Article

Classify earthquakes using Machine Learning algorithms

Constantina Kopitsa[1], Ioannis G. Tsoulos[2], Vasileios Charilogis[3] and Chrysostomos Stylios[4,*]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece;k.kopitsa@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece; itsoulos@uoi.gr
[3] Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr
[4] Department of Informatics and Telecommunications, University of Ioannina, Greece; stylios@uoi.gr
* Correspondence: stylios@uoi.gr

Abstract

The predictability of earthquakes remains a central challenge in seismological research. Are earthquakes inherently unpredictable phenomena, or can they be forecasted through advances in technology? Contemporary seismological research continues to pursue this scientific milestone, often referred to as the 'Holy Grail' of earthquake prediction. In the direction of earthquake prediction based on historical data, the Grammatical Evolution technique of GenClass demonstrated high predictive accuracy for earthquake magnitude. Similarly, our research team follows this line of reasoning, operating under the belief that nature provides a pattern that, with the appropriate tools, can be decoded. What is certain is that, over the past 30 years, scientists and researchers have made significant strides in the field of seismology, largely aided by the development and application of artificial intelligence techniques. Artificial Neural Networks (ANNs) were first applied in the domain of seismology in 1994. The introduction of Deep Neural Networks (DNNs), characterized by architectures incorporating two hidden layers, followed in 2002. Subsequently, Recurrent Neural Networks (RNNs) were implemented within seismological studies as early as 2007. Most recently, Grammatical Evolution (GE) has been introduced in seismological studies (2025). Despite ongoing advancements, the so-called "triple prediction" accurately forecasting the time, location, and magnitude of a seismic event, remains unachieved. Beyond that, machine learning and soft computing techniques have maintained a longstanding presence in the field of seismology. Concerning these approaches, significant advancements have been achieved, both in mapping seismic patterns and in predicting seismic characteristics on a smaller geographical scale. In such a way, our research will analyze historical seismic events from 2004 to 2011, for the Latitude 21°-79° & Longitude 33°-176°. The data will be categorized and classified, with the aim of employing Grammatical Evolution techniques to achieve more accurate and timely predictions of earthquake magnitudes. This paper presents a systematic effort to enhance magnitude prediction accuracy using GE, contributing to the broader goal of reliable earthquake forecasting.Subsequently, in this paper we present the superiority of GenClass, a key element of the Grammatical Evolution techniques, with an average error of 19%, indicating an overall accuracy of 81%.

Keywords: Earthquakes; Machine learning; Neural networks; Grammatical Evolution;

Received:
Revised:
Accepted:
Published:

Citation: Kopitsa, C.; Tsoulos, I.G.; Charilogis, V.; Stylios, C. . Predicting the magnitude of earthquakes using Grammatical Evolution. *Journal Not Specified* **2025**, *1*, 0. https://doi.org/

1. Introduction

Seismology was first established as an academic discipline in 1876 at the Imperial College of Engineering in Tokyo, when John Milne (1850-1913) was invited to teach there, and in 1886, S. Sekiya (1855-1896) was appointed as the world's first professor dedicated specifically to the field of seismology [1]. Nevertheless, the seismologists who left a lasting legacy lending their names to the classification of earthquakes, were Charles Richter (1900-1985) and Giuseppe Mercalli (1850-1914). The former developed, in 1935, a scale measuring the magnitude of seismic events, it is outlined in Table ranging from 1 to 9 [2] while the latter devised a scale that categorizes the destructive effects of an earthquake, it is shown in Table ranging from levels 1 to 12 [3]. The introduction of the Richter scale marked a significant advancement in seismology by establishing a standardized method for quantifying earthquake magnitude. This standardization enabled consistent comparisons between seismic events, thereby enhancing the analysis, preparedness, and management of earthquake-related disasters [4]. On the other hand, the Mercalli scale assessed the effects of an earthquake using qualitative criteria based on observed impacts on buildings, infrastructure, and human perception.

Table 1. The Richter Scale (each level is 10 times stronger than the previous).

Magnitude	Consequences / Effects
0 - 1.9	Detected only by seismograph
2 - 2.9	Hanging objects may swing
3 - 3.9	People near the epicenter feel this quake
4 - 4.9	Can cause damage around the epicenter
5 - 5.9	Can cause damage to weak buildings in the epicenter area
6 - 6.9	Can cause great damage to well-built structures
7 - 7.9	Can cause serious damage to buildings' foundations
8 - 8.9	Causes death and major destruction
9 - ...	Total destruction

Table 2. The Modified Mercalli Intensity Scale (near the epicenter of the earthquake).

Scale Level	Consequences / Effects
I	Not felt except by very few under especially favorable conditions.
II	Felt only by a few people
III	Felt quite noticeably by people indoors
IV	Felt indoors by many, and outdoors by few
V	Felt by nearly everyone
VI	Felt by all
VII	Damage is negligible in buildings of good design and construction
VIII	Damage is slight in specially designed structures
IX	Damage is considerable in specially designed structures
X	Some well-built structures are destroyed
XI	Few,structures remain standing
XII	Damage is total

These measurements proved particularly applicable in urban environments, especially within Western societies, where systematic documentation of earthquake damage was feasible [5]. The classification of earthquakes has made it possible not only to develop seismic hazard maps but also to organize data systematically for the development of algorithms aimed at predicting seismic events. In Figure 1a hazard map issued by the Global https://www.globalquakemodel.org/product/global-seismic-risk-map is outlined, indicating a 10% probability of being exceeded within 50 years.

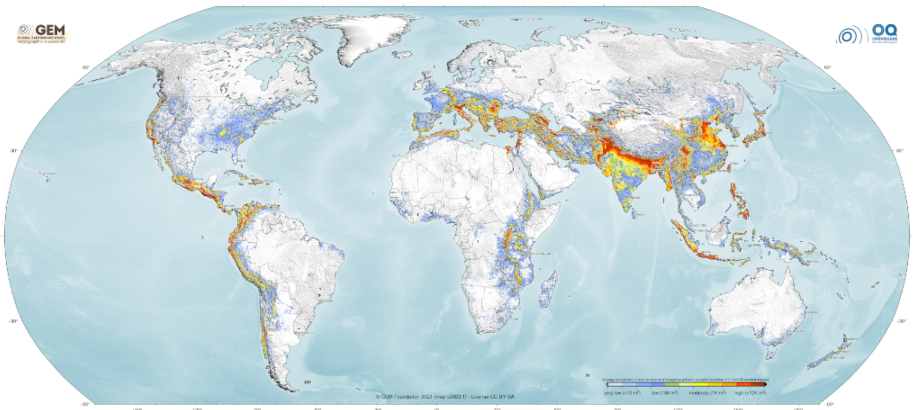

Figure 1. Hazard map from Global Earthquake Model.

