

NESTOREv1.0: A MATLAB Package for Strong Forthcoming Earthquake Forecasting

Stefania Gentili^{*1} , Piero Brondi¹ , and Rita Di Giovambattista² 

Abstract

This article presents the first publicly available version of the NExt STrOng Related Earthquake (NESTORE) software (NESTOREv1.0) designed for the statistical analysis of earthquake clusters. NESTOREv1.0 is a MATLAB (www.mathworks.com/products/matlab, last accessed August 2022) package capable of forecasting strong aftershocks starting from the first hours after the mainshocks. It is based on the NESTORE algorithm, which has already been successfully applied retrospectively to Italian and California seismicity. The code evaluates a set of features and uses a supervised machine learning approach to provide probability estimates for a subsequent large earthquake during a seismic sequence. By analyzing an earthquake catalog, the software identifies clusters and trains the algorithm on them. It then uses the training results to obtain forecasting for a test set of independent data to estimate training performance. After appropriate testing, the software can be used as an Operational Earthquake Forecasting (OEF) method for the next stronger earthquake. For ongoing clusters, it provides near-real-time forecasting of a strong aftershock through a traffic light classification aimed at assessing the level of concern. This article provides information about the NESTOREv1.0 algorithm and a guide to the software, detailing its structure and main functions and showing the application to recent seismic sequences in California. By making the NESTOREv1.0 software available, we hope to extend the impact of the NESTORE algorithm and further advance research on forecasting the strongest earthquakes during seismicity clusters.

Cite this article as Gentili, S., P. Brondi, and R. Di Giovambattista (2023). NESTOREv1.0: A MATLAB Package for Strong Forthcoming Earthquake Forecasting, *Seismol. Res. Lett.* **XX**, 1–11, doi: [10.1785/0220220327](https://doi.org/10.1785/0220220327).

[Supplemental Material](#)




Introduction

Usually, earthquake clusters can be divided into two types: swarm-like and mainshock–aftershock. Swarm-like sequences usually consist of many events of similar magnitude with not well-identified mainshock. Mainshock–aftershock clusters consist of a large event that triggers further events that decay with time according to the modified Omori law (Utsu *et al.*, 1995). For both types, if they have a magnitude comparable to or greater than that of the mainshock (M_m), they pose a significant threat from a disaster management perspective. Namely, they have the potential to cause further damage to already weakened buildings, increasing the risk of collapse and thus further damage and loss of life. A well-documented case is the 2010–2011 Canterbury cluster where, five months after the magnitude 7.1 earthquake, a magnitude 6.3 aftershock caused major additional destruction and killed numerous people (Kaiser *et al.*, 2012; Potter *et al.*, 2015).

To determine the probability of occurrence of strong forthcoming earthquakes, a machine learning algorithm, NExt STrOng Related Earthquake (NESTORE), was recently developed and successfully applied to Italian and California

seismicity; the algorithm and the results were shown in Gentili and Di Giovambattista (2017, 2020, 2022). The related software, NESTOREv1.0, written in MATLAB, is now sufficiently mature to be distributed to the scientific community for applications and testing in new areas.

As recommended in the Open Science agenda, for good scientific practice, not only the data but also research software should be Findable, Accessible, Interoperable, and Reusable (FAIR) to allow full repeatability, reproducibility, and reuse of results. This allows replication of results and ensures that the scientific community can apply the methodology to their data without having to rewrite the calculation codes. In addition, the use of the same software by multiple researchers allows for comparable results when evaluating performance on different datasets.

1. National Institute of Oceanography and Applied Geophysics, Udine, Italy,  <https://orcid.org/0000-0002-7740-7883> (SG);  <https://orcid.org/0000-0002-2341-2900> (PB); 2. Istituto Nazionale di Geofisica e Vulcanologia, Rome, Italy,  <https://orcid.org/0000-0001-5622-1396> (RDG)

*Corresponding author: sgentili@ogs.it

© Seismological Society of America

The primary goal of this article is to demonstrate the validity of NESTOREv1.0 by describing the algorithm and implemented methods in detail. NESTOREv1.0 makes use of several parameters and pattern recognition techniques.

The main idea of this code is to extract seismicity information from data reported in the earthquake catalog, such as location, time of occurrence, and magnitude of events at increasing time intervals after the occurrence of the main earthquake. Specifically, parameters related to the number of earthquakes, their energy, and spatiotemporal distribution (hereafter referred to as “features”) are measured to determine the probability of occurrence of strong subsequent events for the ongoing seismic cluster using a multiparameter pattern recognition approach. Recently, several studies analyzing the magnitude difference (D_m) between the main earthquake and the strongest subsequent earthquake have been proposed to provide methods for forecasting Strong Subsequent Earthquakes (SSEs). The studies are based on the Omori–Utsu law (Shcherbakov, 2014; Shcherbakov *et al.*, 2018), Epidemic Type Aftershock Sequence (ETAS) (Zhuang *et al.*, 2002, 2004, 2005, 2008; Zhuang and Ogata, 2006; Shcherbakov *et al.*, 2019), b -value (Helmstetter and Sornette, 2003; Shcherbakov and Turcotte, 2004; Gulia and Wiemer 2019, 2021; Gulia *et al.*, 2020), or mainshock properties (Persh and Houston, 2004; Tahir *et al.*, 2012; Rodríguez-Pérez and Zúñiga, 2016; Gentili and Di Giovambattista, 2017).

Vorobieva and Panza (1993) and Vorobieva (1999) proposed an SSE forecasting method based on several different seismicity features. They divided the clusters into two classes based on the magnitude value of the SSE: “type A” if the SSE has a magnitude $M_a \geq M_m - 1$ and “type B” otherwise. The aim of the method was to forecast clusters of type A, and each feature contributed to the estimate of the probability of their occurrence using a threshold-based approach. NESTORE follows the same approach, but differently from Vorobieva’s algorithm, it considers a wider set of features, performs its analysis considering smaller time intervals after the mainshock (Gentili and Di Giovambattista, 2017), requires a smaller magnitude range for the analysis (Gentili and Di Giovambattista, 2020), and performs accurate feature’s performance validation during the training (Gentili and Di Giovambattista, 2022). These features increase both the ability of the method to differentiate between the clusters belonging to the two classes and the number of clusters available. This leads to a more reliable validation of the method, making NESTORE a robust method suitable for semi-real-time seismic exposure mitigation. It is important to note that the classification approach of this method does not divide the clusters into the two classes “foreshock-mainshock-aftershocks” (also called Omori sequences) and “swarms”—as it is common in the literature, but the distinction between the classes is based only on the difference in magnitude between the mainshock and the strongest aftershock. Although swarms are class A clusters and class B clusters are Omori sequences,

an Omori sequence can be type A if the magnitude of the strongest aftershock is sufficiently high.

