A review on application of data mining techniques to combat natural disasters," Ain Shams Eng. J., vol. 9, no. 3, pp. 365–378, 2018.
[20] A. Galkina and N. Grafeeva, "Machine learning methods for earthquake prediction: a survey," in Proc. SEIM-2019, 2019, p. 25.
[21] P. Jiao and A. H. Alavi, "Artificial intelligence in seismology: Advent, performance and future trends," Geosci. Front., 2019.
[22] A. Mignan and M. Broccardo, "Neural network applications in earthquake prediction (1994-2019): Meta-analytic insight on their limitations," arXiv preprint arXiv:1910.01178, 2019.
[23] S. Uyeda, "The van method of short-term earthquake prediction," IN-CEDE Newsl., vol. 5, p. 4, 1997.
[24] S. López, A. A. Márquez, F. A. Márquez, and A. Peregrín, "Evolutionary design of linguistic fuzzy regression systems with adaptive defuzzification in big data environments," Cogn. Comput., vol. 11, no. 3, pp. 388–399, 2019.
[25] M. S. Kaiser et al., "A neuro-fuzzy control system based on feature extraction of semg signal for solar-powered wheelchair," Cogn. Comput., vol. 8, no. 5, pp. 946–954, 2016.
[26] G. Sun et al., "Multi-scale mahalanobis kernel-based support vector machine for classification of high-resolution remote sensing images," Cogn. Comput., pp. 1–8, 2019.
[27] I. Aljarah et al., "Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm," Cogn. Comput., vol. 10, no. 3, pp. 478–495, 2018.
[28] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," Int. J. Eng. Res. Appl., vol. 3, no. 5, pp. 605–610, 2013.
[29] A. Lulli, L. Oneto, and D. Anguita, "Mining big data with random forests," Cogn. Comput., vol. 11, no. 2, pp. 294–316, 2019.
[30] S. L. Salzberg, "C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993," 1994.
[31] M. Kaushik and B. Mathur, "Comparative study of k-means and hierarchical clustering techniques," Int. J. Softw. Hardw. Res. Eng., vol. 2, no. 6, pp. 93–98, 2014.
[32] F. Külahcı, M. İnceöz, M. Doğru, E. Aksoy, and O. Baykara, "Artificial neural network model for earthquake prediction with radon monitoring," Applied Radiation and Isotopes, vol. 67, no. 1, pp. 212–219, 2009.
[33] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," J. soc. Ind. Appl. Math., vol. 11, no. 2, pp. 431–441, 1963.

[34] Q. Zhang and C. Wang, "Using genetic algorithm to optimize artificial neural network: A case study on earthquake prediction," in Proc. 2008 2nd Int. Conf. Genet. Evol. Comput. IEEE, 2008, pp. 128–131.

[35] M. Akhoondzadeh, "Genetic algorithm for tec seismo-ionospheric anomalies detection around the time of the solomon (mw= 8.0) earthquake of 06 february 2013," Adv. Space Res., vol. 52, no. 4, pp. 581–590, 2013.

[36] C. Li and X. Liu, "An improved pso-bp neural network and its application to earthquake prediction," in 2016 IEEE CCDC, 2016, pp. 3434–3438.

[37] G. Zhao, H. Huang, and X. Lu, "Discriminating earthquakes and explosion events by seismic signals basing on bp-adaboost classifier," in Proc 2016 IEEE ICCC, 2016, pp. 1965–1969.

[38] Y. Wang, Y. Chen, and J. Zhang, "The application of rbf neural network in earthquake prediction," in Proc. WGEC, 2009, pp. 465–468.

[39] Y. Wang, Z. Wang, Z. Cao, and J. Lan, "Deep learning for magnitude prediction in earthquake early warning," arXiv preprint arXiv:1912.05531, 2019.

[40] A. Panakkat and H. Adeli, "Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators," Comput.-Aided Civ. Infrastruct. Eng., vol. 24, no. 4, pp. 280–292, 2009.

[41] H. Cai et al., "Modeling marked temporal point process using multi-relation structure rnn," Cogn. Comput., pp. 1–14, 2019.

[42] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake prediction based on spatio-temporal data mining: an lstm network approach," IEEE Trans. Emerg. Top. Comput., 2017.

