DEEP

Innovation for:
De-Risking Enhanced
Geothermal Energy Projects

Deliverable

Deliverable 3.2 - Development of Machine Learning based induced seismicity forecasting models

Report information	
Work package	WP3 Innovation in forecasting models and uncertainty quantification
Lead	ETH
Authors	Arnaud Mignan, Federica Lanza, Antonio Rinaldi
Reviewers	Nori Nakata
Approval	Luigi Passarelli
Status	Draft
Dissemination level	Internal
Will the data supporting this document be made open access?	Yes
If No Open Access, provide reasons	-
Delivery deadline	30.11.2022
Submission date	30.03.2022
Intranet path	[DOCUMENTS/DELIVERABLES/Deliverable3.1.docx]

DEEP Project Office | ETH Department of Earth Sciences
Sonneggstr. 5, 8092 Zurich | deep-office@sed.ethz.ch | www.deepgeothermal.org

This project has been subsidized through the Cofund GEOTHERMICA, which is supported by the European Union's HORIZON 2020 programme for research, technological development and demonstration under grant agreement No 731117.

Table of Contents

.

1. **Part 1:** Review of existing ML models and feature extraction specific to forecasting seismicity

A review of published studies on Machine Learning (ML) for seismicity forecasting is a necessary preliminary step for the use of state-of-the-art techniques and for new developments in the context of induced seismicity traffic light systems (TLS). ML applications in Seismology took hold after the mid 2010's, a few years after the start of a new "Artificial Intelligence" era in computer science. Most earthquake applications have however focused on computer vision and other deep learning techniques dedicated to seismic waveform analysis. The number of ML applications to natural seismicity forecasting remains limited and to induced seismicity very scarce despite a significant increase in the number of studies in the past few years. Most studies are so far limited to the spatial prediction of induced seismicity in regions of fracking and wastewater disposal. The logistic regression is commonly used, when from a few to two dozen features are defined (based, e.g., on operational and geological data). When up to hundreds of features are engineered, decision tree ensembles are preferred for their flexibility. Time series forecasting remains limited to the natural seismicity regime and laboratory experiments with decision tree ensembles and various types of neural networks being employed. In this case, the main features are statistical in nature (e.g., nth-order moments, temporal correlation, etc.). New applications for ML-based TLS, in particular in the context of Enhanced Geothermal System (EGS) stimulation, can benefit from the tens of features already defined or engineered in previous studies. Since there is so far no preferred ML method nor many, if any, applied to EGS TLS, there is a lot of flexibility for the development of new ML applications. We recommend testing a broad set of ML techniques (especially from the Generalized Linear Model – GLM - family), from the simplest ones to more complex ones depending on the number of features defined, data characteristics, available model training sample size, and inherent constraints linked to (pseudo-)prospective forecasting in a TLS.

1.1 Introduction

The present report provides a review of both feature extraction approaches and Machine Learning (ML) models available in the scientific literature, which can help guide the choice of the best ML-based forecasting strategies in the context of induced seismicity hazard. It considers the specificities of seismicity time-series and their variations in space, and therefore the ML pipelines developed for both natural seismicity (precursory seismicity, aftershock sequences) and induced seismicity in different stimulation regimes (fracking, gas extraction, geo-energy, but also lab experiments). In contrast to natural seismicity analysis, induced seismicity forecasting must consider the direct role of operational parameters, which are known in advance or in real-time. However, in both cases, the black-box nature of tectonic loading also plays a role.

While the different ML methods to be described are applicable to any type of seismicity, feature engineering depends on the problem to be solved. As such, the reviewed methods which have been developed for different seismicity environments are discussed in terms of the degree of applicability to the enhanced geothermal system (EGS) case. Since the focus of DEEP deliverable 3.2 of the DEEP project is to implement an ML version of an advanced traffic light system (ATLS), discussions are tailored to include the constraints associated to an ATLS.

This Part is written in such a way that it could be directly included in the introduction and background section of a future scientific article. It is timely since ML applications in Seismology are relatively recent (Kong et al., 2018; Bergen et al., 2019; Beroza et al., 2021; Johnson et al., 2021) and applications to induced seismicity, especially for TLS and/or EGS, remain rarely reported (Atkinson et al., 2020; He et al., 2020). The scarcity of ML studies for induced seismicity forecasting was confirmed as recently as in 2021 (Yu et al., 2021), despite an exponential increase of applications since 2018 in the geothermal industry at large (Okoroafor et al., 2022).

We limit this review to the last 6 years (2017-2022), which can be considered the start of the modern ML era in Seismology. However, a 2020 review of neural network applications in seismicity forecasting that go back to the 1990's is also included.

1.2 Rationale

Induced seismicity forecasting at EGS projects is at present accomplished using standard statistical methods based on a linear regression $y = cx$ where y (the target) is the induced seismicity rate and x (the variable, or feature) is the flow rate of the injected fluids (Fig. 1). This is the case of the model called EM1_*,

$$\lambda(t, m \geq m_0, \theta) = \begin{cases} 10^{a_{fb} - bm_0} \dot{V}(t) & \text{for } t \leq t_s \\ 10^{a_{fb} - bm_0} \dot{V}(t_s) \exp\left(-\frac{t - t_s}{\tau}\right) & \text{for } t > t_s \end{cases}, \quad (1)$$

where λ is the induced seismicity rate (i.e., the target y), $\dot{V}$ the injection flow rate (i.e., the variable x), and $\theta = (a_{fb}, b, \tau)$ the model parameter set (respectively the overall underground seismic feedback, the Gutenberg-Richter slope, and the post-injection relaxation time). Eq. (1) is constrained for data with magnitude m above the minimum threshold m_0 and time t in two windows, the injection phase (with $\dot{V} > 0$) and the post-injection phase starting after the shut-in time t_s (Mignan et al., 2017 with first-row equation derived from Shapiro et al., 2007). Note that the target y can also be the maximum magnitude m_{max}, which is then formalised from the Gutenberg-Richter distribution (e.g., van der Elst et al., 2016; Broccardo et al., 2017)

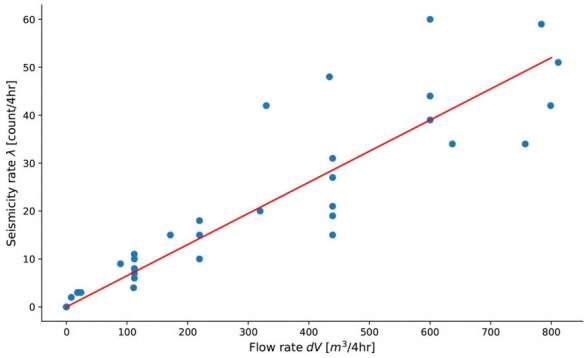

Figure 1. Linear relationship between injected flow rate and induced seismicity rate in the case of the 2006 Basel experiment with data binned in 4-hour intervals, flow rate digitized from Häring et al. (2008) and earthquake catalogue from Kraft and Deichmann (2014) (stimulation period with magnitude equal or greater than 0.8). The linear regression (in red) is the simplest model applicable to induced seismicity, here not considering the time component of Eq. (1).

