Deliverable

D8.9 Operational Earthquake Forecast Services in Italy, Switzerland and Europe

Deliverable information	
Work package	D8.9 OEF services in Italy, Switzerland and Europe wide operational setup
Lead	Eth Zurich
Authors	Laura Sarson, ETH Zurich Nicolas Schmid, ETH Zurich Leila Mizrahi, ETH Zurich Marta Han, ETH Zurich Warner Marzocchi, UNINA Naples
Reviewers	Laurentiu Danciu, ETH Zurich
Approval	[Management Board]
Status	[Draft]
Dissemination level	[Public/internal]
Delivery deadline	31.08.2023
Submission date	23.05.2023
Intranet path	DOCUMENTS/DELIVERABLES/ Deliverable 8.9

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 821115

Table of Contents

Summary

This deliverable is to summarize and document the development of the operational earthquake forecast services for Switzerland, Italy, and Europe. At the core of these operational services is RT-RAMSIS (Real Time Risk Assessment and Mitigation for Induced Seismicity) software, designed, customized, and developed within the project, along with other operational tools to deliver results for general use.

1. RT-RAMSIS Overview

The enhanced software RT-RAMSIS (Real Time Risk Assessment and Mitigation for Induced Seismicity) is the core of the system developed within the RISE project. Leveraging the diverse range of forecast models created in Work Package 3, with RT-RAMSIS allows to establish a modular and adaptable framework. This framework will facilitate the seamless integration of one or multiple OEF codes for a specific region or country, enabling the construction of dynamically weighted ensemble models (as outlined in WP3). The results will be effectively displayed and broadcasted, either in a web-platform or stand-alone solutions such as Jupyter notebooks.

RT-RAMSIS package is designed to run, configure, and orchestrate Seismicity Forecast Models (SFMs). A SFM is a model that takes input data such as an earthquake catalogue, other data (seismogenic sources, active faults) or information (i.e., completeness) that may be incorporated by a model, and produces forecast seismicity rates for selected periods of time. The earthquake forecast is spatially and temporally variable, and it can be represented may be discretized magnitude and recurrence rates at specific locations. Optionally the forecast may be provided as a stochastic catalogue, or parameters describing the statistical probability of an event occurring.

The package is a python-based software that has the following main dependencies:

- SQLAlchemy: The SQLAlchemy ORM (object relational mapper) is used to load, save and interact with database entries as python objects.
- GeoAlchemy2: This adds spatial capabilities to the PostgreSQL database used.
- Prefect: This is a workflow orchestrator that is used for scheduling and managing tasks. A server and agent are run separately in the background to manage the running of forecasts.
- Marshmalow: takes care of the conversion of data between sqlalchemy objects and json format when data is transferred via web service
- Typer: is used for the command line interface.
- Ramsis.utils: is a library share with other RT-RAMSIS based software that provides cross-application functions (https://gitlab.seismo.ethz.ch/indu/ramsis.utils).
- Ramsis.datamodel: is a library that provides the SQLAlchemy datamodel and serialization functions: https://gitlab.seismo.ethz.ch/indu/ramsis.datamodel/-/tree/feature/48-restructure

A seismicity model worker must also be running which allows remote running of a seismicity model on demand by RT-RAMSIS. This requires the following to be running on a machine with hardware appropriate to the level of usage required by the model:

- A 'model worker' web service (https://gitlab.seismo.ethz.ch/indu/ramsis.sfm.worker/-/tree/feature/rise_updates).
- A model available in the same environment as the web service is running in. In this instance, we are using the ETAS model code (https://github.com/swiss-seismological-service/etas).
- A 'model wrapper' which translates data to the format required by the model and also results into the format required by RT_RAMSIS (https://gitlab.seismo.ethz.ch/indu/ramsis-nsfm/-/tree/main)

The solutions to these challenges will become clearer as RT-RAMSIS is used more widely.

