Llácer, D. [et al.]. Random forest parameterization for earthquake catalog generation. A: International Conference on Machine Learning, Optimization, and Data Science. "Machine Learning, Optimization, and Data Science, 6th International Conference, LOD 2020: Siena, Italy, July 19-23, 2020: revised selected papers, part I". Berlín: Springer, 2020, p. 233-243. ISBN 978-3-030-64583-0. The final authenticated version is available online at https://doi.org/DOI 10.1007/978-3-030-64583-0_22

Random Forest Parameterization for Earthquake Catalog Generation

David Llácer[1], Beatriz Otero[1], Rubén Tous[1], Marisol Monterrubio-Velasco[2], José Carlos Carrasco-Jiménez[2], and Otilio Rojas[2,3]

[1] Universitat Politécnica de Catalunya. Barcelona, Spain
[2] Barcelona Supercomputing Center. Barcelona, Spain
[3] Universidad Central de Venezuela, Facultad de Ciencias, Caracas, Venezuela

Abstract. An earthquake is the vibration pattern of the Earth's crust induced by the sliding of geological faults. They are usually recorded for later studies. However, strong earthquakes are rare, small-magnitude events may pass unnoticed and monitoring networks are limited in number and efficiency. Thus, earthquake catalog are incomplete and scarce, and researchers have developed simulators of such catalogs. In this work, we start from synthetic catalogs generated with the TREMOL-3D software. TREMOL-3D is a stochastic-based method to produce earthquake catalogs with different statistical patterns, depending on certain input parameters that mimics physical parameters. When an appropriate set of parameters are used, TREMOL-3D could generate synthetic catalogs with similar statistical properties observed in real catalogs. However, because of the size of the parameter space, a manual searching becomes unbearable. Therefore, aiming at increasing the efficiency of the parameter search, we here implement a Machine Learning approach based on Random Forest classification, for an automatic parameter screening. It has been implemented using the machine learning Python's library Sci-Kit Learn.

Keywords: Earthquakes· Synthetic Catalogs· Machine Learning· Random Forest

1 Introduction

The study of earthquakes helps us to estimate their possible occurrence and observed magnitudes, and allow assessing potential actions to lessen their impact and terrible effects [3][10][12]. For instance, conventional seismic hazard analyses of a region start from a local earthquake catalog. Unfortunately, seismic catalogs cover a limited time span compared to some long recurrence earthquake intervals, and our monitoring and processing resources are still limited. Thus, these catalogs may present deficiencies. Earthquakes can be studied from either physical or statistical points of view [5]. The model proposed in [7], and named TREMOL, aims at generating synthetic earthquake catalogs by using the Fiber Bundle Model (FBM), which is a stochastic modeling technique that

requires parameter tuning. Under appropiate parameterization, TREMOL reproduces an earthquake mainshock and the following aftershock sequence until the modeled seismicity ceases, as observed in nature. The collection of these events, with associated magnitudes, represents a TREMOL synthetic catalog. In [6], some machine learning (ML) techniques were developed to assist in the TREMOL screening parameter. These techniques use as a basis a real seismic catalog that record the full earthquake sequence and observed magnitudes, along with the epicentral and hipocentral coordinates, origin times, among other additional information. With this data, they fed a ML system implementing Support Vector Machine, Flexible Discriminant Analysis and Random Forest, in order to determine the best combination of TREMOL input parameters, and which statistical features of a catalog were more important to analyze synthetic and real sequences. In [6], the ML technique yielding the best results was Random Forest, that exhibits the lowest mean errors on the statistical features that allow comparing a synthetic to a real catalog. Thus, the work in [6] motivates our current implementation that extends the efforts for parameter screening with a more complex database. Following [6], we here use exclusively Random Forest to optimize TREMOL input parameters.

The main contribution of this work is the use of a more general dataset with respect to [6], and the development of a flexible and automatic tool based on the Random Forest classification technique for parameter searching. These topics are discussed in Sections 2 and 3. Basically, the previously used data in [6] was generated with a TREMOL-2D version [7]. Simulations on this model take place on a 2D horizontal representation of the study region, where depth is omitted, and this fact may limit the modeling of some mainshock-aftershock sequences. A first 3D TREMOL prototype was developed and tested in [9], that shows potential improvements to solve complex mainshock-aftershock scenarios. Although, TREMOL-3D makes output seismicity more realistic, it also extends the number of parameters to be optimized, making more computationally demanding the parameter screening. Thus, in this work we start by developing a new Random Forest module for parameter screening, using as a basis the previous implementation in R language. We decide to switch to Python [11] for our new implementation, given that the new TREMOL-3D has also been programmed in Python and Julia [1]. Finally, we compare the TREMOL parameters found by an heuristic analysis in [9] with the automatically results found in this work. Details on the experimental framework and results are given in Section 4, and Section 5 states our final conclusions.