In this context, researchers are increasingly utilizing artificial intelligence and other advanced techniques to enhance the timely forecasting of earthquakes and other natural disasters, with the ultimate aim of helping communities take protective measures and mitigate potential impacts [6]. A series of machine learning techniques were applied on data gathered from earthquakes. For example, Artificial neural networks (ANNs) [7] were first applied to the field of seismology in 1994, specifically through research focused on earthquake prediction using seismic electric signals [8]. A more advanced form of an Artificial Neural Network (ANN) was employed in the related study [9]. Furthermore, Mousavi et al. [10] suggested an attentive deep-learning model for simultaneous earthquake detection. Moreover, researchers utilized convolutional neural networks (CNNs) [11] trained on more than 18 million manually annotated seismograms from Southern California to estimate earthquake parameters directly from raw waveform data, eliminating the need for manual feature extraction. The model exhibited remarkable precision, achieving a standard deviation of only 0.023 seconds in arrival-time estimation and a 95% accuracy rate in polarity classification [12].

The earliest application of Deep Neural Networks (DNNs) [13], incorporating two hidden layers, was introduced in 2002 [14]. Subsequently, a high-resolution earthquake catalog was developed using DNN techniques, offering valuable insights into the complexity and duration of earthquake sequences, as well as their relationships with recent neighboring seismic events, as discussed in the related work [15]. Additionally, Recurrent Neural Networks (RNNs) [16] within the field of seismology was introduced in 2007 [17]. Furthermore, a subsequent study regarding Earthquake magnitude prediction using machine learning techniques was introduced recently [18], concentrated on forecasting earthquake magnitudes in the Hindukush region. Four machine learning approaches neural network-based pattern recognition, Recurrent Neural Networks, Random Forests [19], and a Linear Programming Boost ensemble classifier were individually applied to model the relationships between calculated seismic parameters and the likelihood of future earthquake occurrences. Also, a subsequent study demonstrated that nearest-neighbor diagrams provide a simple yet effective approach for differentiating between distinct seismic patterns and assessing the reliability of earthquake catalogs [20]. Another research team, used Nearest Neighbor method, determined that the Weibull model offered a more accurate fit for seismic data in California, showing well-structured tail behavior. They further validated its robustness by successfully applying it to independent datasets from Japan, Italy, and New Zealand, in the related publication [21].

Building on recent advancements, Rouet-Leduc et al. applied machine learning techniques to datasets obtained from shear laboratory experiments, with the objective of identifying previously undetected signals that could potentially precede seismic events

[22]. In a subsequent study, the same research team utilized a machine learning-based approach, originally developed in the laboratory, to analyze extensive raw seismic data from Vancouver Island. This methodology enabled the differentiation of pertinent seismic signals from background noise and holds promise for evaluating whether, and under what conditions, a slow slip event might be associated with or evolve into a major earthquake [23]. Additionally, in a recent study a predictive model was developed, capable of estimating both the location and magnitude of potential earthquakes in the following week, utilizing seismic data from the current week and focusing on seismogenic regions in southwestern China. The model demonstrated a testing accuracy of 70%, with corresponding precision, recall, and F1-score values of 63.63%, 93.33%, and 75.66%, respectively [24]. Moreover, in another publication, the research team demonstrated that machine learning techniques can effectively predict the timing and magnitude of laboratory-induced earthquakes by reconstructing and interpreting the system's complex spatiotemporal loading history [25].

In this paper, a technique based on Grammatical Evolution [26] is proposed for the efficient creation of classification rules on seismic data that are widely available on the internet. Grammatical Evolution is a genetic algorithm [27], where the chromosomes are series of production rules from a provided Backus-Naur form (BNF) grammar [28]. Among the numerous cases of application of Grammatical Evolution, one can find problems such as function approximation[29,30], economic problems[31], network security issues [32], water quality problems [33], medical problems [34], evolutionary computation [35], prediction of temperature in data centers [36], solving trigonometric problems [37], composing music [38], construction of neural networks [39,40], numerical problems [41], video games [42, 43], energy issues [44], combinatorial optimization [45], security issues [46], evolution of decision trees [47], problems that appear in electronics [48] etc.

The remainder of this paper is organized as follows: in section 2 the used dataset is described as well as the steps of the proposed method, in section 3 the experimental results are presented and finally in section 4 some conclusions are discussed.

2. Materials and Methods

In this section, an extensive presentation of the seismic data used, their post-processing, as well as the computational rule construction method applied for the effective classification of earthquakes will be given.

2.1. The Dataset Employed

In this study, we utilized open data provided by the NSF Seismological Facility for the Earth Consortium (SAGE), specifically accessed through the http://ds.iris.edu/ieb/index.html?format=text&nodata=404&starttime=2010-01-01&endtime=2010-12-31&orderby=time-desc&src=iris&limit=1000&maxlat=86.99&minlat=-84.16&maxlon=180.00&minlon=-180.00&zm=1&mt=ter Interactive Earthquake Browser. The choice of NSF data was motivated by its enhanced functionality. Although the GEOFON program offers similar information, it imposes a restriction on the maximum number of earthquakes retrievable per query (1000 events), thereby limiting the temporal scope of data extraction. This constraint is particularly significant, as nearly 1000 seismic events may occur globally within a single day. The NSF SAGE Facility is recognized as a reliable data repository, having been certified by the CoreTrustSeal Standards and Certification Board.