As stated before, RT-RAMSIS was initially designed for induced seismicity models, however the transition from induced to natural seismicity is not a great leap. The software now handles both cases with a simple set of configuration parameters.

1.1 Operation Setup

The most recent code that we are using for this project is on a branch that is currently in the review process, so throughout the document links have been given referring to this branch 'ramsis2-0'. When this branch has been merged in the coming weeks, it will exist on the 'main' branch.

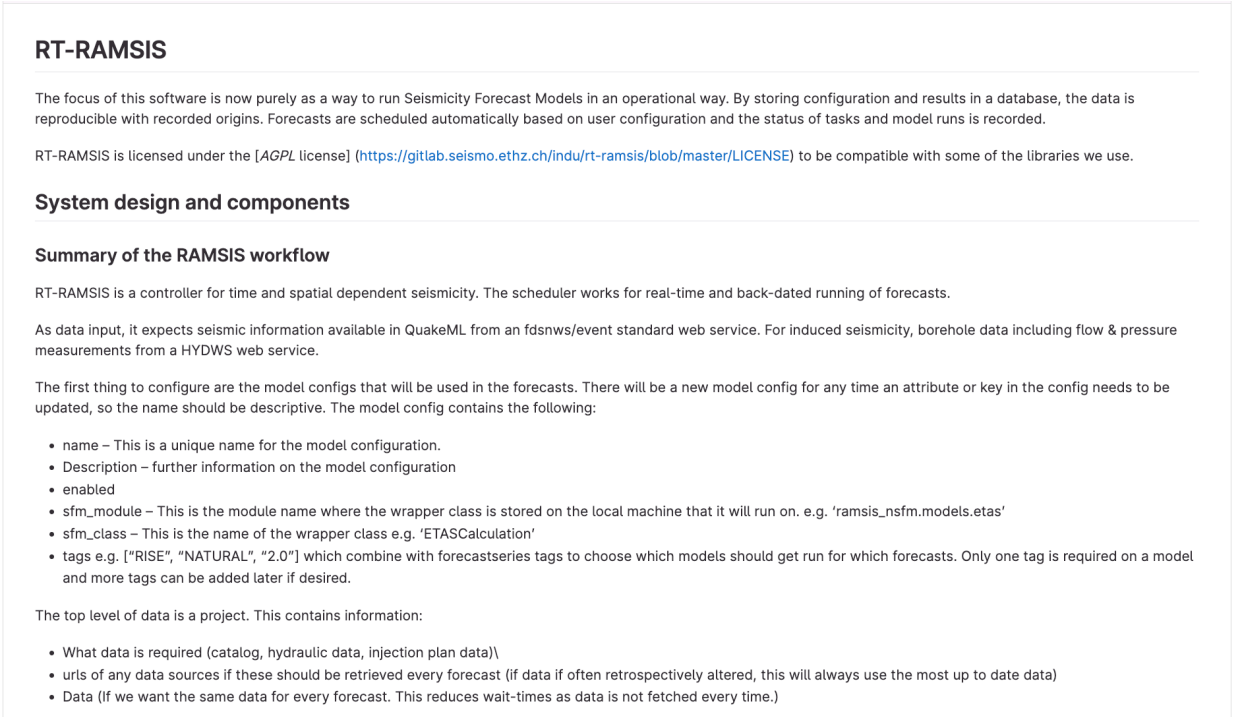
When the GitLab branch is finalized and is running stably, the main branch will be tagged to indicate a release of the code. This ensures that a specific version of RT-RAMSIS is being used by creating a snapshot of the code, and this is that pattern that will be followed for all upcoming releases.

Other dependencies such as ramsis.datamodel and ramsis.utils will also be tagged specifically for this project.

1.2 Documentation

The readme of RT-RAMSIS is provided at the following address:
https://gitlab.seismo.ethz.ch/indu/rt-ramsis/-/blob/feature/ramsis2-0/README.md

The installation instructions are described, and an overview of the configuration and running is given. For more details about the software and RISE specific developments, refer to Deliverable 8.6.