2 Data

Figure 1 shows some of the essential data recorded in a real seismic catalog. For each event, these data comprise occurrence time, hypocentral location, magnitude, among other seismological information. TREMOL models a mainshock-aftershock sequence and yields a similar catalog of synthetic earthquakes [9]. Table 1 lists some statistical parameters or features, associated to three fun-

damental empirical laws in Seismology, and commonly analyzed on earthquake catalogs [3][5]. The Gutenberg-Richter frequency-magnitude event distribution is given in terms of the linear regression parameters b_{value} and a_{value} values, and the largest and smallest magnitudes, M_{max} and M_{min}, respectively. The term M_{Bath} corresponds to the second largest event according to the Bath law, Tr_{max} estimates the rupture time of the mainshock, while the Omori's law predicts the aftershock decaying rate in terms of p_{Omori} and c_{Omori}. Among all these 8 features, c_{Omori}, given by the physical time span of the earthquake sequence, has been poorly predicted by TREMOL results in previous works [6][9]. The explanation may rely on the fact that TREMOL time formulation is dimensionless, and then c_{Omori} is low sensitive to input model parameters. The sensitivity and importance of remaining statistical features would highly depend on the modeled seismicity, and therefore, on the input TREMOL parameters.

Origin	Time	Lat. (N)	Lon. (E)	Depth (km)	Nsta	Gap	M_L	M_{L10}
01/10/95	07:51:16.12	23.66	121.41	15	12	170	5.23	4.77
02/23/95	05:19:12.15	24.22	121.66	18	25	139	6.33	5.60
03/24/95	04:13:52.14	24.62	121.86	79	24	180	5.81	5.16
04/03/95	11:54:47.98	23.95	122.31	16	25	230	5.92	5.30
04/24/95	10:04:01.52	24.65	121.65	63	28	60	5.44	5.05
06/25/95	06:59:08.74	24.58	121.69	43	36	84	6.50	5.89
07/07/95	03:04:48.29	23.88	121.10	15	28	49	5.61	5.08
07/14/95	16:52:47.89	24.36	121.76	6	23	178	5.70	4.98

Fig. 1. Example of a real earthquake catalog taken from [13].

Statistical Features	**Description**	**Value**
b_{value}	Parameter of the Gutenberg-Richter law	0.96
M_{max}	The magnitude of the mainshock	7.2
M_{Bath}	Magnitude related to the Bath's law	5.7
Tr_{max}	Maximum rupture time duration	29
a_{value}	Parameter of the Gutenberg-Richter that is proportional to the seismicity rate	5.89
M_{min}	Minimum magnitude recorded in the synthetic catalogue	3
p_{Omori}	Exponent of the Omori-Utu law typically close to 10. It is a value that controls the decay rate of aftershock activity	1
c_{Omori}	Parameter of the Omori-Utsu law.	0.5

Table 1. Relevant statistical features of an earthquake catalog and values associated to the 2010 El Mayor-Cucapah earthquake sequence.

In this work, we use as reference the real catalog corresponding to the Mw 7.2 *El Mayor-Cucapah* earthquake, a big event occurred in 2010 in Baja California, Mexico, and whose descriptive statistical features are given on the last column of Table 1. The data is taken from [4], and stored in a *csv* file for our parameter screening implementation. On the other hand, Table 2 presents the input parameters of TREMOL, which are related to the initial conditions of the fault system and given in terms of load and roughness parameters. These parameters allow to configure the initial state and the TREMOL simulation environment. The statistical coherence of an output synthetic catalog is quantified in terms of the features in Table 1. Thus, a good input parameterization should allow TREMOL to produce a synthetic catalog with similar statistical features to the ones observed in the real catalog.

Name	**Description**
S_a	Roughness size
$\overrightarrow{B}$	Vector that favors nucleation in Roughness
Θ	Load distribution percentage to diagonal cells
Π_{Asp}	Load distribution percentage of asperity cells
Π_{Faults}	Load distribution percentage of fault cells

Table 2. Input paramaters of TREMOL.