The complexity of the model represented by Eq. (1) could be increased by implementing it in a Bayesian framework for instance (which was already done by Broccardo et al. (2017) – model EM1_BH instead of EM1_MLE in Deliverable 3.1 (Rinaldi & Passarelli, 2021) for statistical model comparison). Some of the parameters could also be turned into variables, e.g., $b(t)$. Other variables could be included with one obvious choice being the wellhead pressure $P(t)$ (main variable in physical models, such as HMo in Rinaldi & Passarelli, 2021). As the number of variables (or features) increases, more complex models – taken from a large panoply of ML techniques – could be tested. He et al. (2020), in their review on ML-based induced seismicity studies, mentioned a very limited number of such applications, and suggested using the Mignan et al. (2017) TLS method as initial framework from which ML methods should be developed. Since the relationships between the main parameters a_{fb} and b and the underlying physical parameters have yet to be validated (e.g., Dinske & Shapiro, 2013; Mignan, 2016 for a_{fb}; e.g., Bachmann et al., 2012; Spada et al., 2013 for b), defining many potential features and then let ML find the most relevant appears as a logical approach. Indeed, observations have so far been deemed "too heterogeneous and too few in number to allow firm conclusions to be drawn on the basis of single-parameter correlation with seismic response" (Evans et al., 2012), operational parameters have been judged of secondary importance to explain different induced seismicity behaviours (Göbel, 2016), and "for formations where no triggered seismicity has yet been observed, there is no accepted methodology to forecast the likelihood of an event" (Atkinson et al., 2020). It should already be noted that if observations may be too few for robust ML model training, simulations from physical models could be used instead (deliverable 2) with trained ML models acting as surrogates when tested on real data (deliverable 3). Distinction must also be made between spatial forecasts and temporal forecasts. If wide uncertainties in the seismic response a_{fb} exist between different sites (e.g., Mignan et al., 2021), this uncertainty is reduced at any specific location once data starts to be available in the early stage of a new stimulation (Broccardo et al., 2017). Those represent two different problems, requiring different ML protocols. Both will be reviewed in this report.

Although the project's focus is temporal forecasting during a TLS, site-specific conditions may be implemented in a prior for instance (e.g., as in EM1_BH).

Although we will need regression techniques to forecast the induced seismicity rate (as in Eq. 1), classification methods could also be employed with the target y possibly defined by classes representing the different alarm levels of a TLS (Broccardo et al., 2020). As such, all types of ML models will be reviewed in this report. Figure 2 shows an evolutionary tree of some of the main ML methods including their connections. This provides a synoptical view of the complex field of ML and can be summarised as follows: At the basis of any ML method is Probability Theory with its laws being describable in a simple Venn diagram. Bayes' Theorem directly follows leading to all Bayesian inference frameworks (lower right branch). Via the concept of entropy, decision trees and their ensembles can be defined (lower left branch). Going up, all probability distributions can be derived from different sets of assumptions including the Normal distribution. The linear regression (the workhorse of ML) can be probabilistically described as a Normal distribution, as well as non-linear regressions (central branch). From the linear regression, hyperplanes can be used for classification purposes via Support Vector Machines (SVM, upper left branch); changing the noise distribution from Normal to Bernoulli transforms the linear regression into a logistic regression for classification. A logistic regression is equivalent to one artificial neuron from which a variety of neural network architectures can be built (upper right branch) – Note that this probabilistic approach follows the model definitions given in Murphy (2012), but the evolutionary tree of Figure 2 was developed by the author of the present review report. In this figure, simpler models are nested in the higher-branch models, which are therefore more complex. If two models of different complexity perform the same, then the simplest model should always be preferred to reduce overfitting and increase interpretability (e.g., Mignan & Broccardo, 2019).

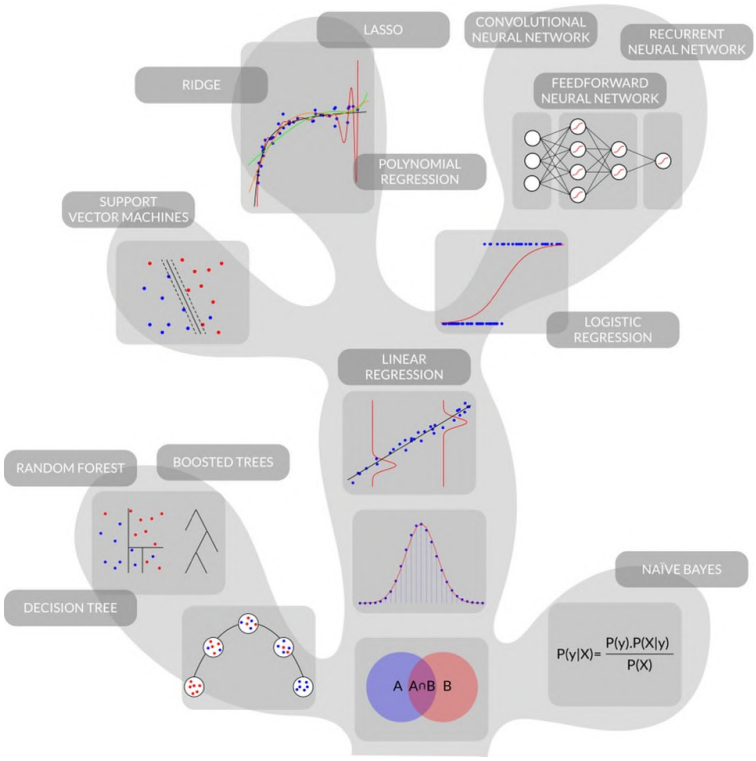

Figure 2. Evolutionary tree of ML techniques showing their statistical connections. Simpler models are nested in the higher-branch, more complex, models.

1.3 Feature extraction (and target definition) approaches

Feature extraction, which includes direct variable use and the engineering of new features derived from other observed variables, provides the input vector $X = (x_1, \cdots, x_n)$ to the chosen model $f(X)$. As for the output of the model y in the context of seismicity forecasting, it can be a value such as the seismicity rate, the maximum magnitude, an occurrence time, a category, binary (e.g., seismicity present or not) or multi-class (e.g., alarm scheme of a TLS). Since feature definition may relate to the chosen ML model, we will mention for which ML technique a given feature engineering

strategy was developed in each article. However, details on the matching ML models will be given in the next section (Sec. 4). The reviewed studies are listed by seismicity type and then in chronological order. Each study is a different bullet point with the main characteristics written in brackets for clarity with [n features x, model f, target y type]. As a side note, lessons could also be learned from physical models directly (e.g., Izadi & Elsworth, 2015) for feature engineering.

1.3.1 Natural seismicity case

Despite some differences between natural and induced seismicity, many features defined for natural seismicity forecasting could directly be used in the induced seismicity case. Moreover, some techniques of feature engineering developed for mainshock or aftershock prediction can provide additional insights since the ML literature for induced seismicity forecasting remains so thin.

1.3.1.1 Foreshock-mainshock sequences

Selected articles are considered, which exemplify a rich variety of techniques. The approaches used in other existing articles can be related to the ones shown below. The features described are often derived from seismicity observations which are expected to represent an exogenic process leading to a large earthquake (i.e., passive precursors) or some endogenic process of event interaction leading to a larger event (Table 1). In the context of induced seismicity forecasting, the exogenic process is known (i.e., the injection hydraulic data).