RT-RAMSIS

The focus of this software is now purely as a way to run Seismicity Forecast Models in an operational way. By storing configuration and results in a database, the data is reproducible with recorded origins. Forecasts are scheduled automatically based on user configuration and the status of tasks and model runs is recorded.

RT-RAMSIS is licensed under the [*AGPL* license] (https://gitlab.seismo.ethz.ch/indu/rt-ramsis/blob/master/LICENSE) to be compatible with some of the libraries we use.

System design and components

Summary of the RAMSIS workflow

RT-RAMSIS is a controller for time and spatial dependent seismicity. The scheduler works for real-time and back-dated running of forecasts.

As data input, it expects seismic information available in QuakeML from an fdsnws/event standard web service. For induced seismicity, borehole data including flow & pressure measurements from a HYDWS web service.

The first thing to configure are the model configs that will be used in the forecasts. There will be a new model config for any time an attribute or key in the config needs to be updated, so the name should be descriptive. The model config contains the following:

- name – This is a unique name for the model configuration.
- Description – further information on the model configuration
- enabled
- sfm_module – This is the module name where the wrapper class is stored on the local machine that it will run on. e.g. 'ramsis_nsfm.models.etas'
- sfm_class – This is the name of the wrapper class e.g. 'ETASCalculation'
- tags e.g. ["RISE", "NATURAL", "2.0"] which combine with forecastseries tags to choose which models should get run for which forecasts. Only one tag is required on a model and more tags can be added later if desired.

The top level of data is a project. This contains information:

- What data is required (catalog, hydraulic data, injection plan data)\
- urls of any data sources if these should be retrieved every forecast (if data if often retrospectively altered, this will always use the most up to date data)
- Data (If we want the same data for every forecast. This reduces wait-times as data is not fetched every time.)

2. RT-RAMSIS Configuration for ETAS (Epidemic-Type Aftershock Sequence) algorithm

The following describes setting up the configuration used by RT-RAMSIS for running the ETAS Epidemic-Type Aftershock Sequence model for Switzerland. The configuration for the project is in the following folder:

https://gitlab.seismo.ethz.ch/indu/rt-ramsis/-/tree/feature/ramsis2-0/rise/config

This contains three configuration files which are configured for operational running of a single model on a daily schedule. The configuration parameters are described in the readme.md on the repository. In the following file, some simple commands are given to set up a model, project and forecast series.

The paths used are specific to the ramsis-rise.ethz.ch machine setup, and a different user will need to modify these: https://gitlab.seismo.ethz.ch/indu/rt-ramsis/-/blob/feature/ramsis2-0/rise/setup_rise.sh , where some simple commands are given to set up a model, project and forecast series. The paths used are specific to the ramsis-rise.ethz.ch machine setup, and a different user will need to modify these.