3 Methodology

3.1 Data Processing

As real data hereafter we mean the 1D vector with the 8 statistical values associated to *El Mayor-Cucapah* earthquake in Table 1. A previous time consuming simulation stage based on TREMOL under a variety of input parameters with seismological significance, we obtain the training set for our parametric search engine. The resulting training set is given by 1680 shock simulations, each one with different combinations of input parameters. With this set, we train Random Forest and finally generate a prediction, that we later compare with the real data. As mentioned before, this comparison is performed in terms of the 8 statistical features that represent and aggregate all seismic events in a catalog. An example of this training data is given by the row shown in Table 3. As one can see, the training data is stored in a 1D vector with the same format as the real one, with an additional label. The first eight columns collect the statistical features for the candidate catalog, and the last column is a label representing the input parameter, that we have chosen to make the analysis for. For example, the label value given in Table 3 corresponds to the Roughness size (S_a) used in TREMOL simulations.

b_{value}	M_{Max}	M_{Bath}	Tr_{max}	a_{value}	M_{min}	p_{Omori}	c_{Omori}	$label_{\text{input}}$
0.94	7.0	5.1	32	5.46	3	1	0.5	5

Table 3. Example of a training row data for Random Forest.

3.2 Random Forest

Random Forest consists of a large number of individual decision trees that operate as an ensemble [2]. Decision trees are the building blocks of the Random Forest algorithm, which is a non-parametric supervised learning method used for classification or regression. Decision trees break down a dataset into smaller subsets, while an associated tree is incrementally developed in the process. The final result is a tree with decision nodes and leaf nodes. Each individual tree in the Random Forest splits a class prediction, and the class with the most votes becomes our model's prediction. The fundamental concept behind Random Forest is that a large number of trees, operating as a committee, will outperform any of the individual constituent models (see Figure 2). The main reason for this attribute is that decision trees protect each other from their individual errors.

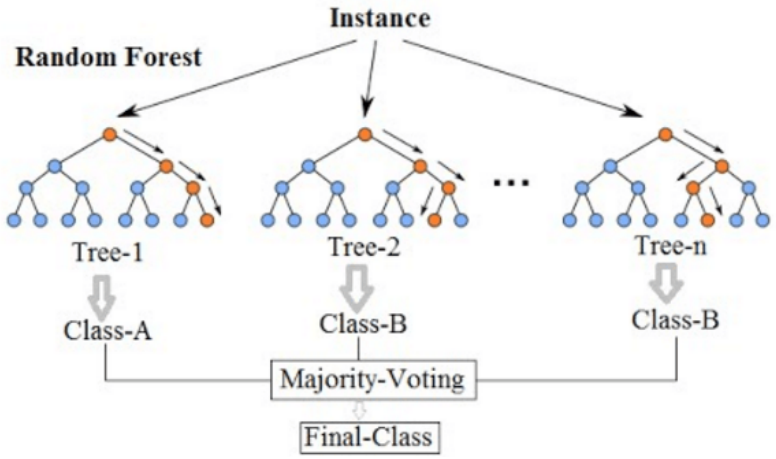


Fig. 2. Random Forest schema.

In this case we used a Random Forest classifier with 500 estimators. Random trees are also interesting because allows us to determine the importance of the features in order to see which parameters affects more when classifying. To compute the Random Forest, we used the Python library Sci-Kit Learn [8], an efficient group of tools for predictive data analysis. The input for the Random Forest algorithm is formed by two variables: a matrix ($1680x8$) with the eight

statistical values of each row and the label ($1680x1$) we selected to make the analysis with one of the five labels. We fit the model with this data and generate the Random Forest. Then, with the real data, we obtain a prediction from the trained system, that is a label. So we take the rows of the synthetic data that have the same label as the prediction and calculate the mean, obtaining an approximation of the real data. We can see the different approximations depending on the input label we selected in Section 4.

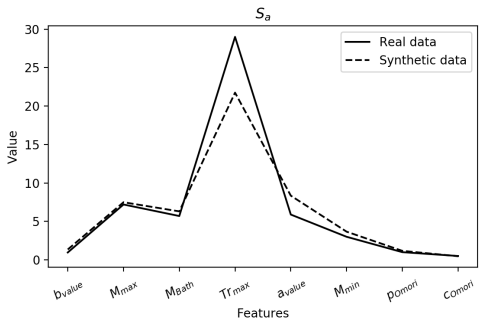
4 Experiments

We perform several experiments according to the input parameter we want to analyze. Given that our methodology considers five different input parameters, we then use five labels in our dataset. We have trained the Random Forest algorithm with our entire dataset, and later test it using the real data. In this way, the algorithm outputted labels to establish the correspondence to real data. Finally, we take the mean column vector of the synthetic catalogs having this label as the definitive catalog. These outputs clearly depend on the considered input parameter. The accuracy of our simulation outputs is assessed through comparative plots between the synthetic and real catalogs, as shown in Figure 3.