2.2. Dataset Description

We download and analyzed 1,035,971 earthquakes events, from IEB, between 2004-2011 (starting date 2004/04/01 - ending date 2011/03/16), count days 2487. The dataset includes the following variables: Year, Month, Day, Time, Latitude (Lat), Longitude (Lon), Depth, Mag, Region, and Timestamp. We utilize the coordinates for (latitude 21°-79° & Longitude 33°-176°), select the magnitude measurements (range 1.0-10.0), and select by default for Depth Range, in order to use all available depths. From our dataset, the daily average is 41,655 earthquakes, the minimum earthquakes in a day, was 33, on 2005/03/02, and the maximum in a day was 2326, on 2011/03/11 (Tohoku earthquake 9.1 mag). The regions encompassed within these geographical coordinates are 255, while globally approximately 708 seismogenic zones have been recorded since 1970, based on data from the Interactive Earthquake Browser. The features of the original dataset are show in Table 3.

Table 3. The description of the original dataset.

FEATURE	RANGE
YEAR	2004-2011
MONTH	1-12
DAY	1-31
TIME	00:00:00 - 23:59:59
LATITUDE	21.00°-79.00°
LONGITUDE	33.00°-176°
DEPTH	0.00-800.00
MAGNITUDE	1.0-10.0
TIMESTAMP	

2.3. Pre-processing steps

The initial dataset was enhanced after processing the original data. An important role in this process was played by the d_c distance, which will be used to determine when two earthquakes are close in distance. The proposed value in the present implementation was d_c =5km. Based on the above distance, the following features were added to the datasets:

1. Number of seismic events at a distance less than the d_c that have occurred in a previous time.
2. Average magnitude of seismic events at a distance less than d_c, which have preceded the current seismic event in time.
3. The greatest magnitude of a seismic event recorded in the past, at a distance of less than d_c from the current seismic event.
4. The time in seconds since the largest seismic event that has occurred in the past at a distance less than d_c from the current seismic event.
5. The distance in kilometers from the largest seismic event that has occurred in the past at a distance less than d_c from the current seismic event.

In addition, the size of the seismic recordings was divided into two large classes:

- The first class contains all the events with magnitude ≤ 3
- The second class contains all the remaining events, with magnitude>3.

2.4. The used method

The method that constructs classification rules was initially presented in [49] and an implementation in C++ was suggested later by Anastasopoulos et al [50]. The method can produce a series of classification rules in a human readable form in order to effectively classify patterns in predefined classes. This method does not require prior knowledge of the specifics of a dataset and can furthermore discover hidden correlations between the features of the dataset or even isolate only those features that play the most significant role in the successful classification of patterns. The method has been successfully applied in many cases, such as pollution detection [51], network problems [52] etc. The main steps of this method have as follows:

1. **Initialization step**.
   (a) **Set** as N_c the number of chromosomes in the genetic population and as N_g the maximum number of allowed generations.
   (b) **Set** as p_s the selection rate and as p_m the mutation rate.
   (c) **Initialize** the chromosomes $g_i,\ i = 1, \ldots, N_c$ as sets of random integers.
   (d) **Set** $k = 0$, the generation counter.
2. **Fitness calculation step**.
   (a) **For** $i = 1, \ldots, N_c$ **do**
      i. **Create** a classification program C_i for the chromosome g_i. The BNF grammar used for this production is outlined in Figure .
      ii. **Apply** the classification program to the train set of the objective problem and set as f_i the classification error for this program. The variable f_i denotes the fitness of the chromosome g_i.
   (b) **End For**
3. **Application of genetic operations**.
   (a) **Selection**: The chromosomes are sorted with respect to their fitness values. The $(1 - p_s) \times N_c$ of them with the lowest fitness values will be copied to the next generation without changes. The remaining will be substituted by chromosomes produced during crossover and mutation.
   (b) **Crossover**: In this procedure a series of offspring are produced, through a process similar to biological crossover in nature. For every couple $(\tilde{z}, \widehat{w})$ of produced offsprings two chromosomes are selected from the current population, using tournament selection. Subsequently, these chromosomes will produce the set $(\tilde{z}, \widehat{w})$ with the application of one-point crossover. A graphical example of the one-point crossover method is outlined in Figure 3.
   (c) **Mutation**: In this step, for every element $g_{i,j}$ of each chromosome g_i a random number $r \in [0, 1]$ is selected. If $r \leq p_m$, then the element $g_{i,j}$ is substituted by another random integer.
4. **Termination check step**.
   (a) **Set** $k = k + 1$
   (b) **If** $k \leq N_g$ **then go to** Fitness Calculation step **else** terminate.

```
<S>::=if(<BEXPR>) CLASS=0 else CLASS=1 (0)
<BEXPR>::=<XLIST><BOOLOP><EXPR>           (0)
         |!(<BEXPR>)                      (1)
         |<XLIST><BOOLOP><EXPR>&<BEXPR>   (2)
         |<XLIST><BOOLOP><EXPR>|<BEXPR>   (3)
<BOOLOP>::= >                             (0)
          | >=                            (1)
          | <                             (2)
          | <=                            (3)

<EXPR>::= (<EXPR><BINARYOP><EXPR>)        (0)
        | <FUNCTION>(<EXPR>)              (1)
        | <TERMINAL>                      (2)
<BINARYOP>::= +                           (0)
            | -                           (1)
            | *                           (2)
            | /                           (3)
<FUNCTION>::= sin | cos | exp | log       (0-3)
<TERMINAL>::= <XLIST>                     (0)
            | <DIGITLIST>.<DIGITLIST>     (1)
            | (-<DIGITLIST>.<DIGITLIST>)  (2)
 <XLIST>::= x1 | x2 | ...|xD              (0-D-1)
<DIGITLIST>::=<DIGIT>                     (0)
             | <DIGIT><DIGIT>             (1)
             | <DIGIT><DIGIT><DIGIT>      (2)
<DIGIT>::= 0 | 1 | 2 | 3 | 4 |5 |6 |7 |8 |9    (0-9)
```

Figure 2. The grammar of the proposed method.