- **Asencio-Cortés et al. (2018)** [**16 features** | various | **magnitude regression**]: Based on many previous works by the same authors, this study forecasted the earthquake magnitude in the next seven days per grid cell in California, using various regression algorithms. Each grid cell being independent and representing one data sample, a set of 16 features was developed based on earthquake data: a- and b-values of the Gutenberg Richter law, 5 increments of b in successive time windows, maximum magnitude, probability of an event greater than 6, mean square deviation, magnitude deficit, elapsed time, mean time, coefficient of variation, rate of square root of seismic energy and mean magnitude.
- **Huang et al. (2018)** [**10,000s features** | convolutional neural network (CNN) | **spatial binary classification**]: This study is one of the only ones applying computer vision (i.e., CNN) to predict the location of large earthquakes of magnitude 6 or above ($y = 1$ if true). Typically, each pixel of the analysed image (e.g., seismicity rate) becomes a feature. In the case of earthquake catalogues, each grid cell acts as a pixel. It means that the number of features is very high. In the present study, the authors defined $256 \times 256 = 65536$ features. However, their number of samples was likely far too low for any robust CNN training.
- **Mignan & Broccardo (2020)** [various | neural networks | various - **REVIEW**]: This study reviewed the literature of artificial neural networks applications for earthquake prediction. It included 77 articles published in the period 1994-2019. They compiled the types of features used as input, which were derived from earthquake catalogue data (i.e., time, magnitude, location). From the magnitude distribution, derived features included: mean value, quantiles (including maximum) and seismicity law parameters such as the a- and b-values. Location variables included: longitude, latitude, depth distribution metrics and clustering metrics. From the temporal data, the following features were defined: seismicity rate, interevent time, aperiodicity, aftershock law parameters, and even the application of various metrics derived from the financial markets. Other types of features included: slip rate, static stress, as well as geo-electric, ionospheric, and geochemical data. The target would vary from study to study and could be: the maximum magnitude, the probability to exceed a given magnitude threshold, felt intensity, the location of the largest event or its occurrence time (see table 1 of Mignan & Broccardo, 2020). Figure 3 (their figure 2) illustrates a common method used by many studies to define features for neural networks. In addition to m selected metrics from the above list, the total number of features is increased to $m \times n_t$ with n the number of time bins considered. This allows to encode the potential dynamics of a hidden process. As n increases, the input data, which was originally structured/tabulated, becomes unstructured. This increase in complexity requires the use of more flexible models such as neural networks (feedforward or recurrent, see Sec. 4.2).

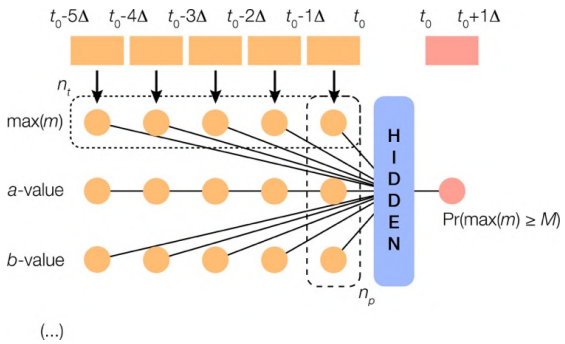

Figure 3. Figure reproduced from Mignan & Broccardo (2020) showing a common technique used in the neural network literature to create features for precursory seismicity time series analysis. The number of parameters (n_p) and time samples (n_t) define the size of the input layer.

1.3.1.2 Mainshock-aftershock sequences

Aftershock prediction using ML so far consists in forecasting their location. With the main features related to stress and to distance to relevant structures, they could be applicable to induced seismicity to forecast where induced seismicity could occur or not (see examples in Sec. 3.2).

- **DeVries et al. (2018)** [**12 features** | deep neural network (DNN) | **spatial binary classification**]: This study forecasted the location of aftershocks via a binary classification with $y = 1$ for geographical cells with at least one aftershock, or $y = 0$ otherwise. For their deep neural network, they defined a set of 12 features, the absolute value of the 6 components of the mainshock stress tensor plus their 6 negatives. It should be noted that this approach destroys the dipolar nature of the stress tensor information and that considering the negatives as additional features is of very limited use, if of any. This is illustrated in Figure 4 (top panel).
- **Mignan and Broccardo (2019)** [**1-2 features** | logistic regression | **spatial binary classification**]: This is a comment of the study by DeVries et al. (2018) in which the authors proved that a logistic regression could perform as well as a deep neural network. They reduced the number of features to one by using the sum of the absolute stress components, maximum shear stress or the von Mises yield criterion (Fig. 4, lower panel). They also tested the association of two other features: distance to mainshock rupture and mainshock slip.

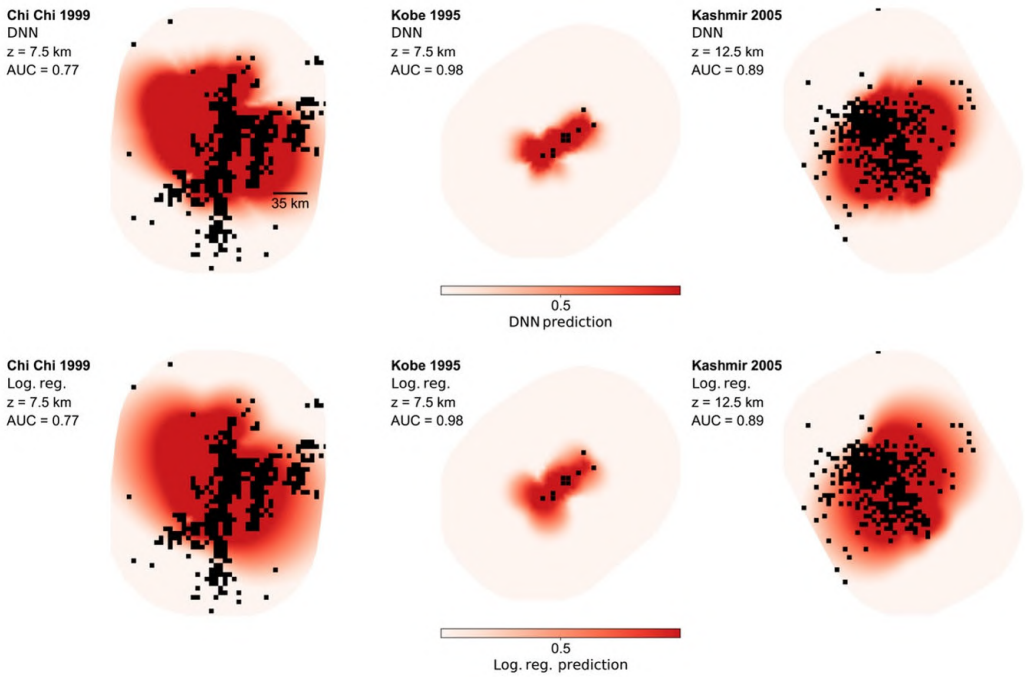

Figure 4. Heatmaps showing the expected locations of aftershocks, i.e., $\Pr(y = 1)$, following three different mainshocks. Top: Reproducing the results of DeVries et al. (2018) with 12 stress features as input X to a DNN. Bottom: Reproducing the results of Mignan & Broccardo (2019) with 1 stress feature, here the von Mises yield criterion.

Table 1. List of previously tested features in the natural seismicity context.

Explored feature $\mathbf{x_i}$	Description	References
Physical features		
Geochemical data	Gas content	M&B20
Geophysical data	Geo-electric, ionospheric, etc.	M&B20
Rheological data	Coulomb stress, maximum shear stress, von Mises yield criterion	M&B19, M&B20, DVetal18
Seismicity	Rate, a-value, b-value, maximum magnitude, location, inter-event time, clustering, aftershock law parameters, $\Pr(\mathrm{m})$, seismic energy	Aetal18, Hetal18, M&B20
Tectonic data	Slip rate, slip, distance to fault	M&B19, M&B20
Statistical features		
Moments	Mean, variance, skewness, kurtosis	M&B20, Aetal18
Quantiles	Minimum, median, maximum, etc.	M&B20, Aetal18
Thresholds	Various threshold values	
Various	Coefficient of variation, aperiodicity, financial market metrics	M&B20, Aetal18

1.3.2 Induced seismicity (IS) case

In their review, Atkinson et al. (2020) mentioned three fluid-injection processes that have been linked to induced seismicity: (i) hydraulic fracturing (Sec. 3.2.1), (ii) enhanced geothermal stimulation (Sec. 3.2.2), and (iii) saltwater disposal (Sec. 3.2.3). Here, we additionally include gas depletion (Sec. 3.2.4), and laboratory experiments (Sec. 3.2.5), as ML-based methods developed in those regimes may also be informative for the EGS case. Only methods forecasting seismicity are considered here, but ML is also useful for other forecasting such as shale gas production for instance (e.g., Hui et al., 2021). A summary list of features in the IS context is given in Table 2.