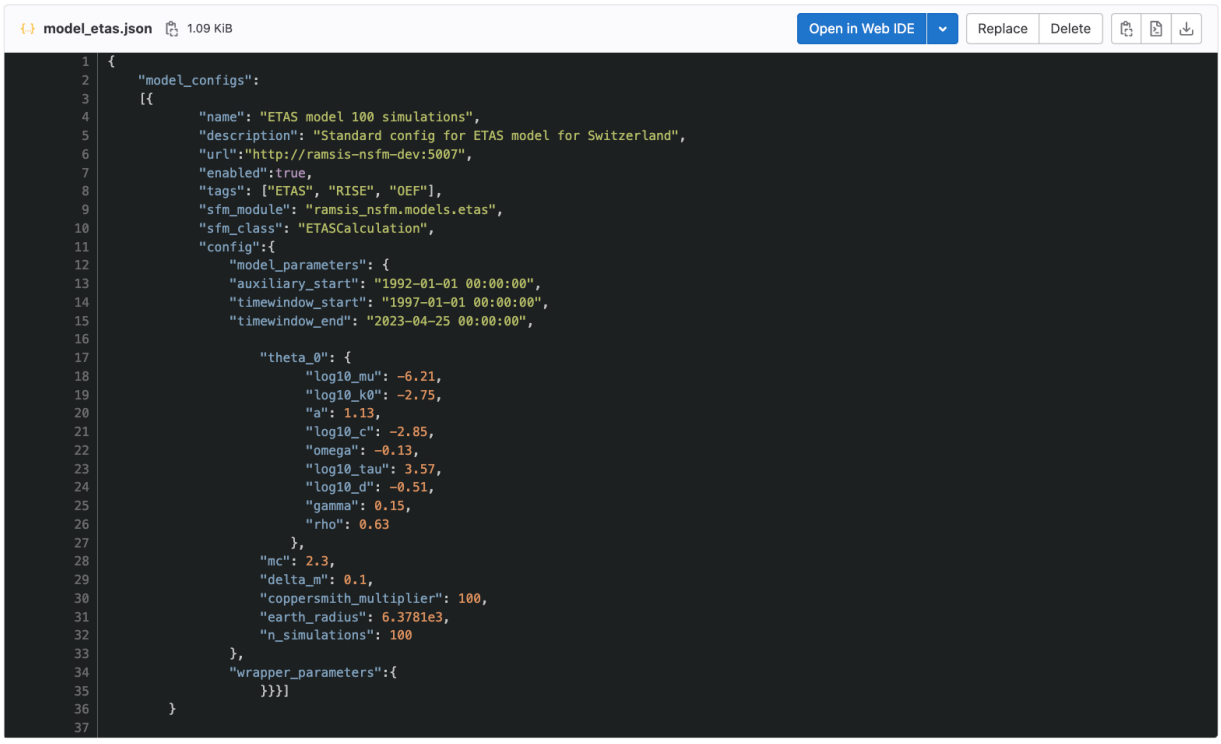


Figure 1: An example of ETAS configuration parameters for Switzerland

This file can be run as described in the file itself.

2.1 Configuration specifics for ETAS

- The area of interest for forecasts is defined as a bounding box around Switzerland. This is defined in the forecast series config as the '*geometryextent*'
- The time that forecasts will run is at 5.00 UTC which is defined in the '*starttime*'
- The period of time being forecast for is approximately 1 years (assuming 1 year has 365 days) and is described in seconds in the '*forecastduration*' parameter.

- The model is defined by the URL on which the ramsis-sfm-worker is running, and the module and class name of the model entry point.
- The configuration given to the model is stored in the 'config' parameter. This is defined as a nested dictionary which provides all the configuration parameters required to run the ETAS model successfully.
- Tags are stored on the models and the forecast series. This defines which models will be run for a series of forecasts. The model must contain at least one of the same tags as on the forecast series to run the model. These tags can be updated on the forecast series or the model at any time in the future, so more models be run with little effort.
- Catalogs will be used from the http://arclink.ethz.ch/fdsnws web service.
- The ETAS model will be run with two sets of configuration parameters. One set of parameters will calibrate the model between 2010-2012 with a magnitude of completeness (mc) of 1.8. The other configuration will calibrate the model between 1992-1997 with an mc=2.3.

These specifics are subject to change under the guidance of author scientists and stakeholders. Any change to the configuration will be version controlled.

3. Operational Earthquake Forecasting for Italy and Europe

A similar setup can be used for running models for different regions.

3.1 RT-RAMSIS setup for further models and regions.

The same installation of RT-RAMSIS would be used to run forecasts under a new 'project' and the FDSNWS that would be used for these regions are: https://gitlab.seismo.ethz.ch/indu/ramsis-nsfm

A new virtual machine will be setup for this model. These models should be started at the same time for direct comparison of results, and multiple models on the same VM may create competition for resource. Daily running of these new models is also proposed.