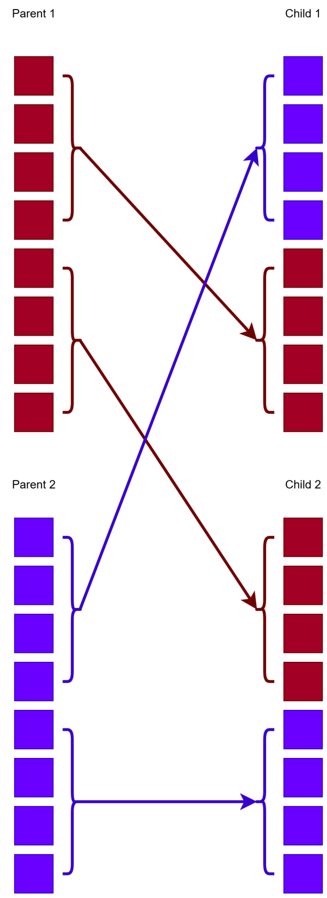

Figure 3. An example of the one - point crossover method.

3. Experiments

The code used in the experiments was implemented using the C++ programming language as well as the freely available optimization tool [53]that can be downloaded from https://github.com/itsoulos/GlobalOptimus.git (accessed on 12 October 2025). Also, the WEKA programming tool [54] was employed. Every experiment was repeated 30 times, using different seed for the random generator each time and the average classification error was reported. The validation of the experimental results was performed using the ten-fold cross validation method. The values of parameters for the used methods are shown in Table 4.

Table 4. The values for the experimental settings.

PARAMETER	MEANING	VALUE
N_c	Chromosomes	500
N_g	Generations	2000
p_s	Selection rate	0.1
p_m	Mutation rate	0.05
d_c	Critical Distance	5km
H	Number of weights	10

3.1. Experimental results

The following machine learning methods were used in the conducted experiments as denoted in Table 5.

1. RBF, a Radial Basis Function (RBF) network [55,56] was incorporated with 10 weights.
2. MLP, an artificial neural network [57,58] with 10 processing nodes. The neural network was trained using the BFGS optimization method [59].

3. BAYES, where the Naive Bayes method [60] was utilized on the dataset.
4. The column BAYESNN represents the incorporation of the Bayesian optimizer as implemented in BayesOpt [61] library to train a neural network with $H = 10$ processing nodes. The code can be downloaded from https://github.com/rmcantin/bayesopt (accessed on 12 October 2025).
5. NNC, used to represent the application of Neural Network Construction method [62], which creates the architecture of neural networks using Grammatical Evolution.
6. GENCLASS, represents the application of the proposed method.

Table 5. Experimental results on the obtained datasets using a series of machine learning methods.

YEAR	RBF	MLP	BAYES	BAYESNN	NNC	GENCLASS
2004	25.85%	26.55%	26.59%	28.23%	22.12%	18.92%
2005	29.49%	28.10%	29.09%	28.73%	23.35%	19.56%
2006	26.69%	27.67%	26.17%	25.50%	24.51%	17.19%
2007	25.02%	26.93%	25.83%	24.89%	22.77%	18.14%
2008	28.17%	29.97%	29.88%	28.53%	25.42%	19.51%
2009	26.80%	25.90%	26.09%	25.08%	21.07%	17.81%
2010	28.08%	29.43%	28.00%	26.22%	23.31%	19.89%
2011	27.67%	35.83%	28.25%	28.56%	25.73%	20.97%
AVERAGE	**27.22%**	**28.80%**	**27.49%**	**26.97%**	**23.54%**	**19.00%**

Also, the classification error for all methods per year is presented in Figure 4.

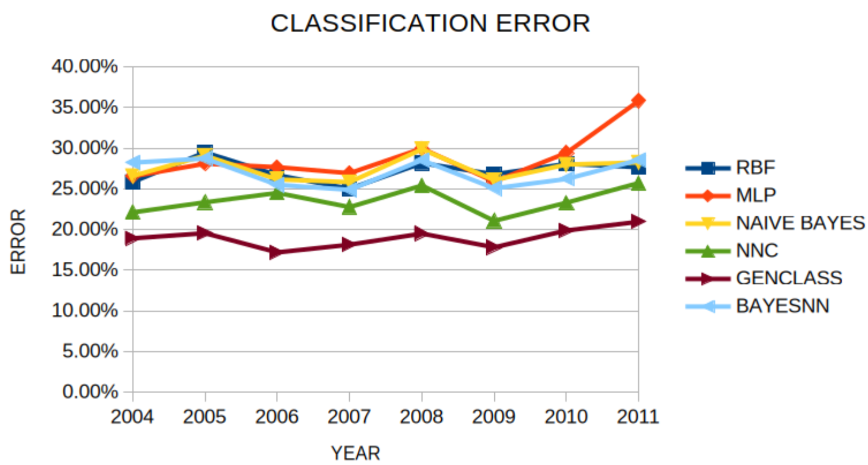

Figure 4. The classification error of each machine learning method per year.

Table 5 reports yearly classification error rates (2004-2011) for six machine-learning models. The proposed GENCLASS method consistently ranks first every year, achieving the lowest average error of 19.00%, which corresponds to an overall accuracy of about 81%. The runner-up is NNC with an average error of 23.54%, so GENCLASS reduces error by roughly 4.54 percentage points on average about a 19% relative reduction versus the closest competitor. The annual margin of GENCLASS over NNC ranges from 3.20 to 7.32 points, peaking in 2006. Beyond accuracy, GENCLASS also shows the smallest across-year variability (standard deviation ≈ 1.16), indicating stable performance throughout the evaluation period. In contrast, MLP exhibits the highest mean error (28.80%) and the largest variability, with a pronounced deterioration in 2011. Overall, these results support the article's claim that Grammatical Evolution is particularly effective for this seismic classification problem, as GENCLASS delivers both the lowest average error and the most consistent year-to-year behavior.

The R-based significance analysis reveals a clear overall difference among models (Friedman $p = 1.21 \times 10^{-5}$, very strong evidence), with pairwise tests indicating where these differences lie (Figure 5). GENCLASS outperforms MLP with $p = 7.52 \times 10^{-4}$ (extremely significant) and BAYES with $p = 0.006$ (highly significant), and the comparison with RBF is also statistically significant at $p = 0.0125$. In contrast, GENCLASS vs BAYESNN ($p = 0.0516$) and GENCLASS vs NNC ($p = 0.8937$) are not statistically significant at the 0.05 threshold. Overall, while the omnibus test confirms marked differences across models, the superiority of the proposed method is clearly supported against MLP, BAYES, and RBF, whereas the contrasts with BAYESNN and NNC do not reach conventional significance.