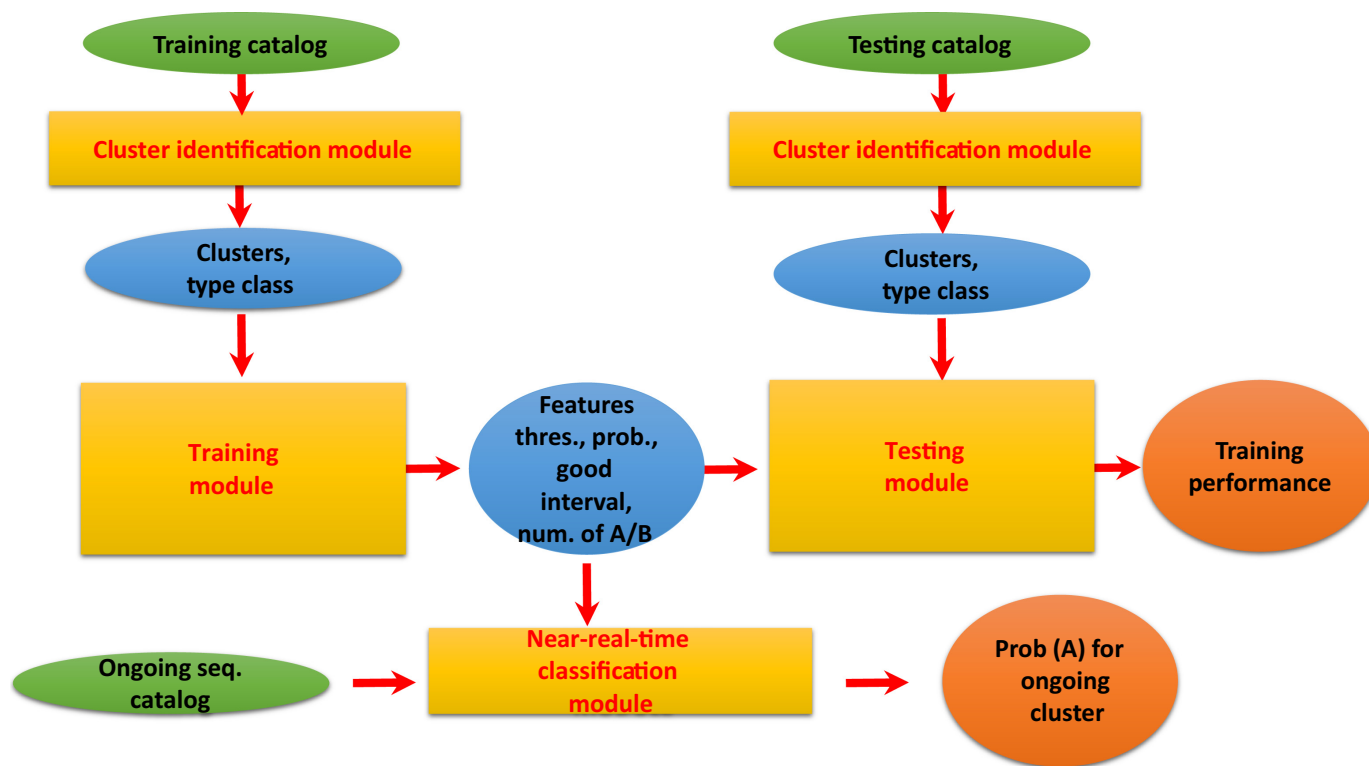
Structure of NESTOREv1.0

NESTOREv1.0 is written in MATLAB (required toolboxes: “Mapping toolbox” and “Statistics and Machine Learning toolbox”), and the source code is freely available under a GPLv3 license in a dedicated GitHub repository (see [Data and Resources](#)). It analyzes the evolution of seismicity at different time intervals after a strong event using a multiparameter pattern recognition approach. In particular, given the occurrence of a strong mainshock, it aims to estimate the probability that the ongoing seismic cluster is type A. The software uses a supervised machine learning approach and is optimized to be trained even when the number of available clusters is small, that is, on the order of a few tens, even requiring decades of seismic data records in certain regions. To this end, NESTOREv1.0 computes features for each cluster and distinguishes between type A and B clusters by training a separate one-level decision tree for each feature, minimizing the probability of overfitting due to the small training set. During the training process, the trees that provide the best classification are selected. After training, the classifications of the selected trees are merged using a Bayesian approach. NESTOREv1.0 consists of four main parts:

1. The cluster identification module extracts clusters from an input catalog and provides their actual classification and statistical information about them.
2. The training module computes the features for each cluster and estimates thresholds and probabilities using a machine learning approach to obtain a reliable forecasting of type A cases.
3. The testing module allows the user to check the results of the training procedure on a different set of clusters.
4. The near-real-time classification module performs a near-real-time analysis of an ongoing cluster, providing the probability of occurrence of a cluster of type A.

The workflow of the algorithm is illustrated by the flowchart in Figure 1.

Each of the modules consists of a main script and its associated functions. To make the correspondence between the main script, its functions, and its subfunctions easily recognizable, we have chosen a naming scheme based on logical dependence. In particular, each subfunction name contains the name of its “parent” function. An exception occurs when the subfunction belongs to MATLAB programs of other authors; in that case, the name is left unchanged. A detailed list of functions can be found in the supplemental material, available to this article, and under the software link in the [Data and Resources](#). The structure of the NESTOREv1.0 folders is shown in Figure 2. The input and output data flows of the



main program of each module with the directories containing the data are also shown.

Cluster Identification Module

This module identifies seismicity clusters within a given area using a window-based method: first, all events in the input catalog whose magnitude is greater than a fixed threshold T_{HM} are considered mainshocks; then, for each of these events, a time window τ and a spatial distance r are estimated as a function of their magnitude M_m . Only earthquakes occurring in the time window τ after the mainshock and within a circular area of radius r around the location of the mainshock are assigned to the corresponding cluster. The code automatically merges the overlapping clusters. Different types of r and τ functions have been proposed in the past (Gardner and Knopoff, 1974; Uhrhammer, 1986; Knopoff, 2000; Kagan, 2002; Lolli and Gasperini, 2003; Gentili and Bressan, 2008) because the functions depend on the analyzed region; in the NESTOREv1.0 software, the user can specify the most appropriate function for the region under study through a dedicated input file. This module also selects the foreshocks for future extension of the NESTOREv1.0 software. Spatial selection of foreshocks is done using a circular region of radius $x \cdot r$, in which r is the radius previously specified for the aftershocks and x is a variable that can be chosen by the user (the default is $x = 1.5$). The time window for the foreshock can be selected by the user (the default is one month). The cluster identification module provides as output for each cluster the mainshock and the list of foreshocks and aftershocks. It also provides information about the SSE and the completeness magnitude M_c of the cluster. The completeness

Figure 1. NESTOREv1.0 flowchart. The color version of this figure is available only in the electronic edition.

magnitude is evaluated with the function “ $calc_M_c$ ” by D. Schorlemmer and J. Woessner, which belongs to the software Zmap (Wiemer, 2001). The function allows to choose the method to evaluate the completeness magnitude by calling different functions. In NESTOREv1.0, we use the function “ $calc_McMaxCurvature$,” which uses the maximum curvature method, to which we added a correction factor of 0.2, as recommended by Woessner and Wiemer (2005). The error is estimated by bootstrapping. A user with average MATLAB programming experience can add other functions of Zmap to choose a different method or develop a new method autonomously.

Figure 3 shows an example of a possible input configuration file of the cluster identification module. The user can specify the equation for determining the cluster identification window, the lower limit of mainshock magnitude, the x -value and the time duration for foreshocks, and so forth. The detailed description of the input file for each module is distributed with the software in the “documents” folder.