3.2 Italy

The OEF Italy system (Marzocchi et al., 2014) is aimed at estimating, in real time, the probability of occurrence of seismic events and ground shaking in Italy. These estimates are expressed in terms of weekly probabilities of occurrence (i.e., in the week following the time instant in which the calculation starts) in the cells of an equally spaced grid (0.1° x 0.1°), covering the entire national territory, excluding Sardinia and Etna region. De facto, the forecasts may be strongly affected in the Etna area, due to the particular tectonic structure, which would require the use of ad hoc algorithms. The exclusion of Sardinia is due to the negligible seismicity of the area and the poor local coverage of the seismic network.

The OEF Italy forecasting scheme consists of an ensemble of forecasts produced by three different clustering models, two different flavors of ETAS, and one STEP model. The ensemble modeling weighs the forecasts of each single model according to its past forecasting performance and allows proper estimation of the epistemic uncertainty that is of paramount importance for testing (17) and for communicating uncertainties to the decision-makers.

Each OEF Italy run produces five kind of forecasts (that are mapped on the Italian region):

- probability of occurrence of one or more seismic events with ML>=4.0 and ML>=5.5 (ML, local magnitude) in each cell of the spatial grid.
- probability of occurrence of one or more seismic events with MMI>=VI, MMI>=VII and MMI>=VIII (MMI, modified Mercalli macroseismic intensity) in each cell of the grid.

These forecasts are generated every day (at 00:00) and immediately after one or more earthquakes (if entered in the database in a time interval of less than 15') of magnitude ML>=3.5.

OEF Italy also produces the forecasts in each spatial region that has been selected by the end-user. It also allows you to graph the time trend of the probabilities described above, for a single cell or for a larger area defined by the user. The probability estimates displayed run from April 16, 2005.

The reliability of the estimates produced can be significantly reduced immediately after a strong event (ML>=5.5), due to the increase of the magnitude detection threshold of the seismic network and the consequent incompleteness of data in the seismic catalogue. In particular, the probabilities produced in the days immediately following the event could be affected by an underestimation. So far, these forecasts have been revised and updated by operators in the first hours after a major earthquake.

The estimated probabilities may be subject to local instantaneous variations, more visible in the maps relating to the ML parameter, not directly attributable to the real occurrence of seismic events. They derive from random fluctuations intrinsic to the stochastic models used and do not, however, lead to a significant alteration of the expected probability.

It is possible to visualize the data produced through a graphical interface, which can be queried by the user.

3.3 Switzerland

The WP3/WP8 teams at ETH Zurich developed and tested an earthquake forecasting model for Switzerland as part of the RISE project (Mizrahi et al. 2023, in prep.). This earthquake forecast model aims to broaden its approach to earthquake forecasting in Switzerland in order to supplement the current time-independent (i.e. long-term) earthquake forecasts (see SUIhaz2015). Earthquakes typically occur in clusters, resulting in significant temporal fluctuations in their rate that are not reflected in such long-term forecasts. The goal of the Operational Earthquake Forecasting (OEF) project is to provide operationally updated earthquake probabilities that take into account both recent and long-term data. This new model will form the foundation of the operational services for earthquake forecast in Switzerland.

3.4 Europe

Similarly with the efforts for generating an earthquake forecast model and service for Switzerland, a pan-European forecast model was also initiated during the RISE project. This ETAS based model is calibrated using the European earthquake catalog used in the development of the 2020 European Seismic Hazard Model (i.e, ESHM20, Danciu et al., 202). Calibration was performed utilizing the Python library developed by Mizrahi, Schmid, and Han (2023), which can in theory be used to calibrate basic ETAS models on any given earthquake catalog. This model will be the reference model to begin with the operational phase of the OEF across the Euro-Mediterranean region.

4. Cassandra – an OEF System for Italy

Within the WP8 activities, the OEF system was installed and tested as an alternative system for assessing the earthquake forecast in Switzerland.