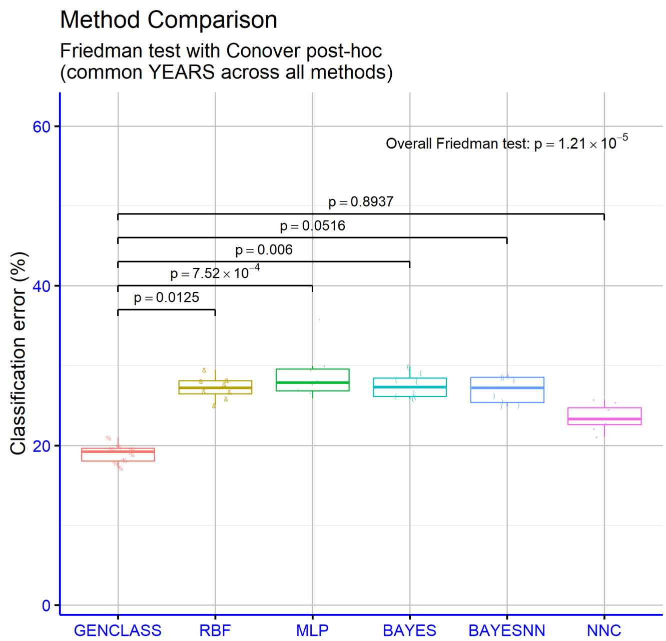

Figure 5. Statistical tests performed on the experimental results using the variety of machine learning methods.

3.2. *Experiments with the critical distance d_c*

The critical distance d_c was used to generate the final datasets as the distance between seismic events. In order to determine the correlation of this parameter with the experimental results produced, another experiment was conducted where this distance ranged from 2 to 20km. In this experiment, the proposed classification rule generation technique was used. The experimental results from this experiment are presented in Table 6.The results indicate that the critical distance d_c influences performance, but the effect size is small. The average classification error decreases slightly as the distance increases: 19.08% for $d_c = 2km$, 19.00% for 5km, 18.82% for 10km, and 18.71% for 20km. The gap between the worst and best setting is 0.37 percentage points, i.e., about a 1.9% relative error reduction, so the overall advantage of the 20km setting is real but modest. Year-by-year, $d_c = 20km$ achieves the lowest error in 4 out of 8 years (2004, 2005, 2009, 2010), $d_c = 5km$ is best in 2008, $d_c = 10km$ in 2007, and $d_c = 2km$, indicating no monotonic or universally optimal choice across years but a mild preference toward larger distances. Regarding stability, $d_c = 2km$ shows the narrowest range over time (17.51%–20.70%), whereas $d_c = 20km$ exhibits the largest spread mainly due to 2011 (16.97%–21.78%). Overall, the method benefits marginally from increasing d_c, with 20km yielding the lowest average error, but the choice should also consider temporal stability and yearly idiosyncrasies, since the per-year optimum shifts across settings.

Table 6. Experiments with the GENCLASS method and different values of critical distance d_c

YEAR	$d_c = 2km$	$d_c = 5km$	$d_c = 10km$	$d_c = 20km$
2004	18.69%	18.92%	18.54%	18.44%
2005	18.47%	19.56%	18.26%	17.56%
2006	17.51%	17.19%	17.38%	18.48%
2007	20.03%	18.14%	18.06%	18.20%
2008	20.70%	19.51%	20.69%	20.14%
2009	17.90%	17.81%	17.39%	16.97%
2010	19.21%	19.89%	19.32%	18.08%
2011	20.09%	20.97%	20.91%	21.78%
AVERAGE	**19.08%**	**19.00%**	**18.82%**	**18.71%**

The significance analysis across critical distance settings dc provides no statistical evidence of an effect on classification error (Figure 6). The omnibus Friedman test is non-significant ($p = 0.4153$), indicating no systematic differences across d_c levels. Pairwise contrasts corroborate this: $d_c = 2km$ vs 5km ($p = 0.9803$), 10km ($p = 0.5274$), and 20km ($p = 0.5274$) are not significant, nor are $d_c = 5km$ vs 20km ($p = 0.7675$) and $d_c = 10km$ km vs 20km ($p = 1$). The $d_c = 5km$ vs 10km comparison returned $p =$ NA, which typically reflects a degenerate post-hoc case (e.g., complete rank ties or identical values per block) and does not alter the overall conclusion. In sum, within the 2–20km range, dc does not yield statistically significant performance differences for the proposed method, so this hyperparameter may be selected based on practical or stability considerations rather than expected accuracy gains.

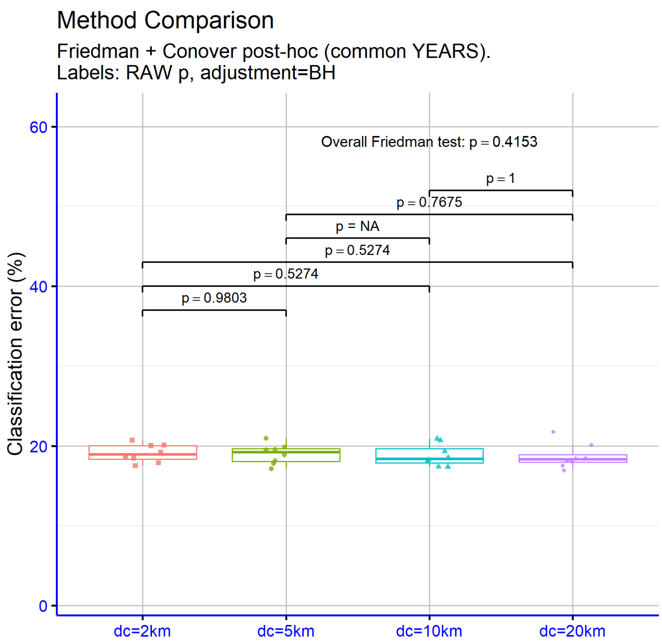

Figure 6. Statistical comparison for the results obtained by the application of the GENCLASS method, using a variety of values for the critical distance d_c.