[43] J. Zheng et al., "Hierarchical neural representation for document classification," Cogn. Comput., vol. 11, no. 2, pp. 317–327, 2019.

[44] Z. Bofeng and L. Yue, "Customized explanation in expert system for earthquake prediction," in Proc. IEEE ICTAI'05, 2005, pp. 5–pp.

[45] M. Zhong and Q. Zhang, "Prediction of reservoir-induced earthquake based on fuzzy theory," in Proc. ISNNS, 2010, pp. 101–104.

[46] A. Konstantaras et al., "Detection of weak seismo-electric signals upon the recordings of the electrotelluric field by means of neuro-fuzzy technology," IEEE Geosci. Remote Sens. Lett., vol. 4, no. 1, pp. 161–165, 2007.

[47] M. Mirrashid, M. Givehchi, M. Miri, and R. Madandoust, "Performance investigation of neuro-fuzzy system for earthquake prediction," Asian J. Civ. Eng., vol. 17, pp. 213–223 ·, 01 2016.

[48] B. Rahmat, F. Afiadi, and E. Joelianto, "Earthquake prediction system using neuro-fuzzy and extreme learning machine," in Proc. ICST 2018, 2018.

[49] A. Ikram and U. Qamar, "Developing an expert system based on association rules and predicate logic for earthquake prediction," Knowl.-Based Syst., vol. 75, pp. 87–103, 2015.

[50] A. Konstantaras, F. Vallianatos, M. R. Varley, and J. P. Makris, "Soft-computing modelling of seismicity in the southern hellenic arc," IEEE Geosci. Remote Sens. Lett., vol. 5, no. 3, pp. 323–327, 2008.

[51] L. Dehbozorgi and F. Farokhi, "Notice of retraction: Effective feature selection for short-term earthquake prediction using neuro-fuzzy classifier," in Proc. 2010 IITA-GRS, vol. 2. IEEE, 2010, pp. 165–169.

[52] W. Zeng, Y. Zhao, and Q. Yin, "Sugeno fuzzy inference algorithm and its application in epicentral intensity prediction," Appl. Math. Model., vol. 40, no. 13-14, pp. 6501–6508, 2016.

[53] A. Andalib, M. Zare, and F. Atry, "A fuzzy expert system for earthquake prediction, case study: the zagros range," arXiv preprint arXiv:1610.04028, 2016.

[54] M. N. Shodiq, D. H. Kusuma, M. G. Rifqi, A. R. Barakbah, and T. Harsono, "Adaptive neural fuzzy inference system and automatic clustering for earthquake prediction in indonesia," Int. J. Inform. Vis., vol. 3, no. 1, pp. 47–53, 2019.

[55] R. Kamath and R. Kamat, "Earthquake magnitude prediction for andaman-nicobar islands: Adaptive neuro fuzzy modeling with fuzzy subtractive clustering approach," J. Chem. Pharm. Sci., vol. 10, no. 3, pp. 1228–1233, 2017.

[56] A. Pandit and K. C. Biswal, "Prediction of earthquake magnitude using adaptive neuro fuzzy inference system," Earth Sci. Inform., vol. 12, no. 4, pp. 513–524, 2019.

[57] M. Mirrashid, "Earthquake magnitude prediction by adaptive neuro-fuzzy inference system (anfis) based on fuzzy c-means algorithm," Nat. hazards, vol. 74, no. 3, pp. 1577–1593, 2014.

[58] B. Bahrami and M. Shafiee, "Fuzzy descriptor models for earthquake time prediction using seismic time series," Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol. 23, no. 04, pp. 505–519, 2015.

[59] C. Jiang, X. Wei, X. Cui, and D. You, "Application of support vector machine to synthetic earthquake prediction," Earthq. Sci., vol. 22, no. 3, pp. 315–320, 2009.

[60] W. Astuti, R. Akmeliawati, W. Sediono, and M.-J. E. Salami, "Hybrid technique using singular value decomposition (svd) and support vector machine (svm) approach for earthquake prediction," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 7, no. 5, pp. 1719–1728, 2014.