1.3.2.1 Fracking case

- **Pawley et al. (2018)** [**10 features** | logistic regression | **spatial binary classification**]: This work elaborates on previous studies which found some geological proxies that explain (in some cases) where induced seismicity can be located. Those are tectonic, geomechanical, and hydrological conditions. Their target is categorical (binary) with $\mathrm{y} = 1$ at locations exhibiting induced seismicity (with completeness magnitude $\mathrm{m} \geq 2.5$, in clusters of $\mathrm{n} \geq 10$ events occurring after onset of injection within 5 km of injection well surface location), and $\mathrm{y} = 0$ otherwise. The application was made in the Duvernay Formation in Canada. Using an information gain score, features in order of decreasing importance were: proximity to basement, Duvernay pressure depth ratio, minimum horizontal stress, distance to Upper Reef Edge, Lithium concentration, distance to platform edge, natural seismicity rate, distance to lineaments, Duvernay thickness and Dolomite occurrence.
- **Wozniakowska and Eaton (2020)** [**8 features** | logistic regression | **spatial binary classification**]: This study is similar in nature to Pawley et al. (2018) but applied to another area of Canada, the Montney Formation. Here, the feature importance ranking shows that the most informative variable is the distance to the Cordilleran Belt, followed by injection depth, vertical distance to Precambrian, pressure gradient, SH_{max} azimuth variance, vertical distance to Debolt, depth index and distance to faults.
- **Hicks et al. (2021)** [**16 features** | logistic regression | **spatial binary classification**]: This study estimated the likelihood of induced seismicity depending on location in the Permian Basin of SW United States (with $\mathrm{y} = 1$ if at least one magnitude 2.2 or above has occurred within 5 km, $\mathrm{y} = 0$ otherwise). An innovation is that it considered wide areas subject to fracking and wastewater disposal. The authors considered both industrial and geological factors. Their original 16 features include: depth-to-basement, minimum distance to faults, difference between orientation of maximum horizontal compressive stress and fault strike (as a proxy to optimal orientation to failure), well location and vertical total depth (with three depth classes defined), various fluid extraction and injection volume variables at a month-level resolution. Highly correlated features were then removed. The most significant features were found to be proximity to active faults, followed by shallow injection/extraction.

1.3.2.2 Geothermal (incl. EGS) case

- **Yu et al. (2021)** [**759 features** | unclear | **IS rate binary classification**]: This study provides a binary classification of the rate of induced seismicity during fluid injection at the Rotokawa geothermal field of New Zealand (with $y = 1$ if the rate crosses a specific threshold). The number of features is derived from 8 main physical parameters: the injection rate, well head pressure, hydraulic energy, total injection, and their respective derivatives. From these variables, various statistical features were defined leading to a total of 759 features. Those are mean, median, standard deviation, autocorrelation, entropy, regressor parameters, quantiles, and others (not mentioned). Their main conclusion is that there is correlation between long-term total injection and long-term averaged microseismicity rates. The role of using hundreds of features is not discussed.
- **Feng et al. (2022)** [**1-2 features** | linear regression (Bayesian) | **IS rate regression**]: This study (in revision, as of Nov. 2022) modified Eq. (1) to include the wellhead pressure as additional variable to the flow rate to forecast the rate of induced seismicity at different EGS sites. It also considered a time shift in feature values for application in real-time forecasting. For this specific case, the authors considered the (planned) flow rate expected in the future time bin but the pressure from the last observed bin.

1.3.2.3 Wastewater disposal

- **Hincks et al. (2018)** [**7 features** | Bayesian network | **seismic moment release regression**]: This study forecasted the annual seismic moment release at wastewater injection wells in the Arbuckle Group, Oklahoma. The considered the following features: well depth, depth relative to basement, distance to basement, monthly and annual volume, cumulative volume, and geospatial correlation (i.e., a proxy for unobserved geological data). The authors found that depth combined to volume was the most critical aspect.
- **Hicks et al. (2021)** [**16 features** | logistic regression | **spatial binary classification**]: See Sec. 1.3.2.1.
- **Qin et al. (2022)** [**12 features** | Random Forest | **IS rate regression**]: This study forecasted the rate of induced seismicity by using 12 features as model input. Rates for different time windows were tested (30, 60, 90, and 180 days). Features were ranked, from the most important to the least important one, as follows: Pore pressure rate, poro-elastic stress rate, pore pressure, injection PSI, cumulative volume, poro-elastic stress, (monthly) injection rate, 1-year injection volume, injection depth, basement depth, distance to basement and region parameters.

1.3.2.4 Gas depletion

- **Limbeck et al. (2021)** [**63 features** | various | **IS rate regression**]: In this study, the target y was the number of induced earthquakes within a given cell within a day (for magnitudes 1.5 or above). The spatiotemporal approach considered space and time in a similar way with each spatiotemporal cell one sample. The authors defined a total of 63 potential features and removed the most correlated ones (i.e., more-or-less copies of other variables). Too numerous to list here, examples of features included: fault characteristics, reservoir thickness, proportion of shale versus sandstone, gas column height versus water column height, porosity, compressibility, pore pressure characteristics, produced gas volume, etc. The study further compared the performance of different ML models but provided no ranking of the tested features.

1.3.2.5 Laboratory experiments

Laboratory experiments on rock fracturing, by definition, induce lab-quakes, and are closely related to in-situ injection experiments (e.g., Villiger et al., 2020).

- **Rouet-Leduc et al. (2017)** [**84 features** | Random Forest | **failure time regression**]: This study defined statistical features from strain data to predict the time until the next failure in a rock specimen under load. Their 84 features represented 42 features times two, as these statistical metrics were calculated for two successive time periods to include some potential temporal dynamics. The metrics are: statistical moments (mean, variance, skewness and kurtosis, normalized and not normalized for the last 3), the 1^{st}-to-9^{th} and 91^{st}-to-99^{th} percentiles, minimum and maximum and some thresholds and their opposites, and finally some time correlation features (Fourier transforms on different frequency bands, autocorrelation and partial autocorrelation). Figure 5 shows

some examples of statistical features using similar data to the study (as provided in the related Kaggle competition).

It should be noted that some of the statistical features defined in Rouet-Leduc et al. (2017), such as the statistical moments of different orders, are also commonly used in natural earthquake forecasting when using some unstructured and continuous data as input – for example geo-electric data (e.g., Chen et al., 2021).

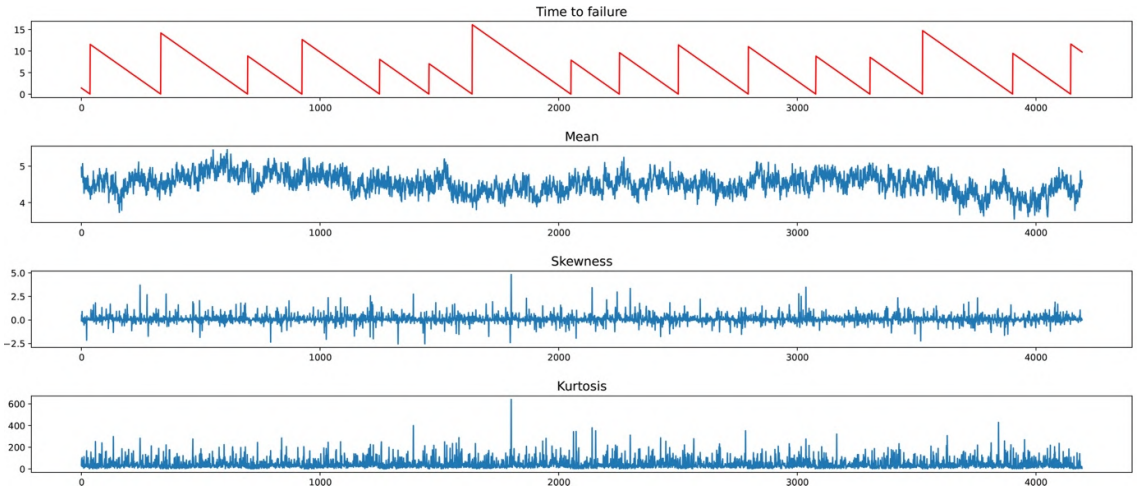

Figure 5. Target (in red) and three features (in blue) as defined in Rouet-Leduc et al. (2017).