Training Module

The training code takes as input the clusters identified by the previous module and extracts nine features based on the energy and seismic productivity of each cluster. Machine learning approach is then used to estimate an appropriate set of

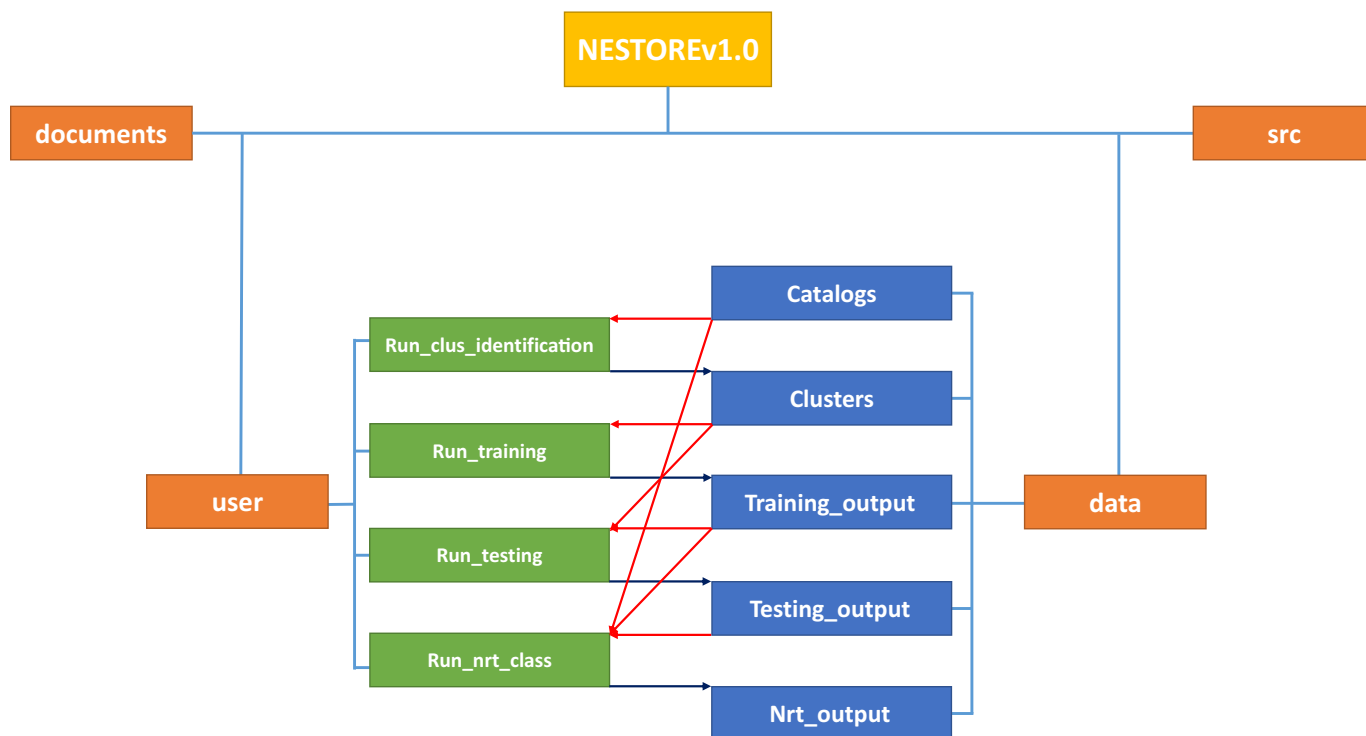


Figure 2. Folders' scheme of NESTOREv1.0. Green rectangles denote main runs. Arrows denote input (red) and output (blue) data flows between the main runs and subfolders of "data" directory (blue rectangles). The color version of this figure is available only in the electronic edition.

thresholds for the features to differentiate type A cases from type B cases. The training procedure consists of four parts: (1) features extraction, (2) decision trees training, (3) selection of a time interval characterized by good performance for each feature (good interval), and (4) inheritance and validation.

Feature extraction

The features are extracted from events with magnitude $\geq M_m - 2$ for increasing time intervals T_i from the occurrence of the mainshock (Gentili and Di Giovambattista, 2017, 2020). Of the nine features (see supplemental material), two relate to the number and spatiotemporal distribution of events (the number of events $N2$ and the linear concentration of events Z), four to the source area and magnitude trend over time (the cumulative change in magnitude between events Vm , the normalized source area of events S , and the cumulative deviation S from the long-term trend with two different approaches SLCum and SLCum2), and three to the energy contribution E of the individual event over time (the radiated energy Q and the cumulative deviation Q from the long-term trend with two different approaches QLCum and QLCum2). E is estimated as a function of magnitude M as follows:

$$\text{Log}_{10}(E) = \frac{3}{2}M + 4.8. \quad (1)$$

All features are computed by considering events occurring from 1 minute after the mainshock on increasing time intervals ending every 6 hr during the first day and every 24 hr during

the first week (10 time intervals). The starting time of the analysis was chosen to avoid problems related to the increase of M_c after the mainshock (see Gentili and Di Giovambattista, 2022 for more details).

Decision trees training

Nine one-node decision trees are trained for each of the 10 time periods T_i . Considering the small training set available for NESTORE applications, we used linear classification decision trees implemented in the "fitree" function in MATLAB, thus avoiding more complex algorithms that provide optimal performance for large training sets (see also Gentili and Di Giovambattista, 2017). Specifically, for each feature and period T_i , the corresponding tree finds a threshold such that most clusters of type A have feature values above the threshold. If no threshold meets this condition, the threshold is set to "Not a Number" (NaN).

Selection of good interval

To ensure high reliability of the determined thresholds, a detailed statistical analysis is performed to verify the performance of the decision trees through a procedure called "good interval selection." Specifically, the "Leave One Out" (LOO)


```

1 | %-----CATALOG FILE NAME-----
2 | Iside_catalog_2005_04_16_to_2021_04_30.zmap
3 | %-----OUTPUT CLUSTERS FILE NAME-----
4 | cluster_Iside_catalog_2005_04_16_to_2021_04_30.dat
5 | %-LOWER LIMIT ON MAINSHOCK MAGNITUDE (Mm) FOR CLUSTERS -----
6 | 4.0
7 | %-----Mm VALUE FOR TIME DURATION FORM CHANGE (Thwt)-----
8 | 6.5
9 | %--EQUATION FOR SPATIAL SELECTION OF AFTERSHOCKS -----
10 | 0.02*10.^(M/2)
11 | %-EQUATION FOR TEMPORAL SELECTION OF AFTERSHOCKS IF Mm <Thwt--
12 | 10.^(0.5409*M-0.547)
13 | %-EQUATION FOR TEMPORAL SELECTION OF AFTERSHOCKS IF Mm>=Thwt--
14 | 10.^(0.032*M+2.7389)
15 | %-----COEFFICIENT FOR SPATIAL SELECTION OF FORESHOCKS-----
16 | 1.5
17 | %-----TEMPORAL SELECTION OF FORESHOCKS [YEARS]-----
18 | 0.082
19 | %%%PLEASE DO NOT MODIFY THE ABOVE COMMENT LINES!!
20 |

```

Figure 3. Cluster identification module input file. The color version of this figure is available only in the electronic edition.

method is used to calculate four statistical evaluators of the performance of each decision tree corresponding to different feature and time period. The adopted evaluators are Accuracy, Precision, Recall, and Informedness (Gentili and Di Giovambattista, 2020). The first three evaluators range from 0 to 1, which correspond to the worst and best performance, respectively, and the last one ranges from -1 to 1 (-1 = worst, 1 = best). According to the results of different tests (Gentili and Di Giovambattista, 2017, 2020, 2022), these four performance evaluators increase with T_i up to a maximum value; for the subsequent intervals, their trend is constant or decreases. Let s_2 be the value of T_i for which the Informedness is maximum; NESTOREv1.0 defines “good interval” as the set of periods of T_i in which:

1. Accuracy, Precision, and Recall are strictly above 0.5;
2. Accuracy is equal to or greater than that of a tree that classifies every cluster in the more populated class;
3. Informedness is strictly greater than 0; and
4. T_i ends at a time less than or equal to s_2 .