[61] P. Hajikhodaverdikhan, M. Nazari, M. Mohsenizadeh, S. Shamshirband, and K.-w. Chau, "Earthquake prediction with meteorological data by particle filter-based support vector regression," Eng. Appl. Comput. Fluid Mech., vol. 12, no. 1, pp. 679–688, 2018.

[62] Z. Huang, C. Yang, X. Zhou, and S. Yang, "Energy consumption forecasting for the nonferrous metallurgy industry using hybrid support vector regression with an adaptive state transition algorithm," Cogn. Comput., pp. 1–12, 2019.

[63] A. Li and L. Kang, "Knn-based modeling and its application in aftershock prediction," in Proc. ASIA, 2009, pp. 83–86.

[64] N. Prasad, K. K. Reddy, and R. T. Nirjogi, "A novel approach for seismic signal magnitude detection using haar wavelet," in 2014 IEEE ISMS, 2014, pp. 324–329.

[65] A. Panakkat and H. Adeli, "Neural network models for earthquake magnitude prediction using multiple seismicity indicators," Int. j. neural syst., vol. 17, no. 01, pp. 13–33, 2007.

[66] I. U. Sikder and T. Munakata, "Application of rough set and decision tree for characterization of premonitory factors of low seismic activity," Expert Syst. Appl., vol. 36, no. 1, pp. 102–110, 2009.

[67] Marisa, U. A. Sembiring, and H. Margaretha, "Earthquake probability prediction in sumatra island using poisson hidden markov model (hmm)," in AIP Conference Proceedings, vol. 2192, no. 1, 2019, p. 090006.

[68] E. Florido, F. Martínez-Álvarez, A. Morales-Esteban, J. Reyes, and J. L. Aznarte-Mellado, "Detecting precursory patterns to enhance earthquake prediction in chile," Comput. Geosci., vol. 76, pp. 112–120, 2015.

[69] E. Florido, G. Asencio-Cortés, J. L. Aznarte, C. Rubio-Escudero, and F. Martínez-Álvarez, "A novel tree-based algorithm to discover seismic patterns in earthquake catalogs," Comput.Geosci., vol. 115, pp. 96–104, 2018.

[70] M. N. Shodiq, A. R. Barakbah, and T. Harsono, "Spatial analysis of earthquake distribution with automatic clustering for prediction of earthquake seismicity in indonesia," in Proc. KCIC, 2015, pp. 24–26.

[71] M. Shodiq, D. Hidayat Kusuma, M. Ghulam Rifqi, A. Barakbah, and T. Harsono, "Neural network for earthquake prediction based on automatic clustering in indonesia," Int. J. Inform. Vis., vol. 2, p. 37, 02 2018.

[72] M. N. Shodiq, D. H. Kusuma, M. G. Rifqi, A. R. Barakbah, and T. Harsono, "Spatial analisys of magnitude distribution for earthquake prediction using neural network based on automatic clustering in indonesia," in Proc. IES-KCIC, 2017, pp. 246–251.

[73] J. Mejia, K. Rojas, N. Valeza, and A. R. Villagracia, "Earthquake prediction through kannan-mathematical-model analysis and dobrovolsky-based clustering technique," in DLSU Research Congress, 2015.

[74] K. Tan and X. Cai, "Prediction of earthquake in yunnan region based on the ahc over sampling," in Proc. 2010 IEEE CCDC, 2010, pp. 2449–2452.

[75] Y. Lu, E. Sawada, and G. Chakraborty, "Earthquake aftershock prediction based solely on seismograph data," in Proc. BigComp, 2019, pp. 1–6.

[76] X. Shao, X. Li, L. Li, and X. Hu, "The application of ant-colony clustering algorithm to earthquake prediction," in Adv. Electron. Eng. Commun. Manag., 2012, pp. 145–150.

[77] X. Y. Zhang, X. Li, and X. Lin, "The data mining technology of particle swarm optimization algorithm in earthquake prediction," in Adv. Mater. Res., vol. 989, 2014, pp. 1570–1573.

[78] A. S. Alarifi, N. S. Alarifi, and S. Al-Humidan, "Earthquakes magnitude predication using artificial neural network in northern red sea area," J. King Saud University-Science, vol. 24, no. 4, pp. 301–313, 2012.