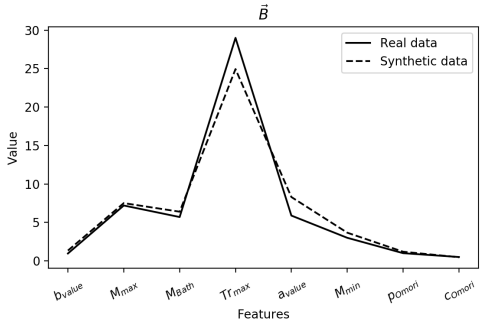
Furthermore, we can analyze the importance or sensitivity of any statistical feature with respect to each parameter of the Random Forest model, as given by Sci-Kit Learn. The quantification of this feature-to-input parameter importance is a key point of this study, because it allows to see which TREMOL input mainly affect the generation of a synthetic catalog with features, as close as possible, to the ones observed in the real case. In decision trees, every node is a condition of how to split values into a single feature, so that similar values of the dependent variable end up in the same set after the splitting. The condition is based on impurity, which in case of classification problems is Gini impurity/information gain (entropy), while the regression its variance. Thus, we can compute how much each feature contributes to decreasing the impurity, during the training of a tree.

Figure 3 displays comparisons between the real data (continuous line) and the synthetic catalog (dashed line) according to our five input parameters. The synthetic catalog was obtained computing the mean of all the synthetic data rows that contained the label given by the Random Forest algorithm when we fed it with the real data. In the X-axis we have the statistical features we mentioned in Section 2, and in the Y-axis we have their respective value. Furthermore, we can check the importance of each statistical feature with a parameter of the Random Forest model given by Sci-Kit Learn. Figure 4 plots this importance measure.

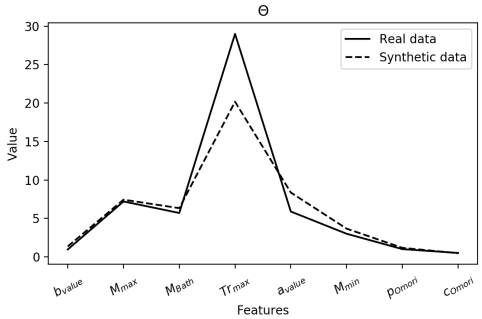
As we can observe, the most representative statistical feature is the b_{value}, followed by the M_{max} and the M_{Bath}. On the other hand, the feature with lower importance is the c_{Omori}, that actually has zero importance. We have to remark that the importance of each statistical feature was plotted for the input label that better approaches the real data, in this case the Π_{Asp} label.



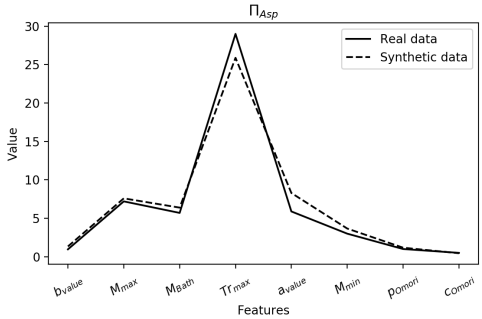
(a) Analysis with S_a label



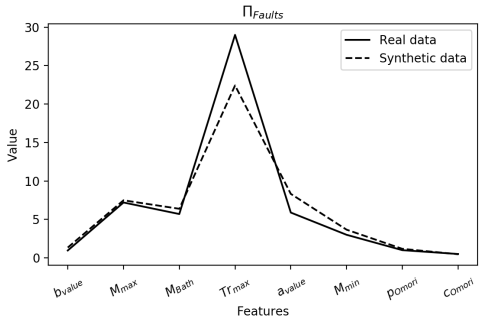
(b) Analysis with $\overrightarrow{B}$ label



(c) Analysis with Θ label



(d) Analysis with Π_{Asp} label



(e) Analysis with Π_{Faults} label

Fig. 3. Comparison between real data and synthetic data based on each input parameter.