An example of the forecast for Switzerland is given in Figure 2. Cassandra package (OEF Italy) was installed at ETH Zurich, adapted for use in Switzerland. The package consists of a compilation of codes written in different languages, Fortran90, MATLAB and Python. The code is maintained and developed by the INGV Italy.

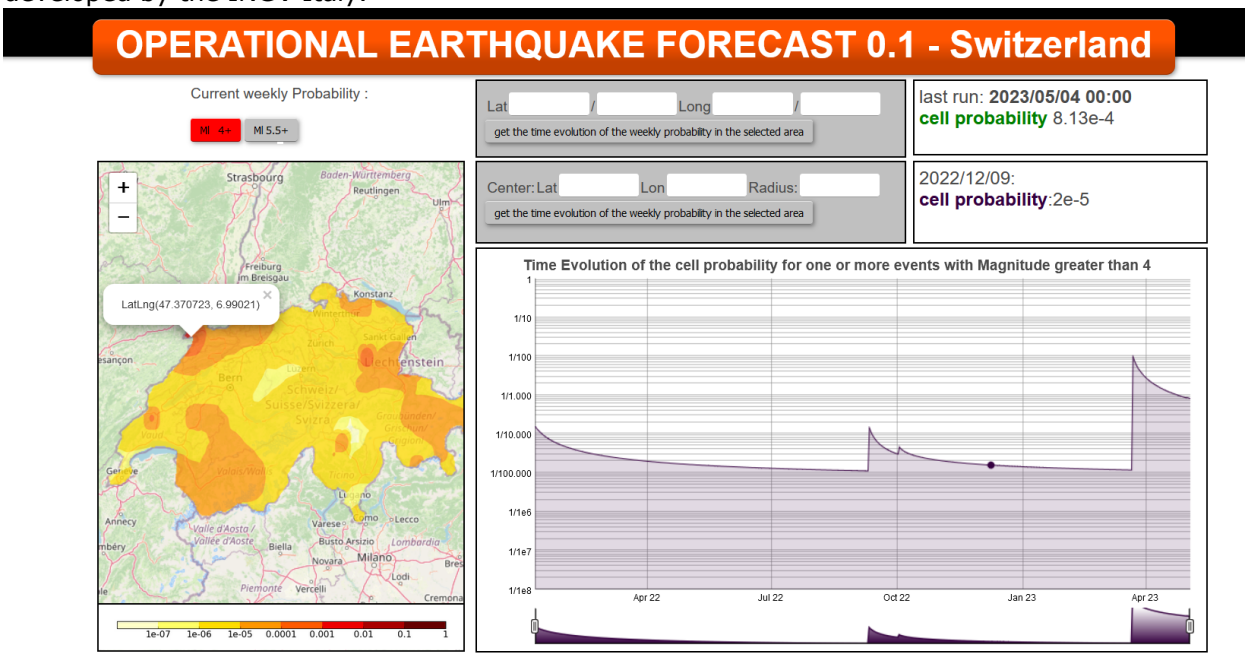


Figure 2: Display of the Operational Earthquake Forecast for Switzerland

5. Running the Operational Forecast

A forecast series has been set up and scheduled, which will run automatically on a daily basis. The earthquake forecasts will continue running until stopped by the user.

Further commands used to control the forecasts being run and to view information about them can be found in the readme at: https://gitlab.seismo.ethz.ch/indu/rt-ramsis/-/tree/feature/ramsis2-0

5.1 Operational running

The RT-RAMSIS software has been set up on the virtual machine: ramsis-rise.ethz.ch
Dependencies such as the prefect server and agent which are used for scheduling are set up with *supervisord* (http://supervisord.org).

This is a system and service manager that handles the running of the services in the background of the VM and restarting the services on failure and logging.

The PostgreSQL database is provided by a docker container, which uses a docker volume to persist data. This has the benefit of being easy to backup or migrate if it is required. This database is initialized on the first *ramsis* command that is run.

The ramsis-sfm worker has been setup on the virtual machine ramsis-nsfm-dev.ethz.ch on port 5007.