Table 2. List of previously tested features in the induced seismicity context.

Explored feature x_i	Description	References
Physical features		
Geochemical data	Lithium concentration, gas saturation	Petal2018
Geological structures	Distance to different structures, structure thickness, occurrence or not, proportion of rock type	Hetal18, Petal2018, W&E20, Hetal21, Letal21, Qetal2022
Operational data	Injection depth, fluid injected (rate, cumulative volume), wellhead pressure, produced gas volume, including their derivatives	Hetal18, W&E20, Hetal21, Letal21, Fetal2022, Qetal2022
Rheological data	Pore pressure, stress, permeability, porosity, compressibility	Petal2018, W&E20, Letal21, Qetal2022
Seismicity	Natural seismicity rate	Petal2018, Letal21
Tectonic structures	Distance to faults, to lineaments	Petal2018, W&E20, Letal21
Statistical features		
Moments	Mean, variance, skewness, kurtosis	RLetal17, Yetal21
Quantiles	Minimum, median, maximum, etc.	RLetal17, Yetal21
Thresholds	Various threshold values	RLetal17
Temporal	Fourier transform, (partial) autocorrelation	RLetal17, Yetal21
Various	Entropy	Yetal21

1.4 Machine Learning models

ML methods are listed below by increasing complexity by following Figure 2. Only seismicity forecasting ML-models are considered. Models for seismicity detection and localisation are often very different, being based on computer vision (e.g., Ross et al., 2018). Such applications for induced seismicity characterisation have appeared in recent years (e.g., Holtzmann et al., 2018; Zhang et al., 2020; Shaheen et al., 2021; Shi et al., 2022). In their review of ML applications in the

geothermal industry, Okoroafor et al. (2022) only mentioned such studies, none on forecasting. That study mentioned induced seismicity forecasting via ML as an opportunity, as a future direction to investigate.

The simplest technique is the **linear regression** as described in Eq. (1) (Shapiro et al., 2007; Mignan et al., 2017). The most basic model structure can be described probabilistically as a Normal distribution,

$$\Pr(y|X,\theta) = \mathcal{N}(y|\mu(X),\sigma^2) \quad (2)$$

with the mean the linear model, for example $\mu = \lambda = c\dot{V}$. In the simplest case, there is only one variable $X = (\dot{V})$ with parameter set $\theta = (a_{fb}, b, \tau)$. This applies when time is not considered in the model, as in Figure 1. As shown in Feng et al. (2022), one could use $X = (\dot{V}, P)$ with P the wellhead pressure and $\mu = \lambda = c_0 + c_1\dot{V} + c_2P$. In the ML context, the parameter set can be simplified in terms of coefficients or weights instead of the correlated a_{fb} and b. When time is explicitly considered with $\lambda(t) = f(\dot{V}(t), \cdots)$ (i.e., Eq. 1), a Non-Homogeneous Poisson Process (NHPP) distribution is used instead of Eq. (2) as done in Broccardo et al. (2017) and Feng et al. (2022) for temporal forecasting. Most ML models however consider each time bin as an independent data sample. The likelihood distribution could be Poisson (instead of Normal in Eq. 2) since the target is a count, but this has yet to be tested (named **Poisson regression**).

Eq. (2) is equivalent to a likelihood distribution, which can be combined to a prior model in a **hierarchical Bayesian inference model** to estimate the model parameter set. A prior model is especially useful to constrain the model when little data is available, as it is the case at the early stage of underground stimulation. This has been done by Broccardo et al. (2017) and more recently Feng et al. (2022). The simplest Bayesian inference-based ML model is the **Naïve Bayes classifier**, which has yet to be tested for induced seismicity forecasting. It has however been previously tested for natural earthquake prediction as an alternative to other classifiers such as neural networks or SVM (Martínez-Álvarez et al., 2013). **Bayesian networks** were used by Hincks et al. (2018) in a similar way.

Linear regression can be extended to **non-linear regression** via a basis function expansion (e.g., polynomial). This has yet to be tested in the context of induced seismicity forecasting. Since the underlying non-linearities remain unknown, it is also reasonable to employ more complex models (e.g., decision tree ensemble regressors – see below) which can automatically capture non-linearities. Going back to Bayesian terminology, Eq. (2) is equivalent to a posterior distribution if the (implicit) prior model is the uniform distribution. Defining the prior as a Gaussian distribution or a Laplace distribution leads to the **Ridge or Lasso regularisation techniques**, respectively. These techniques can prevent overfitting which could occur if too many features are included in a non-linear regression. This has yet to be tested in the context of induced seismicity. They are however worth mentioning since they are general methods which also apply to other ML models to reduce the impact of data noise.

The **logistic regression** is the simplest classification model, which directly derives from the linear regression by changing the noise distribution from Normal to a Bernoulli distribution. Moreover, to have the output of the logistic regression equivalent to a probability, the linear regression is implemented as the variable of a sigmoid function. It results:

$$\Pr(y|X,\theta) = \mathrm{Ber}(y|\frac{1}{1+\exp(-y)}). \quad (3)$$

Like the (non-)linear regression, the method is transparent as each feature is weighted and the model produces well-calibrated probability estimates. Moreover, the classifier's decision surface is smooth (i.e., it reduces the impact of outliers). We can illustrate how a regression problem can be turned into a classification problem by considering the data shown in Figure 1 and then defining $y = 1$ if $\lambda > \lambda_{th}$ and $y = 0$ otherwise. The result is shown in Figure 6 with the matching logistic regression model fit. Note that the class prediction of a logistic regression is simply $\hat{y} = 1 \Leftrightarrow \Pr(y = 1|X) > 0.5$.

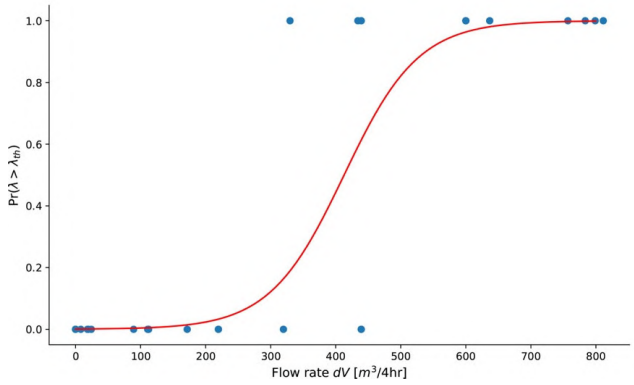

Fig. 6. Illustrating how a regression problem can be turned into a classification problem with $y = 1$ if $\lambda > \lambda_{th}$ and $y = 0$ otherwise, $\lambda_{th} = 25$ events/4hr, and the Basel data shown in Figure 1. The red curve shows the matching logistic regression fit.