3.3. Experiments with the number of generations N_g

An additional experiment was conducted to verify the stability of the proposed classification rule generation technique. In this experiment, the maximum number of generations ranged from 200 to 2000 and the experimental results per year are presented in Table 7. In this table, the results show a clear downward trend in classification error as the maximum number of generations increases. The average error decreases from 20.20% $(N_g = 200)$ to

19.58% $\left(N_g = 500\right)$, 19.25% $\left(N_g = 1000\right)$ and 19.00% $\left(N_g = 2000\right)$, i.e., a total gain of 1.20 percentage points or roughly a 6% relative reduction compared to $N_g = 200$. Improvements exhibit diminishing returns: the largest drop is from 200 to 500 (−0.62), followed by 500 to 1000 (-0.33) and 1000 to 2000 (-0.25). On a per-year basis, $N_g = 2000$ is best in six out of eight years (2006-2011 except 2004-2005), while in 2004-2005 the minimum error occurs at $N_g = 1000$. The 2005 value for $N_g = 2000$ is noticeably higher than the other settings, which may reflect stochastic variability or model over-specialization for that year's data. The spread across years is broadly similar across settings, so the main benefit of increasing generations is a lower mean error rather than a dramatic change in variability. In practice, the 1000-2000 range offers the strongest performance, $N_g = 1000$ comes very close to $N_g = 2000$ (0.25 points apart) and is attractive under tighter computational budgets, whereas $N_g = 2000$ yields the lowest average error when runtime is not a constraint.

Table 7. Experiments with the GENCLASS method and different values for the number of generations N_g

YEAR	$N_g = 200$	$N_g = 500$	$N_g = 1000$	$N_g = 2000$
2004	19.85%	19.14%	18.76%	18.92%
2005	17.94%	17.80%	17.63%	19.56%
2006	20.33%	18.01%	17.95%	17.19%
2007	20.51%	19.91%	19.37%	18.14%
2008	21.83%	21.31%	20.16%	19.51%
2009	18.61%	18.25%	18.15%	17.81%
2010	20.59%	20.57%	20.55%	19.89%
2011	21.90%	21.63%	21.39%	20.97%
AVERAGE	**20.20%**	**19.58%**	**19.25%**	**19.00%**

The significance levels in Figure 7 indicate that the maximum number of generations has an overall effect on performance, as evidenced by a strongly significant Friedman test $\left(p = 6.28 \times 10^{-4}\right)$. In pairwise terms, moving from $N_g = 200$ to $N_g = 1000$ yields a statistically significant error reduction ($p = 0.022$), and the contrast between $N_g = 200$ and $N_g = 2000$ is even more significant ($p = 0.0067$), confirming that a low generation budget underperforms relative to higher budgets. By contrast, $N_g = 200$ vs $N_g = 500$ is not significant ($p = 0.8161$), nor are $N_g = 500$ vs $N_g = 2000$ ($p = 0.4306$) and $N_g = 1000$ vs $N_g = 2000$ ($p = 0.9803$), suggesting diminishing returns beyond roughly 1000 generations. The $N_g = 500$ vs $N_g = 1000$ comparison returned $p =$ NA, typically due to a degenerate post-hoc scenario (e.g., complete rank ties or identical per-block values) and does not alter the main conclusion. Overall, increasing generations above 200 significantly improves performance, with gains saturating around 1000 generations and no clear statistical advantage of 2000 over 1000.

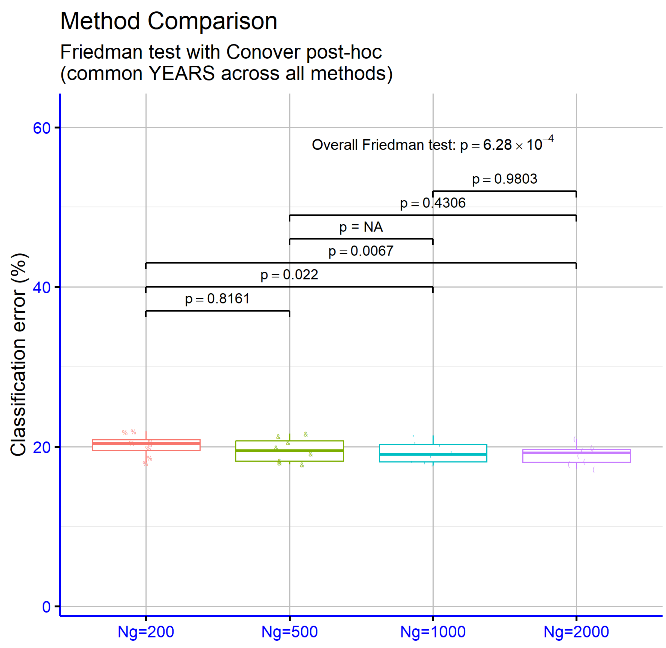

Figure 7. Statistical comparison for the results obtained by the usage of the GENCLASS method, using different values for the maximum number of generations N_g.

4. Conclusions

This study introduces an innovative methodology for earthquake classification employing various machine learning methods, including RBF, MLP, BAYES, NNC, along with the proposed GENCLASS approach. The analysis covers seismic data recorded between 2004-2011, within the geographical bounds of latitude 21°-79° and longitude 33°-176°. Our process first involved classifying the size of the seismic events into two large classes: The first class contains all the events with magnitude ≤ 3. The second class contains all the remaining events, with magnitude>3, followed by magnitude prediction based on the classified data.

To construct the final datasets, the critical distance d_c defined as the distance between seismic events was applied. The results indicate that the proposed method benefits slightly from increasing d_c, with a value of 20km yielding the optimal average error, also we Set as N_c the number of chromosomes in the genetic population. Furthermore, an additional experiment was conducted to evaluate the stability of the proposed generations mechanism N_g. In this test, the maximum number of generations was varied from 200-2000, indicated that the maximum number of generations has an overall effect on performance. Among all evaluated methods, GENCLASS consistently achieved the best performance each year, attaining the lowest average error rate of 19%, corresponding to an overall classification accuracy of 81%.