The decision to use only features for T_i that are in the “good interval” allows us to consider features with good and stable

performance. In particular, point 2 guarantees that imbalances in classes A and B do not lead to unrealistic estimates of the algorithm’s ability to distinguish between the two classes. For times longer than s_2 , the features are still used in combination with the others, but both the value of the features and the thresholds are “inherited” from the time interval ending at s_2 . This selection is essential to obtain reliable results when the number of training examples is small. An example of such a selection can be found in Figures 4 and 5 in the Application of NESTOREv1.0 section and supplemental material figures.

Inheritance and validation process

The last part of the procedure of the training module involves the validation of the computed or inherited features and their thresholds.

In particular, NESTOREv1.0 again applies a LOO method to the training set and computes the percentage of correct classifications for clusters of type A (hit rate), comparing it with the percentage of incorrect classifications for cases of type B (false alarm rate). For T_i periods in which the hit rate is less than or equal to the false alarm rate, the threshold is set to NaN, and the feature is ignored in the classification phase.

The training module outputs are four vectors containing:

1. the “good range ” limits for individual features (Validity vector),
2. the feature threshold for each T_i (Threshold vector),
3. the probability for seismicity clusters to be type A above and below the threshold for each feature and time period (Probability vector), and
4. the number of type A and type B clusters available for each T_i in the training set (NAB vector).

The training module also shows some figures representing the performance of feature classification as a function of the interval of periods T_i chosen by the user (see the supplemental material for more details).

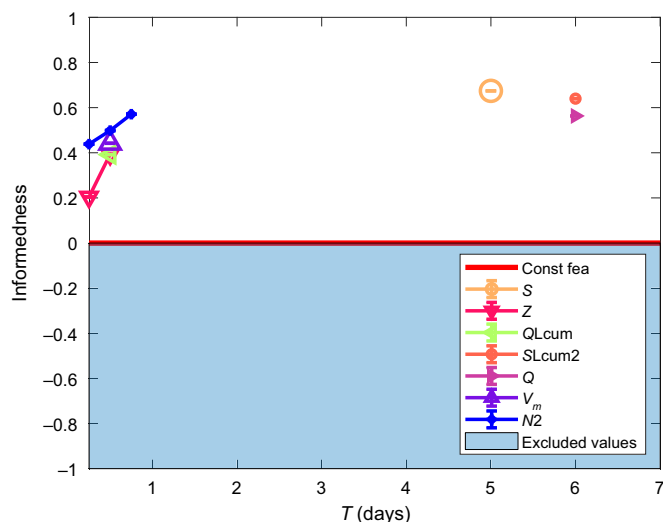


Figure 4. Informedness as a function of analysis periods during training. Different symbols represent the performance of the features in their good range. The blue area represents excluded Informedness values for the selection of the good range. The red line represents response of a classifier that classifies all clusters as B. The color version of this figure is available only in the electronic edition.

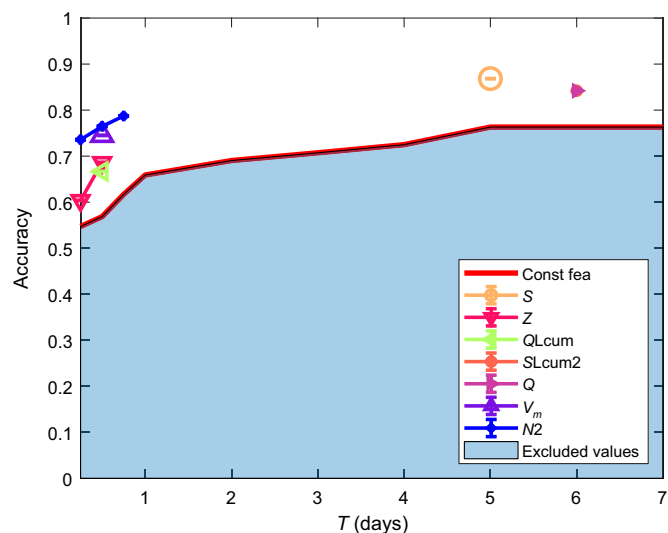


Figure 5. Accuracy as a function of analysis periods during training. Different symbols represent the performance of the features in their good ranges. The blue area represents excluded Accuracy values for the selection of the good range. The red line represents response of a classifier that classifies all clusters as B. The color version of this figure is available only in the electronic edition.

Testing Module

In this module, the information obtained from the training procedure is used to provide forecasts of type A clusters in a test catalog. After the cluster identification step, the procedure extracts the features for each cluster over the time periods T_i , specified in the validity vector and inherits the values for longer time periods. Then for each T_i and each cluster, the Threshold vectors are compared with the corresponding computed features; if the corresponding threshold is not a NaN, NESTOREv1.0 uses the Probability vector to estimate

$$p_{n,i} = P(A|F_{n,i}), \quad (2)$$

which is the probability that the cluster is type A given the value of the n th feature F_n at the period T_i . Using a Bayesian approach (Gentili and Di Giovambattista, 2020), NESTOREv1.0 combines all feature probabilities $p_{n,i}$, to obtain the probability P_i for cluster to be a type A cluster at time T_i :

$$P_i(F_1 \dots F_N) = \frac{[N(B)_i]^{N-1} \prod_{n=1}^N p_{n,i}}{[N(B)_i]^{N-1} \prod_{n=1}^N p_{n,i} + [N(A)_i]^{N-1} \prod_{n=1}^N (1-p_{n,i})}, \quad (3)$$

in which $N(A)_i$ and $N(B)_i$ are the number of clusters of type A and B in the training database, respectively, during the time interval T_i ; they are both listed in the NAB vector. The inclusion of the number of A and B clusters allows the code to account for the possible imbalance between type A and type

B classes in the training set, which is very common in statistical seismology applications. The information about the actual class of test clusters is compared with the classification output to obtain an estimate of the algorithm's performance. The outputs of this module are the cluster classification list along with the Receiving Operating Characteristics (ROC) and the Precision-Recall graphs, which summarize the performances of the training on a testing database. For further details on the graphs, see the [Application of NESTOREv1.0](#) section and the previous articles on NESTORE (Gentili and Di Giovambattista, 2017, 2020, 2022). In addition to the performances of the single feature classifiers and the final Bayesian NESTORE classification, the two graphs also show, for comparison, the result of a simple voting procedure applied to the results of the single classifiers. The [Application of NESTOREv1.0](#) section and the supplemental material show some examples of the two graphs.

Near-Real-Time Classification Module

The latest module of NESTOREv1.0 is designed to provide near-real-time forecasting of cluster typology during an ongoing seismic cluster. First, the cluster is extracted from the catalog of current seismic activity and its features are computed. Then these are compared with the thresholds and probabilities obtained from the training module for the study area, and the Bayesian approach is applied to estimate the probability that it is a type A cluster. It is important that the training module stores its results in output files so that the near-real-

time classification module can use these results later, when needed, and its run time is not affected by the previous modules one. The outputs of the module, in addition to a map of the cluster and a time–magnitude diagram, are the classification of the cluster over time, the class forecasting, the information about the time range corresponding to the forecast, and the space after and around the mainshock for which the forecasting of strong aftershocks was performed. The final class forecasting is summarized using a simple traffic light classification similar to the method of [Gulia and Wiemer \(2019\)](#). The authors proposed a Foreshock Traffic Light System (FTLS) based on the temporal analysis of the b -value measured during foreshocks and aftershocks, which can output three alert levels for the occurrence of a strong aftershock. The near-real-time classification module of NESTOREv1.0 considers an Aftershock-based Traffic Light System (AFTLS) based on the events following a strong earthquake (considered as aftershocks) that is based on the value of type A probability estimated in all analysis periods. Similar to the FTLS of [Gulia and Wiemer \(2019\)](#), the AFTLS displays green and red traffic lights when the probability of occurrence of a strong aftershock is low and high, respectively, whereas the yellow traffic light indicates an uncertain estimate. It is important to remark that the proposed traffic light system is not an Adaptive Traffic Light System (ATLS), for which the threshold is based on a quantitative risk assessment (e.g., [Mignan et al., 2017](#)) but rather, as in [Gulia and Wiemer \(2019\)](#), uses thresholds based on expert judgement to summarize information contained in the other output figures.