Binary classification in the context of induced seismicity often consists in estimating where induced seismicity is likely to occur in a given region depending on some spatial variables. This has been done using logistic regression in several studies, mostly in fracking regimes (Pawley et al., 2018; Wozniakowska and Eaton, 2020; Hicks et al., 2021). In these studies, the number of features goes from a couple to two dozen (see, e.g., Sec. 3.2.1, as well as Tab. 2). For an A-TLS, a simple binary classification could be defined by a magnitude threshold (e.g., $y = 1$ if $m > 3$). However, it is likely that forecasting rate and maximum magnitude would remain preferable for seismic hazard assessment.

A generalisation of the linear regression, logistic regression and other models is called **Generalized Linear Model** (GLM). This also includes the Poisson regression, Lasso and Ridge models already mentioned.

The simplest **decision tree** consists of splitting the data conditioned on a threshold on feature x. Considering Figure 6, if we fix $x_{th} = 400$ m³/4hr, we get 24 samples in the first subset and 15 in the second. The first subset contains 23 samples with $y = 0$ and only 1 with $y = 1$, and the second, 3 with $y = 0$ and 12 with $y = 1$. The split is a good one if it separates the two classes reasonably well, which is evaluated via the concept of entropy. Here the entropy of the original dataset is $H_0 = -{}^{26}/_{39}\log_2({}^{26}/_{39}) - {}^{13}/_{39}\log_2({}^{13}/_{39}) \approx 0.92$, which is very high. The information gain of the split is $IG = H_0 - \sum_i w_i \langle \sum_j -p_{ij}\log_2 p_{ij} \rangle \approx 0.92 - \left[{}^{24}/_{39} \times \left(-{}^{23}/_{24}\log_2({}^{23}/_{24}) - {}^{1}/_{24}\log_2({}^{1}/_{24})\right) + {}^{15}/_{39} \times \left(-{}^{3}/_{15}\log_2({}^{3}/_{15}) - {}^{12}/_{15}\log_2({}^{12}/_{15})\right)\right] \approx 0.49$, which is indicative of a substantial reduction of entropy. The ML-type decision tree optimizes the split to maximise the information gain and reiterates splitting each subset until all subsets are pure (i.e., composed of only one class). The most common decision tree ensemble technique is the **Random Forest classifier and regressor**. A milestone application was done by Rouet-Leduc et al. (2017) to predict the time until the next rock failure (see Fig. 7). It has been used by Qin et al. (2022) to forecast the rate of induced seismicity in the wastewater disposal regime. For classification problems, decision tree ensembles are often compared to the **Support Vector Machine** (SVM) ML class, which derives from the linear regression via the concept of hyperplane. Both approaches are often tested in parallel.

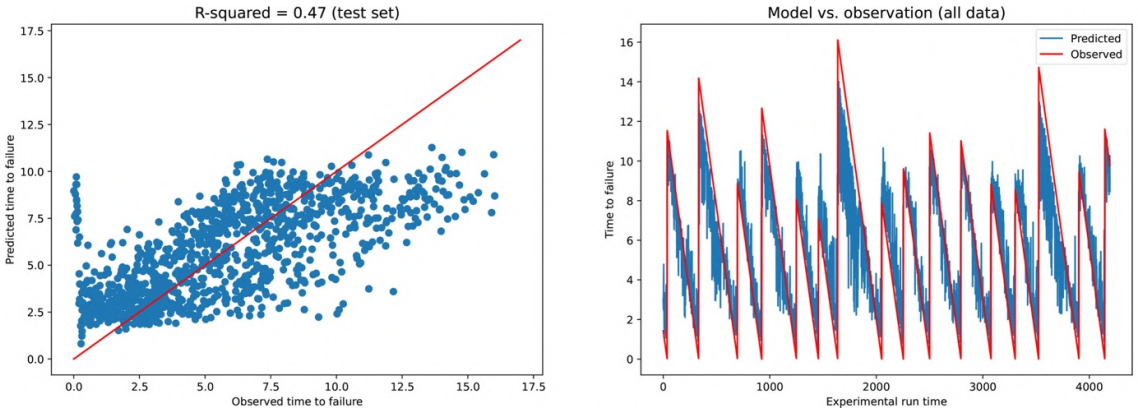

Fig. 7. Reproducing the Random Forest regressor of Rouet-Leduc et al. (2017) with some more heterogeneous data.

A direct extension of the logistic regression is a stack of logistic regressions with each sub-model one artificial neuron. The simplest architecture is the **feedforward neural network**, which is deep if it includes at least two hidden layers (it is called shallow otherwise). These models can be used for both classification (sigmoid output neuron(s)) and regression (linear output neuron). More sophisticated architectures include models of the **recurrent neural network** family (by adding a temporal recursion) and **convolutional networks** (by exploring the translation of patterns in space, or time). While their usefulness has been many times confirmed for seismic waveform analysis (e.g., Ross et al., 2018; Holtzmann et al., 2018; Zhang et al., 2020; Shaheen et al., 2021; Shi et al., 2022), this is not the case for time series forecasting. The review of many results (Mignan and Broccardo, 2020) indicates that the performance of most neural networks has yet to be compared to much simpler null hypotheses. No such model has so far been tested for IS temporal forecasting.

It is important to finally remark that decision trees and neural networks are incapable of extrapolating (they only interpolate). It means that they are ill-defined in a (pseudo-)prospective temporal forecasting context for IS (this will be demonstrated in a later technical report).

1.5 Conclusions

The present review showed that the number of ML studies for induced seismicity forecasting remains limited and that existing studies so far focused on its spatial distribution and not on its temporal evolution. The number of potential features is very large but again mostly of use in spatial distribution prediction. Operational variables, such as flow rate and pressure, in addition to statistical metrics applied to earthquake catalogues, appear as the main features to consider for IS temporal forecasting. A priori, there will be a great interest in testing the wide variety of ML models from the Generalized Linear Model (GLM) since this ML family builds on the well-known linear relationship at the basis of EM1 but with improved flexibility. In contrast to decision tree ensembles and artificial neural networks, GLM models can both interpolate and extrapolate, which is crucial for a TLS.

2. **Part 2:** Machine Learning model training and testing on observed data

In this section, various machine learning (ML) models are developed, trained, and tested on different injection experiment data. All the codes can be requested and will be released at the time of the publication. The goal is to explore the common statistical properties of induced seismicity despite the many variations observed in different experiments, to define and engineer features, and finally to identify the model that provides the best results on average, in the foreseen constraints of an ATLS. The conclusions of this report will be used as input for Parts 3-4.
The models, listed in Table 1, are ordered by increasing complexity. The tested datasets are listed in Table 2.

Table 1. List of ML models proposed

ID	Model	Variables	Likelihood	Prior(s)
ML0	Univariate linear regression	`dV`, `P` or `dV*P`	Gaussian	Uniform
ML0b	Univariate Poisson regression	`dV`, `P` or `dV*P`	Poisson	Uniform
ML1a	Multivariate linear regression	`dV`, `P`, `dV*P`, ...	Gaussian	Uniform
ML1b	Multivariate Ridge	`dV`, `P`, `dV*P`, ...	Gaussian	Gaussian
ML1c	Multivariate Lasso	`dV`, `P`, `dV*P`, ...	Gaussian	Laplace
ML1d	Multivariate Elastic Net	`dV`, `P`, `dV*P`, ...	Gaussian	weighted Gaussian + Laplace
ML2a	Random Forest	`dV`, `P`, `dV*P`, ...	NA	NA
ML2b	AdaBoost	`dV`, `P`, `dV*P`, ...	NA	NA