[79] W. S. Hu, H. Wang, and H. L. Nie, "Regional short-term earthquake prediction model based on bp neural network," in Appl. Mech. Mater., vol. 166, 2012, pp. 2309–2314.

[80] J. Reyes, A. Morales-Esteban, and F. Martínez-Álvarez, "Neural networks to predict earthquakes in chile," Appl. Soft Comput., vol. 13, no. 2, pp. 1314–1328, 2013.

[81] A. Morales-Esteban, F. Martínez-Álvarez, and J. Reyes, "Earthquake prediction in seismogenic areas of the iberian peninsula based on computational intelligence," Tectonophysics, vol. 593, pp. 121–134, 2013.

[82] M. Moustra, M. Avraamides, and C. Christodoulou, "Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals," Expert syst. appl., vol. 38, no. 12, pp. 15 032–15 039, 2011.

[83] F. Xu, X. Song, X. Wang, and J. Su, "Neural network model for earthquake prediction using dmeter data and seismic belt information," in Proc. GCIS, vol. 3, 2010, pp. 180–183.
[84] S. Narayanakumar and K. Raja, "A bp artificial neural network model for earthquake magnitude prediction in himalayas, india," Circuits Syst., vol. 7, no. 11, pp. 3456–3468, 2016.
[85] A. Cheraghi and A. Ghanbari, "Study of risk analysis and earthquake magnitude and timing prediction via tectonic and geotechnical properties of the faults and identifying risky areas in terms of seismicity in larestan city using ann," Rev. QUID, no. 1, pp. 1137–1142, 2017.
[86] J. Xie et al., "The application of neural network model in earthquake prediction in east china," in Adv. Comput. Sci. Intell. Syst. Environ., 2011, pp. 79–84.
[87] F. Zhou and X. Zhu, "Earthquake prediction based on lm-bp neural network," in Proc.9th LDIA, 2014, pp. 13–20.
[88] L. Ma, F. Xu, X. Wang, and L. Tang, "Earthquake prediction based on levenberg-marquardt algorithm constrained back-propagation neural network using demeter data," in Proc. KSEM. Springer, 2010, pp. 591–596.
[89] W. S. Hu, H. L. Nie, and H. Wang, "Applied research of bp neural network in earthquake prediction," in Applied Mechanics and Materials, vol. 204, 2012, pp. 2449–2454.
[90] F. Martínez-Álvarez, J. Reyes, A. Morales-Esteban, and C. Rubio-Escudero, "Determining the best set of seismicity indicators to predict earthquakes. two case studies: Chile and the iberian peninsula," Knowl.-Based Syst., vol. 50, pp. 198–210, 2013.
[91] A. Okada and Y. Kaneda, "Neural network learning: Crustal state estimation method from time-series data," in Proc. 2018 IEEE ICCAIRO, 2018, pp. 141–146.
[92] J.-W. Lin, C.-T. Chao, and J.-S. Chiou, "Determining neuronal number in each hidden layer using earthquake catalogues as training data in training an embedded back propagation neural network for predicting earthquake magnitude," IEEE Access, vol. 6, pp. 52 582–52 597, 2018.
[93] S. S. Lakshmi and R. Tiwari, "Model dissection from earthquake time series: A comparative analysis using modern non-linear forecasting and artificial neural network approaches," Comput. Geosci., vol. 35, no. 2, pp. 191–204, 2009.
[94] S. Niksarlioglu and F. Kulahci, "An artificial neural network model for earthquake prediction and relations between environmental parameters and earthquakes," in Proc. WASET, no. 74, 2013, p. 616.
[95] W.-z. Zhou et al., "Study on seismic magnitude prediction based on combination algorithm," in Proc. ICMIC, 2017, pp. 539–544.
[96] A. A. Suratgar, F. Setoudeh, A. H. Salemi, and A. Negarestani, "Magnitude of earthquake prediction using neural network," in Proc. ICNC, vol. 2, 2008, pp. 448–452.
[97] M. H. Rafiei and H. Adeli, "Neews: a novel earthquake early warning model using neural dynamic classification and neural dynamic optimization," Soil Dyn. Earthq. Eng., vol. 100, pp. 417–427, 2017.