Software Development Principles

When writing the NESTOREv1.0 code, we considered principles of agile software development such as KISS (Keep It Short and Simple) and DRY (Don't Repeat Yourself; [Hunt and David, 1999](#); [Misra, 2004](#)). The first principle recommends keeping things as simple as possible, for example, by splitting complex parts into simple subfunctions, and the second suggests not including duplicate program parts. The advantages of these approaches are easier code maintenance and greater flexibility when making changes or refactoring. The principles are independent of the application domain of the code: for example, KISS has been recently used by clinical oncology software ([Zhou et al., 2021](#)) and healthcare information technology ([Herzlinger et al., 2013](#)), and DRY has been used in experimental business programs (CORAL, [Shaffner, 2013](#)) and web applications ([Jaiswal and Kumar, 2015](#)). We followed the approach of KISS using separate functions that have a specific and simple goal. We also considered DRY, reusing the same functions in different modules (e.g., the near-real-time classification module shares several functions with the other three modules).

According to [Wilson et al. \(2014\)](#), the NESTOREv1.0 code was written with the idea that code should be written for humans, not computers; we made the function names descriptive and adopted a naming convention in which each

subfunction name contains the name of the corresponding function plus the specification of the particular subtask. To simplify the use of NESTOREv1.0, all input parameters are listed in a commented input file (one for each module) that can be saved for later recording of the run. A folder structure is included to separate the inputs and outputs of each module. The examples shown in the following sections are provided along with readme files that explain the use of the code, folders, and input file structure. The modularity of the package and the storage of results in files allow the user to perform slower operations (e.g., cluster identification) only when necessary, avoiding wasted time in analysis. We have made the code available on GitHub to share the developed tools with the whole seismological community. Any suggestion or bug report is welcome to improve the code. Depending on feedback and end-user requirements, a Python version could also be considered.

Application of NESTOREv1.0

Previous versions of the NESTORE algorithm were successfully applied to all of Italy, northeastern Italy, western Slovenia, and California ([Gentili and Di Giovambattista, 2017, 2020, 2022](#)). In the following, we show the results of the new version of the code on California databases. To follow modern open-science principles and ensure that experimental results are versioned, openly accessible, and reproducible, we make all input files available in a dedicated GitHub repository along with the NESTOREv1.0 code, thus providing the “reproducibility package” ([Savran et al., 2022](#), see [Data and Resources](#) for link).

Training module

In this example, we show the application of the NESTOREv1.0 training module to seismicity in California by presenting the key parameters and results. The training was performed on 55 magnitude >4 clusters that occurred in southern California from 1980 to 2019. Data were extracted from southern California earthquake catalog (see [Data and Resources](#)). The good interval is automatically selected based on four evaluators of the feature thresholds performance calculated for each time interval (see the [Training module](#) section). After evaluating the good range for each feature, the code displays the results. Here we present two of the outputs: Informedness (Fig. 4) and Accuracy (Fig. 5), which respectively indicate the amount of information provided by the feature and the performance of a feature in terms of the normalized percentage of correct prediction ([Powers, 2011](#); [Gentili and Di Giovambattista, 2017](#)). In both figures, the blue empty areas mark the thresholds that the algorithm automatically sets to discard features with poor performance. The images allow an immediate evaluation of the time periods T_i in which the features perform better.

After evaluating the good interval for a given feature, for longer time intervals, the features and corresponding thresholds are “inherited” to subsequent time intervals to be used

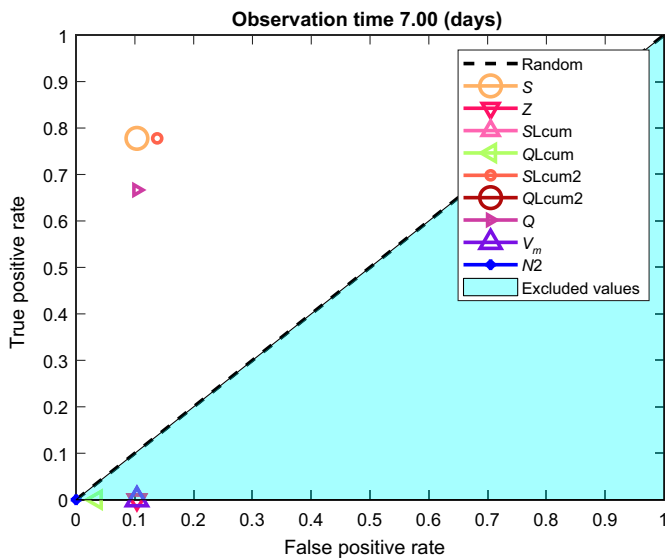


Figure 6. Receiving operating characteristics (ROC) graph showing the performance of the feature thresholds during training. The symbols represent the performances of the different features. The black dashed line corresponds to the random guessing. The color version of this figure is available only in the electronic edition.

together with other features. For each time interval, NESTOREv1.0 shows the trade-off between benefits (normalized percentage of correctly classified type A clusters—true-positive rate) and costs (normalized percentage of incorrectly classified type B clusters—false-positive rate) through an ROC diagram (see Fig. 6). Features in the area below the diagonal (in the example QLcum, V_m , Z) that are underlined in a blue triangular box have worse performance than random guessing and are therefore not included in the training output.

Testing module

The testing module evaluates the performance of the algorithm by testing it on an independent test set and comparing the actual cluster class to the forecasted one. To show these performances, it provides as output the confusion matrix, the ROC graph, the Precision-Recall graph, and the classification of clusters in time. Unlike the ROC graph, for which the best performances correspond to the upper left corner, the best performances for the Precision-Recall graph correspond to the upper right corner. Although Recall coincides with true positive rate, Precision provides information on how well the model identifies type A clusters.