Table 2. List of injections considered, in alphabetical order

ID	Site	Variables	References
`Basel06_KD14`	Basel 2006	Earthquake catalogue (`t`, `m`, `x`, `y`)	`mmc1.txt` from Kraft & Deichmann (2014)
		Operational data (`dV`, `P`)	Digitized from Häring et al. (2008)
`Basel06_H19`	Basel 2006	Earthquake catalogue (`t`, `m`)	`supp1_detection_catalog_+ML.dat` from Herrmann et al. (2019)
		Operational data (`dV`, `P`)	Digitized from Häring et al. (2008)
`Bedretto20_ST2_` (`1a`, `4a-b`, `5`, `6`)	Bedretto 2020	Earthquake catalogue (`t`, `m`)	From inducat
		Operational data (`dV`, `P`)	From inducat
`FORGE22`	FORGE 2022	Earthquake catalogue (`t`, `m`)	From inducat
		Operational data (`dV`, `P`)	From inducat
`Soultz93`	Soultz 1993	Earthquake catalogue (`t`, `m`, `x`, `y`)	`SSFS2000-Catalogue_Cuenot.csv` from CDGP[1]
		Operational data (`dV`, `P`)	`Hydraulics_93.csv` from CDGP[1]
`Soultz00`	Soultz 2000	Earthquake catalogue (`t`, `m`)	`2000-06-30_hydraulics.csv` from CDGP[1]
		Operational data (`dV`, `P`)	`Hydraulics_93.csv` from CDGP[1]
`Soultz03`	Soultz 2003	Earthquake catalogue (`t`, `m`)	`SSFS2003-catalogue.csv` from CDGP[1]
		Operational data (`dV`, `P`)	`SSFS2003-Hydraulic_stimulation.csv` from CDGP[1]
`Soultz04`	Soultz 2004	Earthquake catalogue (`t`, `m`)	`SSFS2004-catalogue_downhole.csv` from CDGP[1]
		Operational data (`dV`, `P`)	`SSFS2004-Hydraulic_stimulation.csv` from CDGP[1]
`Soultz05`	Soultz 2005	Earthquake catalogue (`t`, `m`)	`SSFS2005-catalogue_downhole.csv` from CDGP[1]
		Operational data (`dV`, `P`)	`SSFS2005-hydraulics.csv` from CDGP[1]

[1] https://cdgp.u-strasbg.fr/geonetwork/srv/eng/catalog.search#/search [2] https://tcs.ah-epos.eu/#episodes

Only the injection phase is considered in this report for several reasons: Injection and post-injection are two different processes, which are commonly modelled separately (e.g., EM1 is a piecewise model). The two phases are function of different variables. Parameterisation of the post-injection rate decay is only fittable after shut-in. There is no possible control of the behaviour of the post-injection tail during injection in the context of a traffic light system solely based on statistics. For the purpose of developing ML models, it is important to first focus on the injection phase and avoid making the codes too complex.
Each data sample (X,y) corresponds to a subset Δt of the time series of observations [0,ts], the interval between an initial time 0 and the shut-in time ts. Features X can be derived, for each sample, from earthquake data (occurence time t, magnitude m) and/or operational data (flow rate dV, pressure P), while **the main target y is the seismicity rate λ in Δt** (see Section 2.4.1 for a maximum magnitude $Mmax$ target). Model performance is assessed by calculating the mean absolute error (MAE) on independent test sets. Model comparison is also done using MAE while probability gain is used in Part 3.

2.1 Data preparation

For each injection dataset, we import the raw data and define the array `data` with variables:

- `data[0,:]`: Earthquake's time `t` in decimal days from day of start of injection
- `data[1,:]`: Earthquake's magnitude `m`
- `data[2,:]`: Flow rate `dV` in m^3/day (m^3 to match a_{fb} unit, /day to match ATLS timeseries unit)
- `data[3,:]`: Cumulative volume `cumV` in m^3
- `data[4,:]`: Pressure `P` in bars

Flow rate, cumulative volume and pressure are interpolated at the earthquake occurrence times. Note that we make the distinction between two types of injections because of different interpolation structures and different ML training/testing strategies:

- `data1_files`: Long individual continuous injections (Basel and Soultz).
- `data2_files`: Short successive injections (Bedretto Underground Lab, FORGE)

Figure 1 (a, b) shows the raw data (data1_files, data2_files) per injection experiment.

We systematically predict the seismicity rate y in the time window Δt for the matching variable set $X=(x1,\cdots,xn)$ (retrospective forecasting). This target is the most important one as it is a primary metric, while the maximum magnitude is derivate (depending on the rate and b , with b independently fittable - see Section 2.4.1).
We define the data samples (X,y) in time bins with user-defined width Δt (i.e.,`dt`). There are then two options for the training set-test set split:

- `data1_files`: For long individual injections with many earthquakes generated, we perform a `ratio_train` /`1-ratio_train` split of the injection window.
- `data2_files`: For short successive injections, we select the first injection(s) for the training and the next injection(s) for the test set. This is due to the fact that there are otherwise not enough events to train and test an ML model.

Note that sample shuffling is common practice in ML to avoid the model to be trained and tested on subpopulations of potentially different statistics. However, we must also investigate the case where no shuffling is done since the ML models to be proposed are planned to be used for (pseudo-)prospective forecasting in an ATLS (see Section 2.3 and Part 3). As described in Section 2.2, considering a long forecast horizon (by increasing the test period by reducing `ratio_train`) will, sometimes significantly, reduce model performance. We compare models in a pseudoprospective framework (i.e., refitting the model as new data comes in over time to predict the next time bin Δt) in Section 2.3.