The main challenges encountered in this research were related to the large volume of data, exceeding one million records, and the adaptation of such data into a reliable classification system. In general, seismology pay stil keep many of its secrets well guarded, but not indefinitely, with the rapid advancement of machine learning and artificial intelligence techniques, these mysteries are gradually being unraveled. It is anticipated that future developments in these fields will further contribute to decoding the complex patterns underlying seismic activity.

Author Contributions: Conceptualization, C.K. and I.G.T.; methodology, C.K.; software, I.G.T.; validation, C.S., V.C.; formal analysis, V.C.; investigation, C.K.; resources, C.S.; data curation, C.K..;

writing original draft preparation, C.K.; writing review and editing, I.G.T.; visualization, V.C.; supervision, C.S.; project administration, C.S.; funding acquisition, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been financed by the European Union : Next Generation EU through the Program Greece 2.0 National Recovery and Resilience Plan , under the call RESEARCH-CREATE-INNOVATE, project name "iCREW: Intelligent small craft simulator for advanced crew training using Virtual Reality techniques" (project code:TAEDK-06195).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mosegaard, K., Tarantola, A., Lee, W. H. K., Jennings, P., Kisslinger, C., & Kanamori, H. (2002). International Handbook of Earthquake and Engineering Seismology: Part A. International Geophysics Series, 81, 237-265.
2. Online Archive of California, Guide to the Papers of Charles F. Richter, 1839-1984. https://oac.cdlib.org/findaid/ark:/13030/kt787005jn/admin/
3. Encyclopedia.com, Mercalli, Giuseppe. https://www.encyclopedia.com/people/science-and-technology/environmental-studies-biographies/giuseppe-mercalli#:~:text=Mercalli.
4. GeoVera, A Journey Through Time: The History of the Richter Scale, 2023. https://geovera.com/2023/04/27/history-richter-scale/
5. National Academies of Sciences, Engineering, and Medicine. 2003. Living on an Active Earth: Perspectives on Earthquake Science. Washington, DC: The National Academies Press. https://doi.org/10.17226/10493.https://nap.nationalacademies.org/read/10493/chapter/1#ii
6. Hutchison, Allie. How machine learning might unlock earthquake prediction. 2023. MIT Technology Review. https://www.technologyreview.com/2023/12/29/1084699/machine-learning-earthquake-prediction-ai-artificial-intelligence/
7. Zou, J., Han, Y., & So, S. S. (2008). Overview of artificial neural networks. Artificial neural networks: methods and applications, 14-22.
8. Lakkos, S., Hadjiprocopis, A., Comley, R., & Smith, P. (1994, September). A neural network scheme for earthquake prediction based on the seismic electric signals. In Proceedings of IEEE Workshop on Neural Networks for Signal Processing (pp. 681-689). IEEE.
9. Mousavi, S. M., & Beroza, G. C. (2020). A machine-learning approach for earthquake magnitude estimation. Geophysical Research Letters, 47(1), e2019GL085976.
10. Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nature communications, 11(1), 3952.
11. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.
12. Ross, Z. E., Meier, M. A., & Hauksson, E. (2018). P wave arrival picking and first-motion polarity determination with deep learning. Journal of Geophysical Research: Solid Earth, 123(6), 5120-5129.
13. Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329.
14. Negarestani, A., Setayeshi, S., Ghannadi-Maragheh, M., & Akashe, B. (2002). Layered neural networks based analysis of radon concentration and environmental parameters in earthquake prediction. Journal of environmental radioactivity, 62(3), 225-233.
15. Tan, Y. J., Waldhauser, F., Ellsworth, W. L., Zhang, M., Zhu, W., Michele, M., ... & Segou, M. (2021). Machine-learning-based high-resolution earthquake catalog reveals how complex fault structures were activated during the 2016–2017 central Italy sequence. The Seismic Record, 1(1), 11-19.
16. Caterini, A. L., & Chang, D. E. (2018). Recurrent neural networks. In Deep neural networks in a mathematical framework (pp. 59-79). Cham: Springer International Publishing.
17. Panakkat, A., & Adeli, H. (2007). Neural network models for earthquake magnitude prediction using multiple seismicity indicators. International journal of neural systems, 17(01), 13-33.