Figures 7 and 8 refer to the application of the NESTOREv1.0 testing procedure to an independent test set of nine clusters in northern and southern California (extracted from the ComCat catalogue; see [Data and Resources](#)). In particular, Figure 7 shows the performances in terms of Precision-Recall graph. The final NESTOREv1.0 classification (Bayesian approach; see equation 3)

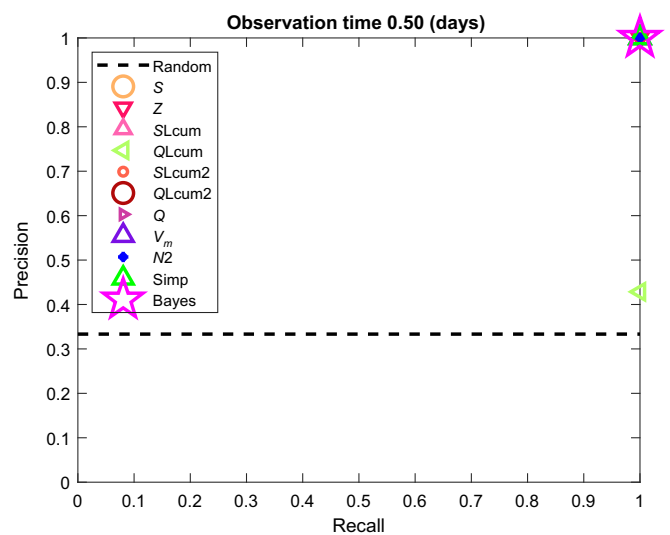


Figure 7. Precision-Recall graph showing the performance of the features thresholds on the testing data set. The different symbols represent the performances of different classifiers; the black dashed line corresponds to random guessing. The green triangle and magenta star correspond to simple and Bayesian classification, respectively. The color version of this figure is available only in the electronic edition.

is shown by a magenta star, and the simple voting, in this case coincident, is represented by a green triangle. Although not all features (e.g., QLcum) perform well, the merging the classification of multiple features gives the best results.

Figure 8 shows the final result of the testing module, that is, the Bayesian probability estimate for each cluster to be a type A, for increasing user-specified time periods. NESTOREv1.0 provides this figure along with a detailed table containing the performance of each feature for each time period and cluster. In the figure, the color of the curve refers to the actual cluster type, that is, red or blue for type A or type B, respectively. In this way, the user can immediately check whether the type A and type B clusters are correctly classified (above and below the 0.5 line, respectively). If the forecasting of the cluster type in the last period is wrong, the corresponding window is underlined in yellow (not shown in the example).

Near-real-time classification module

During the occurrence of a new cluster, the near-real-time classification module can be applied, starting 6 hr after the mainshock. After the cluster is identified, the algorithm displays both the distribution of aftershocks around the epicenter of the mainshock and their magnitude versus time. Figure 9 shows the map of aftershocks of a cluster of the training set for which a near-real-time application is simulated 18 hr after the mainshock.

NESTOREv1.0 then uses the information from the training procedure to evaluate the probability for the cluster of being a type A. The classification of each feature for each time interval

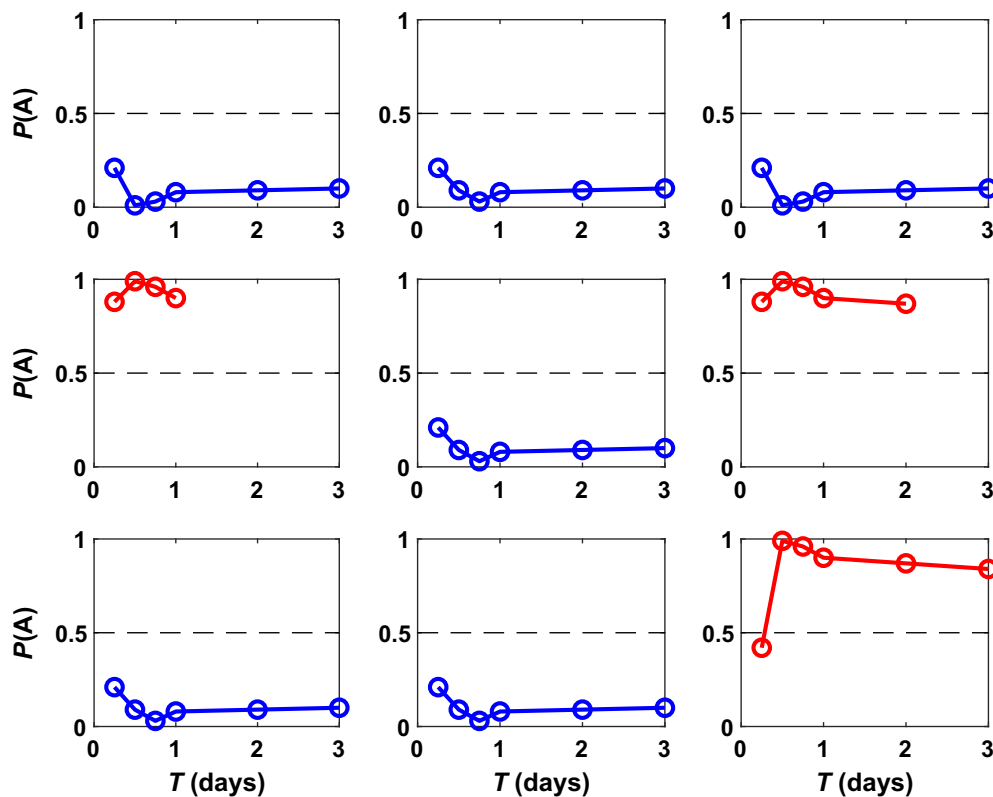


Figure 8. Bayesian probability of being a type A cluster for the test set as a function of analysis periods. The red and blue lines indicate that input clusters are of type A and B, respectively. The color version of this figure is available only in the electronic edition.

is displayed in detail to the user to understand how the features contribute to the final probability estimate (see the supplemental material for further details). As in the case of testing procedure, the classification of each feature for each time interval is used to determine the final probability estimate.

In Figure 10, for each time interval T_i , the Bayesian overall probability that the running cluster is of type A is given. Unlike the testing module, in near-real-time applications, the actual classification is not known, and the color of the probability curve is related to the forecasting of the cluster type: it is red if the probability value is >0.6 in all analysis periods, blue if the probability is always <0.4 , and black otherwise. Similarly, the AFTLS classification, referred to all analysis periods, shows a red light if the type A probability value is always >0.6 (strong aftershock alert), a green light if it is confined <0.4 (no strong aftershock alert), and a yellow light in all other cases. The algorithm also provides the radius of the circular area (centered on the position of the mainshock), the time period after the mainshock for which the forecasting is performed, and the minimum magnitude ($M_m - 1$) investigated for forecasting. If the forecasted class is the same for all studied periods and is outside the range $[0.4, 0.6]$, the corresponding class name is displayed.

Even if the classifications of a few individual features do not provide very clear results for a given period, the integration of different classifiers can provide more accurate results. This makes the multiple feature based approach more robust than the single feature based approach.

Conclusions

NESTOREv1.0 is a new free and fully automated MATLAB-based software package for forecasting a strong aftershock following the occurrence of a strong event. This toolbox uses a multiple feature-based machine learning approach and, after a training and testing procedure, is able to provide a near-real-time classification of the current seismicity cluster. The forecasting starts 6 hr after the intense earthquake (operative mainshock), refers to a well-defined time and space window depending on the

magnitude of the earthquake, and can be repeated at larger time intervals ≤ 7 days after the mainshock. In this article, we showed an example of application to the California seismicity. Performance is estimated and presented by the testing module in the form of ROC and Precision-Recall graphs.

The NESTOREv1.0 package is organized in a folder scheme that helps the user to distinguish the main module codes from their functions, clearly discern input data from output data, and easily retrieve the main figures generated by the algorithm. Previous versions of the NESTOREv1.0 algorithm have been successfully applied to areas in Italy and California (Gentili and Di Giovambattista 2017, 2020, 2022). The current improved version is being applied in Italy (Brondi *et al.*, 2023) and Greece (Gentili *et al.*, 2023). Because NESTOREv1.0 is the first version of a MATLAB-based toolbox for strong aftershocks forecasting, we expect that the algorithm will be improved and extended in subsequent versions. We hope that public availability will be useful for applying the code to many different seismotectonic domains, for further discussion on this topic, and for improving the method. Depending on the end users' feedback and requirements, we are planning to write a Python version as well.