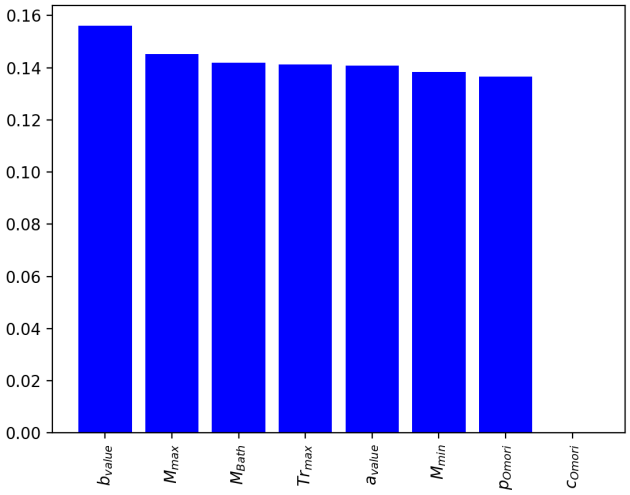


Fig. 4. Importance level of each statistical feature.

The last step of our analysis, representing an important objective of this work, is knowing which are the input parameters of the earthquake simulator that generates the best statistical catalog, with high similarity to the real data. To do so, we take the mean of the input parameters corresponding to the rows that contain the label predicted by the algorithm, and the resulting input parameters are shown in Table 4.

Parameter	**Random Forest**	**Heuristic analysis**
S_a	0.2	0.22
$\overrightarrow{B}$	0.7	0.9
Θ	0.1	0.1
Π_{Asp}	0.95	0.85
Π_{Faults}	0.84	0.75

Table 4. Comparison of the Random Forest TREMOL parameters to the ones previously found by an heuristic analysis.

In order to assess the quality of TREMOL results when using Random Forest parameters and those found by an heuristic analysis in [9], we compare the final statistical features under both parameterizations to the ones observed in the real mainshock-aftershock sequence. This comparison is given in Table 5, where

differences on real features, with respect to those in Table 1, are given by the omission of the foreshock events.

Statistical Feature	Real Value	Heuristic Mean	Heuristic Deviation	Random Forest Mean	Random Forest Deviation
b_{value}	0.99	1.34	0.03	1.36	0.02
M_{max}	7.2	6.94	0.15	7.33	0.09
M_{Bath}	5.7	6.44	0.24	6.31	0.44
Tr_{max}	18.0	10.18	1.29	14.23	1.15
a_{value}	6.55	8.34	0.10	8.39	0.09
M_{min}	3.0	3.67	0.0	3.66	0.0
p_{Omori}	1.0	1.12	0.15	1.18	0.15
c_{Omori}	0.5	0.39	0.28	0.49	0.32

Table 5. The mean and standard deviation of the statistical features from 20 TREMOL executions considering both sets of input parameters given in Table 4. The real statistical features are also shown as reference.

In Table 5, we observe that using Random Forest parameters in TREMOL, the resulting statistical features improve with respect to the real ones. In particular, results are better for b_{value}, M_{Max}, Tr_{max}, and a_{value}. However, the remaining features stay close to those obtained by using as input parameters, the ones delivered by the heuristic analysis, with the exception of the last feature c_{Omori}.

It is worth noting that the heuristic analysis previously applied in [9] is not systematic and may take more time to arrive at similar parameter values than those provided by the Random Forest analysis. However, this heuristic analysis may also fail by delivering values far away from the optimal ones. The comparison in Table 5 also allow the TREMOL sensitivity to the input parameters. Moreover, the differences between some real features and the corresponding synthetics help us to understand the possible improvements to our simulator. For example, in the case of M_{min} both sets of input parameters lead to larger values with respect to the real. This reflects the intrinsic TREMOL requirement of considering well refined meshes to be able to produce small magnitude earthquakes. Similarly, in the case of b_{value}, both TREMOL simulations shows higher values than the expected. These results could be also due to meshes too coarse, and then could be improved by higher mesh refinements. In summary, the collection results in Table 5 confirms that the Random Forest analysis is a proper tool to find TREMOL optimum parameters, and spend small amounts of computational time leading to higher accuracy than by heuristic analysis.