18. Asim, K. M., Martínez-Álvarez, F., Basit, A., & Iqbal, T. (2017). Earthquake magnitude prediction in Hindukush region using machine learning techniques. Natural Hazards, 85(1), 471-486.
19. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
20. Hsu, Y. F., Zaliapin, I., & Ben-Zion, Y. (2024). Informative modes of seismicity in nearest-neighbor earthquake proximities. Journal of Geophysical Research: Solid Earth, 129(3), e2023JB027826.
21. Bayliss, K., Naylor, M., & Main, I. G. (2019). Probabilistic identification of earthquake clusters using rescaled nearest neighbour distance networks. Geophysical Journal International, 217(1), 487-503.
22. Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. Geophysical Research Letters, 44(18), 9276-9282.
23. Rouet-Leduc, B., Hulbert, C., & Johnson, P. A. (2019). Continuous chatter of the Cascadia subduction zone revealed by machine learning. Nature Geoscience, 12(1), 75-79.
24. Saad, O. M., Chen, Y., Savvaidis, A., Fomel, S., Jiang, X., Huang, D., ... & Chen, Y. (2023). Earthquake forecasting using big data and artificial intelligence: A 30-week real-time case study in China. Bulletin of the Seismological Society of America, 113(6), 2461-2478.
25. Corbi, F., Sandri, L., Bedford, J., Funiciello, F., Brizzi, S., Rosenau, M., & Lallemand, S. (2019). Machine learning can predict the timing and size of analog earthquakes. Geophysical Research Letters, 46(3), 1303-1311.
26. O'Neill, M., & Ryan, C. (2002). Grammatical evolution. IEEE Transactions on Evolutionary Computation, 5(4), 349-358.
27. Kramer, O. (2017). Genetic algorithms. In Genetic algorithm essentials (pp. 11-19). Cham: Springer International Publishing.
28. J. W. Backus. The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference. Proceedings of the International Conference on Information Processing, UNESCO, 1959, pp.125-132.
29. C. Ryan, J. Collins, M. O'Neill, Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds) Genetic Programming. EuroGP 1998. Lecture Notes in Computer Science, vol 1391. Springer, Berlin, Heidelberg, 1998.
30. M. O'Neill, M., C. Ryan, Evolving Multi-line Compilable C Programs. In: Poli, R., Nordin, P., Langdon, W.B., Fogarty, T.C. (eds) Genetic Programming. EuroGP 1999. Lecture Notes in Computer Science, vol 1598. Springer, Berlin, Heidelberg, 1999.
31. A. Brabazon, M. O'Neill, Credit classification using grammatical evolution, Informatica **30.3**, 2006.
32. S. Şen, J.A. Clark. A grammatical evolution approach to intrusion detection on mobile ad hoc networks, In: Proceedings of the second ACM conference on Wireless network security, 2009.
33. L. Chen, C.H. Tan, S.J. Kao, T.S. Wang, Improvement of remote monitoring on water quality in a subtropical reservoir by incorporating grammatical evolution with parallel genetic algorithms into satellite imagery, Water Research **42**, pp. 296-306, 2008.
34. J. I. Hidalgo, J. M. Colmenar, J.L. Risco-Martin, A. Cuesta-Infante, E. Maqueda, M. Botella,J. A. Rubio, Modeling glycemia in humans by means of Grammatical Evolution, Applied Soft Computing **20**, pp. 40-53, 2014.
35. J. Tavares, F.B. Pereira, Automatic Design of Ant Algorithms with Grammatical Evolution. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds) Genetic Programming. EuroGP 2012. Lecture Notes in Computer Science, vol 7244. Springer, Berlin, Heidelberg, 2012.
36. M. Zapater, J.L. Risco-Martín, P. Arroba, J.L. Ayala, J.M. Moya, R. Hermida, Runtime data center temperature prediction using Grammatical Evolution techniques, Applied Soft Computing **49**, pp. 94-107, 2016.
37. C. Ryan, M. O'Neill, J.J. Collins, Grammatical evolution: Solving trigonometric identities, proceedings of Mendel. Vol. 98. 1998.
38. A.O. Puente, R. S. Alfonso, M. A. Moreno, Automatic composition of music by means of grammatical evolution, In: APL '02: Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and applications July 2002 Pages 148–155.
39. Lídio Mauro Limade Campo, R. Célio Limã Oliveira,Mauro Roisenberg, Optimization of neural networks through grammatical evolution and a genetic algorithm, Expert Systems with Applications **56**, pp. 368-384, 2016.
40. K. Soltanian, A. Ebnenasir, M. Afsharchi, Modular Grammatical Evolution for the Generation of Artificial Neural Networks, Evolutionary Computation **30**, pp 291–327, 2022.
41. I. Dempsey, M.O' Neill, A. Brabazon, Constant creation in grammatical evolution, International Journal of Innovative Computing and Applications **1** , pp 23–38, 2007.
42. E. Galván-López, J.M. Swafford, M. O'Neill, A. Brabazon, Evolving a Ms. PacMan Controller Using Grammatical Evolution. In: , et al. Applications of Evolutionary Computation. EvoApplications 2010. Lecture Notes in Computer Science, vol 6024. Springer, Berlin, Heidelberg, 2010.
43. N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, M. O'Neill, Evolving levels for Super Mario Bros using grammatical evolution, 2012 IEEE Conference on Computational Intelligence and Games (CIG), 2012, pp. 304-31.
44. D. Martínez-Rodríguez, J. M. Colmenar, J. I. Hidalgo, R.J. Villanueva Micó, S. Salcedo-Sanz, Particle swarm grammatical evolution for energy demand estimation, Energy Science and Engineering **8**, pp. 1068-1079, 2020.
45. N. R. Sabar, M. Ayob, G. Kendall, R. Qu, Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems, IEEE Transactions on Evolutionary Computation **17**, pp. 840-861, 2013.

46. C. Ryan, M. Kshirsagar, G. Vaidya, G. et al. Design of a cryptographically secure pseudo random number generator with grammatical evolution. Sci Rep **12**, 8602, 2022.
47. P.J. Pereira, P. Cortez, R. Mendes, Multi-objective Grammatical Evolution of Decision Trees for Mobile Marketing user conversion prediction, Expert Systems with Applications **168**, 114287, 2021.
48. F. Castejón, E.J. Carmona, Automatic design of analog electronic circuits using grammatical evolution, Applied Soft Computing **62**, pp. 1003-1018, 2018.
49. Tsoulos, I. G. (2020). Creating classification rules using grammatical evolution. International Journal of Computational Intelligence Studies, 9(1-2), 161-171.
50. Anastasopoulos, N.; Tsoulos, I.G.; Tzallas, A. GenClass: A parallel tool for data classification based on Grammatical Evolution. SoftwareX 2021, 16, 100830.
51. Spyrou, E. D., Stylios, C., & Tsoulos, I. (2023). Classification of CO Environmental Parameter for Air Pollution Monitoring with Grammatical Evolution. Algorithms, 16(6), 300.
52. Margariti, S. V., Tsoulos, I. G., Kiousi, E., & Stergiou, E. (2024). Traffic Classification in Software-Defined Networking Using Genetic Programming Tools. Future Internet, 16(9), 338.
53. I.G. Tsoulos, V. Charilogis, G. Kyrou, V.N. Stavrou, A. Tzallas, Journal of Open Source Software **10**, 7584, 2025.
54. M. Hall, F. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update. ACM SIGKDD explorations newsletter **11**, pp. 10-18, 2009.
55. J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation **3**, pp. 246-257, 1991.
56. G.A. Montazer, D. Giveki, M. Karami, H. Rastegar, Radial basis function neural networks: A review. Comput. Rev. J **1**, pp. 52-74, 2018.
57. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11).
58. Suryadevara, S., & Yanamala, A. K. Y. (2021). A Comprehensive Overview of Artificial Neural Networks: Evolution, Architectures, and Applications. Revista de Inteligencia Artificial en Medicina, 12(1), 51-76.
59. M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.
60. G.I. Webb, E. Keogh, R. Miikkulainen, Naïve Bayes, Encyclopedia of machine learning **15**, pp. 713-714, 2010.
61. Ruben Martinez-Cantin, BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits. Journal of Machine Learning Research, 15(Nov):3735--3739, 2014.
62. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing **72**, pp. 269-277, 2008.