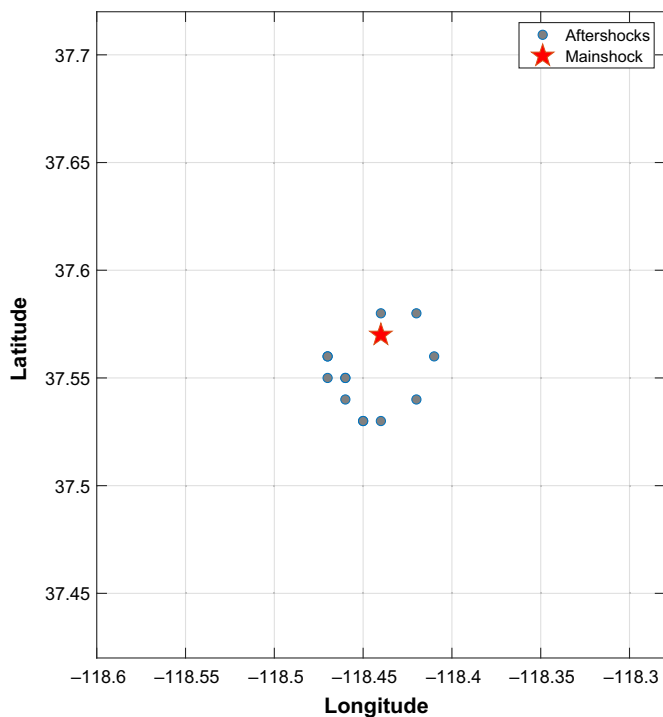


Figure 9. Spatial distribution of aftershocks of an ongoing cluster (gray dots) and mainshock epicenter (red star), shown by the near-real-time classification module. The color version of this figure is available only in the electronic edition.

Data and Resources

The NEXt STRong Related Earthquake (NESTOREv1.0) toolbox is available for free download from GitHub at <https://github.com/StefaniaGentili/NESTORE> (last accessed March 2023), and the reproducibility package is available in Zenodo at <https://zenodo.org/account/settings/github/repository/StefaniaGentili/NESTORE> (last accessed March 2023). At these links, the code is proposed together with the reproducibility package for the examples in this article, including a detailed readme for software usage. The tests shown in this article were obtained using data of the southern California earthquake catalog of the Southern California Earthquake Data Center (SCEDC), 2013 that was downloaded at <https://service.scedc.caltech.edu/eq-catalogs/> (last accessed August 2021). Caltech dataset, doi: 10.7909/C3WD3xH1, was downloaded at https://service.scedc.caltech.edu/eq-catalogs/date_mag_loc.php (last accessed August 2021) and from the Comprehensive Earthquake Catalog (ComCat) that was downloaded from the U.S. Geological Survey (USGS) at <https://earthquake.usgs.gov/earthquakes/search/> (last accessed January 2021). The supplemental material for this article includes an accurate definition of the features used for the classification, tables on the Central Processing Unit (CPU) computational time required for the tests of the reproducibility package on different machines, and some figure of the code output that allow the user to better understand the classification procedure.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

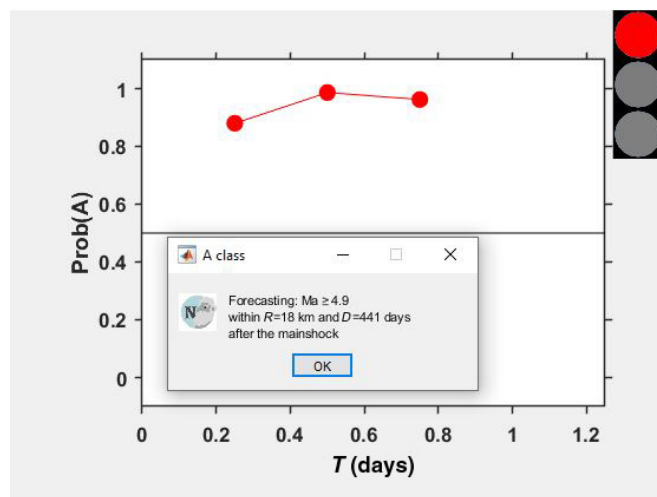


Figure 10. Bayesian probability of being a type A as a function of analysis periods shown by the near-real-time classification module for an ongoing cluster. The Aftershock-based Traffic Light System (AFTLS) classification for the ongoing cluster is shown at the upper right vertex. The color version of this figure is available only in the electronic edition.

Acknowledgments

This study was funded by a grant from the Italian Ministry of Foreign Affairs and International Cooperation. The authors wish to thank Francesco Giannitrapani for NEXt STRong Related Earthquake's (NESTORE) logo. In addition, the authors thank William Savran and an anonymous reviewer whose suggestions helped improve and clarify this article.

References

- Brondi, P., S. Gentili, and R. Di Giovambattista (2023). Forecasting strong aftershocks in the Italian territory: A National and Regional application for NESTOREv1.0, *EGU General Assembly 2023*, Vienna, Austria, 23–28 April 2023.
- Gardner, J. K., and L. Knopoff (1974). Is the sequence of earthquakes in Southern California, with aftershocks removed, Possionian? *Bull. Seismol. Soc. Am.* **64**, no. 5, 1363–1367.
- Gentili, S., and G. Bressan (2008). The partitioning of radiated energy and the largest aftershock of seismic sequences occurred in the northeastern Italy and western Slovenia, *J. Seismol.* **12**, 343–354.
- Gentili, S., and R. Di Giovambattista (2017). Pattern recognition approach to the subsequent event of damaging earthquakes in Italy, *Phys. Earth Planet. In.* **266**, 1–17.
- Gentili, S., and R. Di Giovambattista (2020). Forecasting strong aftershocks in earthquake clusters from northeastern Italy and western Slovenia, *Phys. Earth Planet. In.* **303**, doi: 10.1016/j.pepi.2020.106483.
- Gentili, S., and R. Di Giovambattista (2022). Forecasting strong subsequent earthquakes in California clusters by machine learning, *Phys. Earth Planet. In.* **327**, doi: 10.1016/j.pepi.2022.106879.
- Gentili, S., E. Anyfadi, P. Brondi, and F. Vallianatos (2023). Forecasting strong subsequent earthquakes in Greece using NESTORE machine learning algorithm, *EGU General Assembly 2023*, Vienna, Austria, 23–28 April 2023.