[98] S. Fong, Z. Nannan, R. K. Wong, and X.-S. Yang, "Rare events forecasting using a residual-feedback gmdh neural network," in Proc. ICDMW, 2012, pp. 464–473.
[99] S. K. Majhi, S. S. Hossain, and T. Padhi, "Mfoflann: moth flame optimized functional link artificial neural network for prediction of earthquake magnitude," Evol. Syst., vol. 11, no. 1, pp. 45–63, 2020.
[100] Q. Zhang and C. Wang, "Using genetic algorithm to optimize artificial neural network: A case study on earthquake prediction," in Proc. WGEC, 2008, pp. 128–131.
[101] Z. Tao, "Artificial neural network attempts for long-term evaluation of great earthquakes," in 2015 IEEE ICNC, 2015, pp. 1128–1132.
[102] H. Shah and R. Ghazali, "Prediction of earthquake magnitude by an improved abc-mlp," in Proc. 2011 IEEE DESE, 2011, pp. 312–317.
[103] M. Maya and W. Yu, "Short-term prediction of the earthquake through neural networks and meta-learning," in Proc. CCE, 2019, pp. 1–6.
[104] Y. Liu, Y. Li, G. Li, B. Zhang, and G. Wu, "Constructive ensemble of rbf neural networks and its application to earthquake prediction," in Proc. ISNN, 2005, pp. 532–537.
[105] Y. Chen, J. Zhang, and J. He, "Research on application of earthquake prediction based on chaos theory," in Proc. ICISS, 2010, pp. 753–756.
[106] S. D'Amico et al., "Heuristic advances in identifying aftershocks in seismic sequences," Comput. Geosci., vol. 35, pp. 245–254, 02 2009.
[107] E. Amar, T. Khattab, and F. Zad, "Intelligent earthquake prediction system based on neural network," Int. J. Environ. Chem. Ecol. Geol. Geophys. Eng., vol. 8, no. 12, p. 874, 2014.
[108] S.-Z. Huang, "The prediction of the earthquake based on neural networks," in Proc. ICCDA, vol. 2, 2010, pp. V2–517.
[109] G. Asencio-Cortés, F. Martínez-Álvarez, A. Morales-Esteban, and J. Reyes, "A sensitivity study of seismicity indicators in supervised learning to improve earthquake prediction," Knowl.-Based Syst., vol. 101, pp. 15–30, 2016.
[110] W. Li et al., "Seismic data classification using machine learning," in Proc. BigDataService. IEEE, 2018, pp. 56–63.
[111] K. M. Asim et al., "Seismicity analysis and machine learning models for short-term low magnitude seismic activity predictions in cyprus," Soil Dyn. Earthq. Eng., vol. 130, p. 105932, 2020.
[112] K. M. Asim, A. Idris, F. Martínez-Álvarez, and T. Iqbal, "Short term earthquake prediction in hindukush region using tree based ensemble learning," in Proc. FIT, 2016, pp. 365–370.
[113] S. Karimzadeh, M. Matsuoka, J. Kuang, and L. Ge, "Spatial prediction of aftershocks triggered by a major earthquake: A binary machine learning perspective," ISPRS Int. J. Geo-Inf., vol. 8, no. 10, p. 462, 2019.
[114] E. Celik, M. Atalay, and A. Kondiloğlu, "The earthquake magnitude prediction used seismic time series and machine learning methods," in Proc. ENTECH, 12 2016.
[115] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform," Expert Syst. Appl., vol. 85, pp. 292–304, 2017.
[116] T. Bhandarkar, K. Vardaan, N. Satish, S. Sridhar, R. Sivakumar, and S. Ghosh, "Earthquake trend prediction using long short-term memory rnn," Int. J. Electr. Comput. Eng., vol. 9, no. 2, p. 1304, 2019.
[117] K. Asim, F. Martínez-Álvarez, A. Basit, and T. Iqbal, "Earthquake magnitude prediction in hindukush region using machine learning techniques," J. Chem. Pharm. Sci., vol. 85, no. 1, pp. 471–486, 2017.
[118] G. Asencio-Cortés, A. Morales-Esteban, X. Shang, and F. Martínez-Álvarez, "Earthquake prediction in california using regression algorithms and cloud-based big data infrastructure," Comput. Geosci., vol. 115, pp. 198–210, 2018.