This '*development'* machine will be used to further determine the full requirements of running the ETAS and further similar models in operational mode for this project.

These requirements include disk space, RAM and parallel processing requirements. This information gathering will lead to setting up a further virtual machine which will be named ramsis-nsfm.ethz.ch which provides a means to test bug fixes and new releases in the future, as well as redundancy in running the software.

5.2 Operational Logging

Logging of the earthquake forecast runs is stored at ramsis-rise.ethz.ch:/home/ramsis/repos/rt-ramsis/ramsis.log

Where the log level captured is INFO. This logging is captured by the *prefect* agent which runs the forecast, and captures stdout and stderr from RT-RAMSIS.As well as logging errors produced by the model worker.

Logging of the *prefect* engine can be found at ramsis-rise.ethz.ch:/home/ramsis/repos/rt-ramsis/prefect_server.log which will log information relating to the failures and clean-up of *prefect* flows.

This should also capture any errors given by RT-RAMSIS and errors captured by the model worker.

Logging of the model can be found at ramsis-nsfm-dev.ethz.ch:/home/ramsis/repos/ramsis.sfm.worker/app.log

5.3 Operational Backup

Backup of the database is easily managed by creating an archive of the docker volume that stores the data from the PostgreSQL container. This archive is stored as a *tarball* and can be easily restored. The requirements of how frequently to back up the database will become clearer as the volume of data stored and how long it would take to reproduce becomes known.

One of the benefits of RT-RAMSIS is that all the forecasts that have been made are reproducible given that the data sources are available.

5.4 Testing

A full end-to-end test is provided in the RT-RAMSIS testing suite. This provides data and mocks the various web services that are required for a forecast to complete successfully. To run the tests, navigate to the RAMSIS directory and run '*pytest'*.

As much of the code has recently been changed to meet updated requirements, unit tests are not yet available.

5.5 Results and Visualization

While there are no web-based visualizations available yet, the results are freely accessible through the ramsis web service at http://ramsis-rise.ethz.ch:8000/v1.

Since the webservice, like the rest of the software, is still in review, there is no official documentation available yet. To show and explain the results from the OEF system there has been a Jupyter Notebook created which shows many different aspects of the platform as well as the options and possibilities which are created by this system.

The notebook is called `oef_switzerland.ipynb` (screenshot in Figure 3) and is available, including a small readme, here:
https://gitlab.seismo.ethz.ch/indu/ramsis-ws/-/tree/main/examples

The Visualizations are based on the ones described in Deliverable 8.6. All further discussion concerning the results, including ensemble models and testing, is done in that notebook.