- Gulia, L., and S. Wiemer (2019). Real-time discrimination of earthquake foreshocks and aftershocks, *Nature* **574**, 193–199.
- Gulia, L., and S. Wiemer (2021). Comment on “Two Foreshock Sequences Post Gulia and Wiemer (2019)” by Kelian Dascher-Cousineau, Thorne Lay, and Emily E. Brodsky, *Seismol. Soc. Am.* **92**, no. 5, 3251–3258.
- Gulia, L., S. Wiemer, and G. Vannucci (2020). Prospective evaluation of the foreshock traffic light system in Ridgecrest and implications for aftershock hazard assessment, *Seismol. Res. Lett.* doi: [10.1785/0220190](https://doi.org/10.1785/0220190).
- Helmstetter, A., and D. Sornette (2003). Båth’s law derived from the Gutenberg-Richter law and from aftershock properties, *Geophys. Res. Lett.* **30**, no. 20, 2069, doi: [10.1029/2003GL018186](https://doi.org/10.1029/2003GL018186).
- Herzlinger, R., M. Seltzer, and M. Gaynor (2013). Applying KISS to healthcare information technology, *Computer* **46**, no. 11, 72–74, doi: [10.1109/MC.2013.298](https://doi.org/10.1109/MC.2013.298).
- Hunt, A., and T. David (1999). *The Pragmatic Programmer: From Journeyman to Master*, First Ed., Addison-Wesley, Boston, Massachusetts.
- Jaiswal, S., and R. Kumar (2015). *Learning Django Web Development*, Vol. 336, Packt Publishing, Birmingham, United Kingdom.
- Kagan, Y. Y. (2002). Seismic moment distribution revisited: I. Statistical results, *Geophys. J. Int.* **148**, 520–541.
- Kaiser, A., C. Holden, J. Beavan, D. Beetham, R. Benites, A. Celentano, D. Collett, J. Cousins, M. Cubrinovski, G. Dellow, et al. (2012). The Mw 6.2 Christchurch earthquake of February 2011: Preliminary report, *New Zeal. J. Geol. Geophys.* **55**, no. 1, 67–90, doi: [10.1080/00288306.2011.641182](https://doi.org/10.1080/00288306.2011.641182).
- Knopoff, L. (2000). The magnitude distribution of declustered earthquakes in Southern California, *Proc. Natl. Acad. Sci. Unit. States Am.* **97**, 11,880–11,884.
- Lolli, B., and P. Gasperini (2003). Aftershocks hazard in Italy Part I: Estimation of time magnitude distribution model parameters and computation of probabilities of occurrence, *J. Seismol.* **7**, 235–257.
- Mignan, A., M. Broccardo, S. Wiemer, and D. Giardini (2017). Induced seismicity closed-form traffic light system for actuarial decision-making during deep fluid injections, *Sci. Rep.* **7**, doi: [10.1038/s41598-017-13585-9](https://doi.org/10.1038/s41598-017-13585-9).
- Misra, R. B. (2004). Global IT outsourcing: Metrics for success of all parties, *J. Inform. Technol. Case Appl. Res.* **6**, 21–34.
- Persh, S. E., and H. Houston (2004). Strongly depth-dependent aftershock production in deep earthquakes, *Bull. Seismol. Soc. Am.* **94**, 1808–1816.
- Potter, S. H., J. S. Becker, D. M. Johnston, and K. P. Rossiter (2015). An overview of the impacts of the 2010–2011 Canterbury earthquakes, *Int. J. Disaster Risk Reduct.* **14**, 6–14.
- Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation, *J. Machine Learn. Technol.* **2**, no. 1, 37–63.
- Rodríguez-Pérez, Q., and F. R. Zúñiga (2016). Båth’s law and its relation to the tectonic environment: A case study for earthquakes in Mexico, *Tectonophysics* **687**, 66–77.
- Savran, W. H., J. A. Bayona, P. Iturrieta, K. M. Asim, H. Bao, K. Bayliss, M. Herrmann, D. Schorlemmer, P. J. Maechling, and M. J. Werner (2022). pyCSEP: A Python toolkit for earthquake forecast developers, *Seismol. Res. Lett.* **93**, 2858–2870.
- Shaffner, M. (2013). Programming for experimental economics: Introducing CORAL—a lightweight framework for experimental economic experiments, QuBE Working Papers 016, QUT Business School, Brisbane, Australia.
- Shcherbakov, R. (2014). Bayesian confidence intervals for the magnitude of the largest aftershock, *Geophys. Res. Lett.* **41**, 6380–6388.
- Shcherbakov, R., and D. L. Turcotte (2004). A modified form of Båth’s law bull, *Bull. Seismol. Soc. Am.* **94**, no. 5, 1968–1975.
- Shcherbakov, R., J. Zhuang, and Y. Ogata (2018). Constraining the magnitude of the largest event in a foreshock–mainshock–aftershock sequence, *Geophys. J. Int.* **212**, 1–13.
- Shcherbakov, R., J. Zhuang, G. Zöller, and Y. Ogata (2019). Forecasting the magnitude of the largest expected earthquake, *Nat. Commun.* **10**, doi: [10.1038/s41467-019-11958-4](https://doi.org/10.1038/s41467-019-11958-4).
- Southern California Earthquake Center (SCEDC) (2013). Caltech. Dataset doi: [10.7909/C3WD3xH1](https://doi.org/10.7909/C3WD3xH1).
- Tahir, M., J. R. Grasso, and D. Amorese (2012). The largest aftershock: How strong, how far away, how delayed? *Geophys. Res. Lett.* **39**, L04301, doi: [10.1029/2011GL050604](https://doi.org/10.1029/2011GL050604).
- Uhrhammer, R. (1986). Characteristics of northern and central California seismicity, *Earthq. Notes* **57**, no. 1, 21.
- Utsu, T., Y. Ogata, and R. S. Matsu’ura (1995). The centenary of the Omori formula for a decay law of aftershock activity, *J. Phys. Earth* **43**, 1–33.
- Vorobieva, I. A. (1999). Prediction of a subsequent large earthquake, *Phys. Earth Planet. In.* **111**, 197–206.
- Vorobieva, I. A., and G. F. Panza (1993). Prediction of the occurrence of related strong earthquakes in Italy, *Pure Appl. Geophys.* **141**, 25–41.
- Wiemer, S. (2001). A software package to analyze seismicity: ZMAP, *Seismol. Res. Lett.* **72**, 373–382.
- Wilson, G., D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, et al. (2014). Best practices for scientific computing, *PLoS Biol.* **12**, no. 1, e1001745, doi: [10.1371/journal.pbio.1001745](https://doi.org/10.1371/journal.pbio.1001745).
- Woessner, J., and S. Wiemer (2005). Assessing the quality of earthquake catalogues: Estimating the magnitude of completeness and its uncertainty, *Bull. Seismol. Soc. Am.* **95**, 684–698.
- Zhou, Y., R. Lin, Y.-W. Kuo, J. J. Lee, and Y. Yuan (2021). BOIN Suite: A software platform to design and implement novel early-phase clinical trials, *JCO Clin. Cancer Inform.* **5**, 91–101, doi: [10.1200/CCI.20.0012291](https://doi.org/10.1200/CCI.20.0012291)
- Zhuang, J., C.-P. Chang, Y. Ogata, and Y.-I. Chen (2005). A study on the background and clustering seismicity in the Taiwan region by using a point process model, *J. Geophys. Res.* **110**, no. B5, doi: [10.1029/2004JB003157](https://doi.org/10.1029/2004JB003157).
- Zhuang, J., A. Christophersen, M. K. Savage, D. Vere-Jones, Y. Ogata, and D. D. Jackson (2008). Differences between spontaneous and triggered earthquakes: Their influences on foreshock probabilities, *J. Geophys. Res.* **113**, no. B11, doi: [10.1029/2008JB005579](https://doi.org/10.1029/2008JB005579).
- Zhuang, J., and Y. Ogata (2006). Properties of the probability distribution associated with the largest event in an earthquake cluster and their implications to foreshocks, *Phys. Rev. E* **73**, doi: [10.1103/PhysRevE.73.046134](https://doi.org/10.1103/PhysRevE.73.046134).
- Zhuang, J., Y. Ogata, and D. Vere-Jones (2002). Stochastic declustering of space-time earthquake occurrences, *J. Am. Stat. Assoc.* **97**, 369–380.
- Zhuang, J., Y. Ogata, and D. Vere-Jones (2004). Analyzing earthquake clustering features by using stochastic reconstruction, *J. Geophys. Res.* **109**, no. B5, doi: [10.1029/2003JB002879](https://doi.org/10.1029/2003JB002879).