5 Conclusions

This work applies Random Forest classification for searching input parameters to the mainshock-aftershock simulator TREMOL. Under appropriate parameterization, TREMOL generates synthetic earthquake catalogs with a high statistical similarity to a real one. We also presents an importance measure of these input parameters on the statistical features that describe an earthquake catalog, and therefore serve as the similarity quantification. After training Random Forest classifier with different label values, where each label corresponding to an input TREMOL parameter, the label leading to the best approximation is Π_{Asp}. This label represents the percentage of the physical load distribution of asperity cells. Our results clearly support this important conclusion, in our study case of the Mw 7.2 *El Mayor-Cucapah* earthquake occurred in 2010.

We also would like to emphasize that during the Random Forest training, the most important feature was the b_{value}, followed by the $M_{\max}$. Alternatively, we observed that the c_{Omori} feature has no a significant importance, compared to all other statistical features. Thus, this last feature might be omitted in future similar studies. Moreover, after comparing the heuristic values with those found in this work in our study case, we conclude that the Random Forest TREMOL parameters improved the similarity of the generated catalog with respect to the real one. These results also suggest that this ML parameterization, enable TREMOL for catalog generation with great similarities to other real seismicities.

As future works, we envision the development of a software package that embedded both the TREMOL simulator and the optimal parameter screening modules. In addition, we would like to explore the effectiveness of searching methods, such as Gradient Boosting or even Neural Networks, if the amount of simulation data is sufficient enough.

Acknowledgements

This work is partially supported by the Spanish Ministry of Economy and Competitivity under contract TIN2015-65316-P and by the Catalan Government through the programmes 2017-SGR-1414, 2017-SGR-962 and the RIS3CAT DRAC project 001-P-001723. Moreover, this project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 777778 (MATHROCKS). The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the ChEESE project, grant agreement No. 823844.

References

1. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM review **59**(1), 65–98 (2017). https://doi.org/10.1137/141000671

2. Breiman, L.: Random Forests. Machine Learning **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
3. Burridge, R., Knopoff, L.: Model and theoretical seismicity. Bulletin of the Seismological Society of America **57**(3), 341–371 (1967)
4. Hauksson, E., Stock, J., Hutton, K., Yang, W., Vidal-Villegas, J.A., Kanamor, H.: The 2010 mw 7.2 el Mayor-Cucapah earthquake sequence, Baja California, Mexico and Southernmost California, USA:Active seismotectonics along the Mexican Pacific margin. Pure and Applied Geophysics **168**, 1255–1277 (2011). https://doi.org/10.1007/s00024-010-0209-7
5. Kagan, Y.Y., Knopoff, L.: Stocastic synthesis of earthquake catalogs. Journal of Geophysical Research Solid Earth **86**(B4), 2853–2862 (1981). https://doi.org/10.129/JB086iB04p02853
6. Monterrubio-Velasco, M., Carrasco-Jiménez, J.C., Castillo-Reyes, O., Cucchietti, F., De la Puente, J.: A machine learning approach for parameter screening in earthquake simulation. In: 30th International Symposium on Computer Architecture and High Performance Computing. pp. 348–355 (2018). https://doi.org/10.1109/CAHPC.2018.8645865
7. Monterrubio-Velasco, M., Rodríguez-Pérez, Q., Zúñiga, R., Scholz, D., Aguilar-Meléndez, A., de la Puente, J.: A stochastic rupture earthquake code based on the fiber bundle model (TREMOL v0.1): application to mexican subduction earthquakes. Geoscientific Model Development **12**(5), 1809–1831 (2019). https://doi.org/10.5194/gmd-12-1809-2019
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, P., Brucher, M., Perrot, M., Duchesnay, E., Louppe, G.: Scikit-learn: Machine learning in Phyton. Journal of machine learning research **12**, 2825–2830 (2011)
9. Scholz, D.: Numerical simulations of stress transfer as a future alternative to classical Coulomb stress changes. Master's thesis, University College London, Department of Earth Sciences, London (2018)
10. Turcotte, D.L.: Seismicity and self-organized criticality. Physics of the Earth and Planetary Interiors **111**(3-4), 275–293 (1999). https://doi.org/10.1016/S0031-9201(98)00167-8
11. Van Rossum, G.: Python tutorial. Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam (1995)
12. Vázquez-Prada, M., González, A., Gómez, J.B., Pacheco, A.F.: A minimalist model of characteristic earthquakes. Nonlinear Processes in Geophysics **9**, 513–519 (2002). https://doi.org/10.5194/npg-9-513-2002
13. Wu, Y.M., Shin, T.C., Tsai, Y.B.: Quick and reliable determination of magnitude for seismic early warning. Bulletin of the Seismological Society of America **88**(5), 1254–1259 (1998)