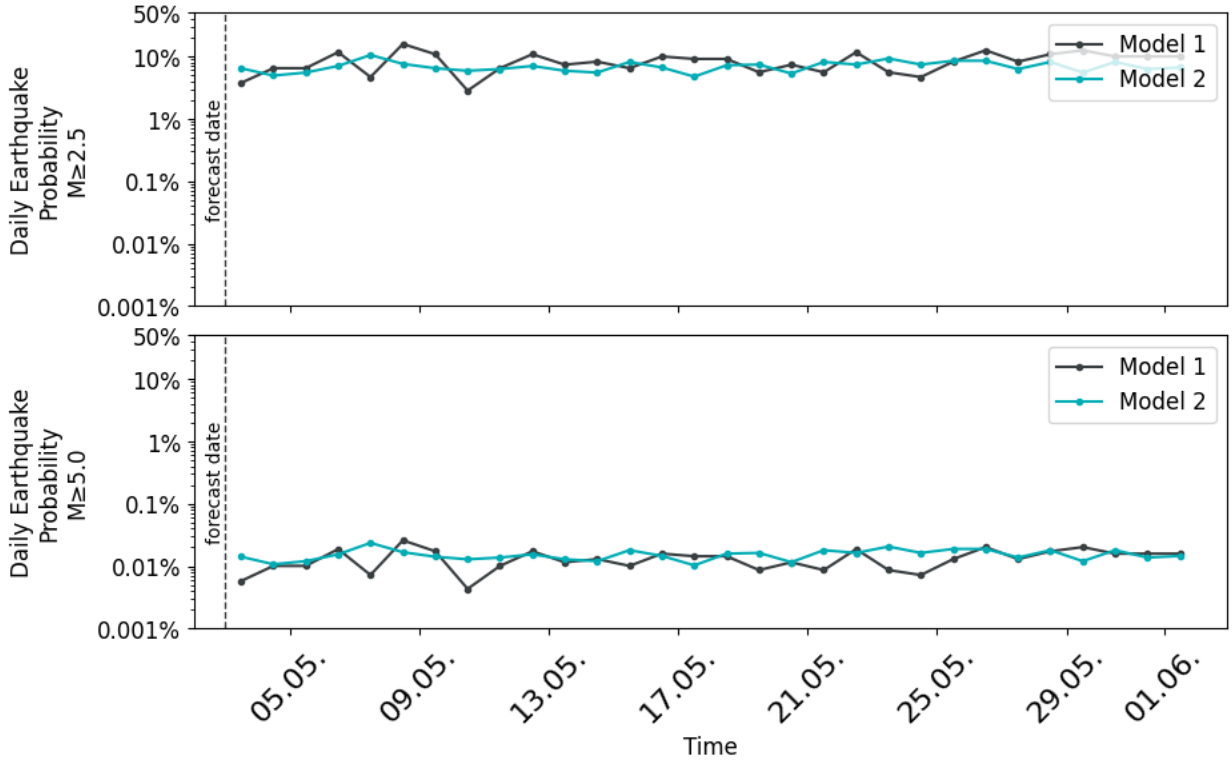


Figure 4: Daily earthquake probabilities of M≥2.5 and M≥5.0 for Switzerland (demonstrator only)

6. References

Danciu, Laurentiu, Graeme Weatherill, Andrea Rovida, Roberto Basili, Pierre-Yves Bard, Céline Beauval, Shyam Nandan, et al. 2022. "The 2020 European Seismic Hazard Model: Milestones and Lessons Learned." In Progresses in European Earthquake Engineering and Seismology, edited by Radu Vacareanu and Constantin Ionescu, 3–25. Springer Proceedings in Earth and Environmental Sciences. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-15104-0_1.

Han, Marta, Leila Mizrahi, Stefan Wiemer. 2023. "Towards Operational Earthquake Forecasting for Europe" In preparation.

Marzocchi, Warner, Anna Maria Lombardi, and Emanuele Casarotti. 2014. "The Establishment of an Operational Earthquake Forecasting System in Italy." Seismological Research Letters 85 (5): 961–69.

Mizrahi, Leila, Shyam Nandan, Stefan Wiemer. 2023. "Developing and Testing ETAS-Based Earthquake Forecasting Models for Switzerland." In preparation.

Mizrahi Leila, Schmid Nicolas, & Han Marta (2023). lmizrahi/etas: ETAS with fit visualization (3.2). Zenodo. https://doi.org/10.5281/zenodo.7584575

RT-RAMSIS Dependencies

lxml>=3.3.3
marshmallow
numpy>=1.22.3
obspy>=1.3.0
PyOpenGL>=3.1.1a1
PyQt5
python-dateutil>=2.8.0
PyYAML>=5.1.1
ramsis.datamodel @ git+https://gitlab.seismo.ethz.ch/indu/ramsis.datamodel.git
ramsis.utils @ git+https://gitlab.seismo.ethz.ch/indu/ramsis.utils.git
requests>=2.18.4
transitions==0.6.9
prefect==0.15.10
pyproj>=3.2.1
psycopg2==2.8.3
jinja2
xmltodict
sqlalchemy>=1.4
typer
python-dotenv

RT-RAMSIS Copyright