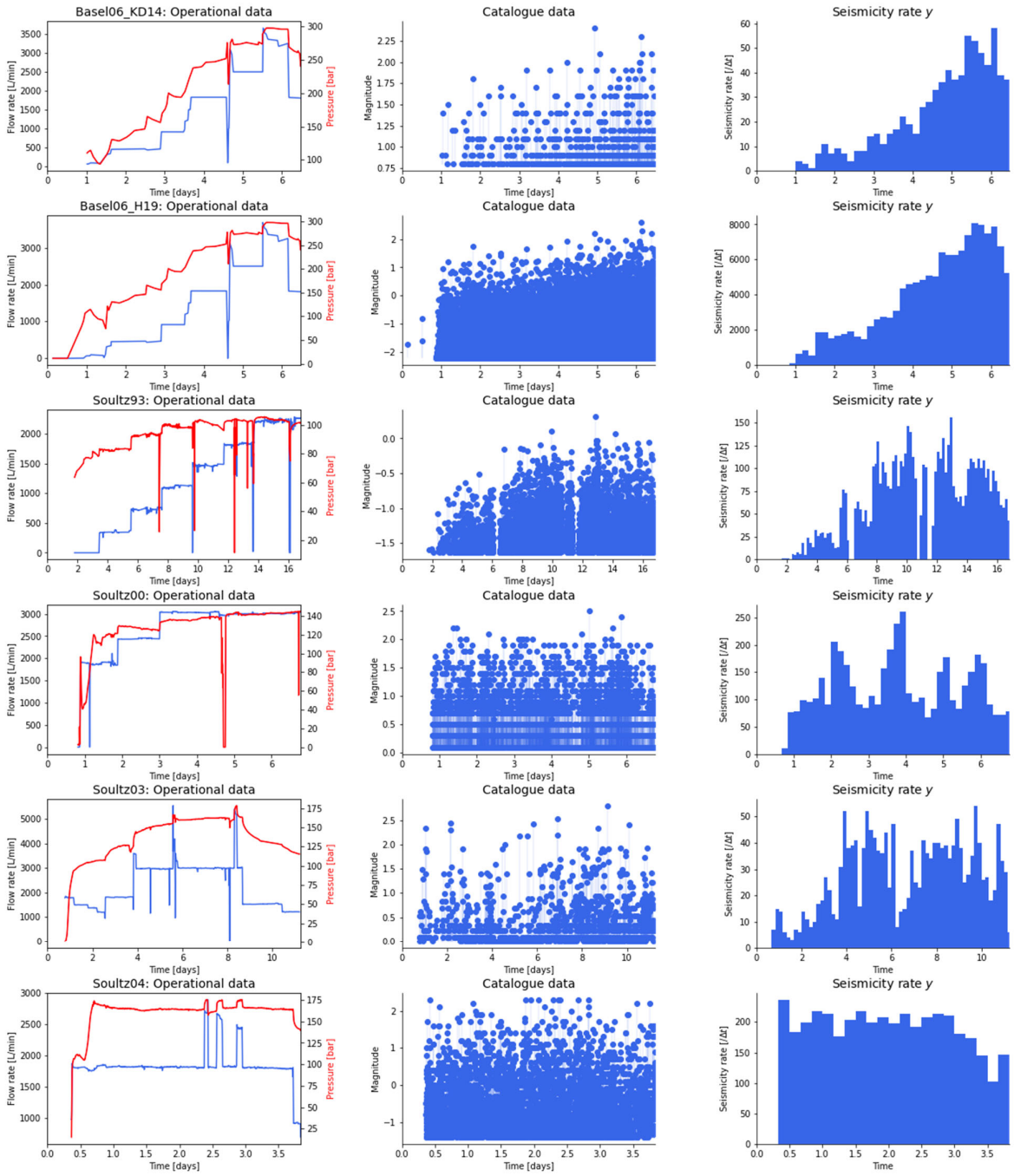

Figure 1a. Raw data (data1_files) per injection experiment (each row), with operational data (left) and catalogue data (center) from which the feature set X is defined, and the induced seismicity rate target λ=y (for the user-defined Δt).

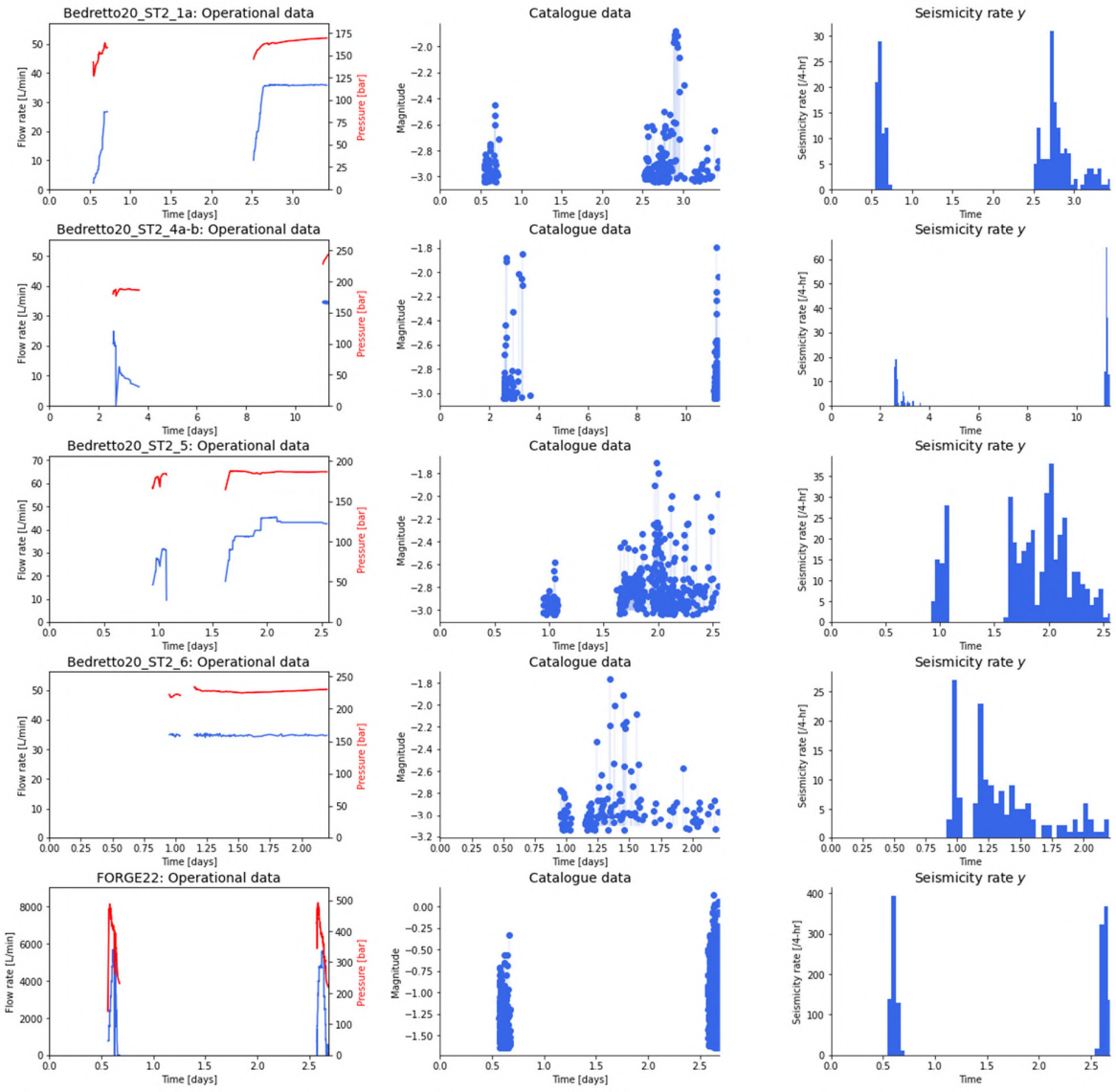

Figure 1b. Raw data (data2_files) per injection experiment (each row), with operational data (left) and catalogue data (center) from which the feature set X is defined, and the induced seismicity rate target λ=y (for the user-defined Δt).

2.2 ML model definition, training, testing, and ranking

We know from preliminary works that the rate of induced seismicity λ is proportional to the injected flow rate $\dot{V}$. The EM1 model proved that this simple relationship is sufficient to forecast induced seismicity reasonably well. As a consequence, **the first category of ML model to test is the Linear Model**, which includes linear regression, weighted linear regression, and generalised linear models (GLM) leading to a panoply of models with different probability distributions for the likelihood and prior functions (see Table 1 and Sections 2.2.1-2.2.3). There are also many features one can test in addition to the flow rate, such as the cumulative volume, the well-head pressure, as well as some seismicity metrics. New features can also be engineered, such as polynomial basis functions and feature crosses. Additional features can also correspond to the previously mentioned metrics but in previous time increments to describe some potential temporal dynamics.

We are also well aware that the model training is limited, for each experiment, two hundreds of data samples at best. As a consequence, **simple, high-bias models should be favored over more complex, high-variance models.** Moreover, decision trees and artificial neural networks cannot extrapolate. We will prove in a pseudo-prospective setting that, for example, **decision trees are ill-defined when ($\dot{V}$,λ) increase beyond the training data range** (Section 2.2.3). This means that constraints on ML strategies are very much limited in the context of an ATLS.

2.2.1 MLo - Univariate Linear Regression

We define MLo as the simplest possible ML model, i.e., a linear regression defined by:

$$y = w\dot{V}$$

with $x = \dot{V}$ the average flow rate. We also test other individual features including the pressure and the feature cross $\dot{V} \times P$ (by changing `indfeature: indfeature = 0` ($\dot{V}$), `1` (P) and `2` ($\dot{V} \times P$)). These are the main primary features which are likely to drive induced seismicity (any other feature is secondary and to be considered in the next sections on multivariate models). This first ML model is called MLo as it is simpler than EM1 for instance. We make the distinction between MLo, the standard least-squares regression, and MLob, the Poisson regressor.

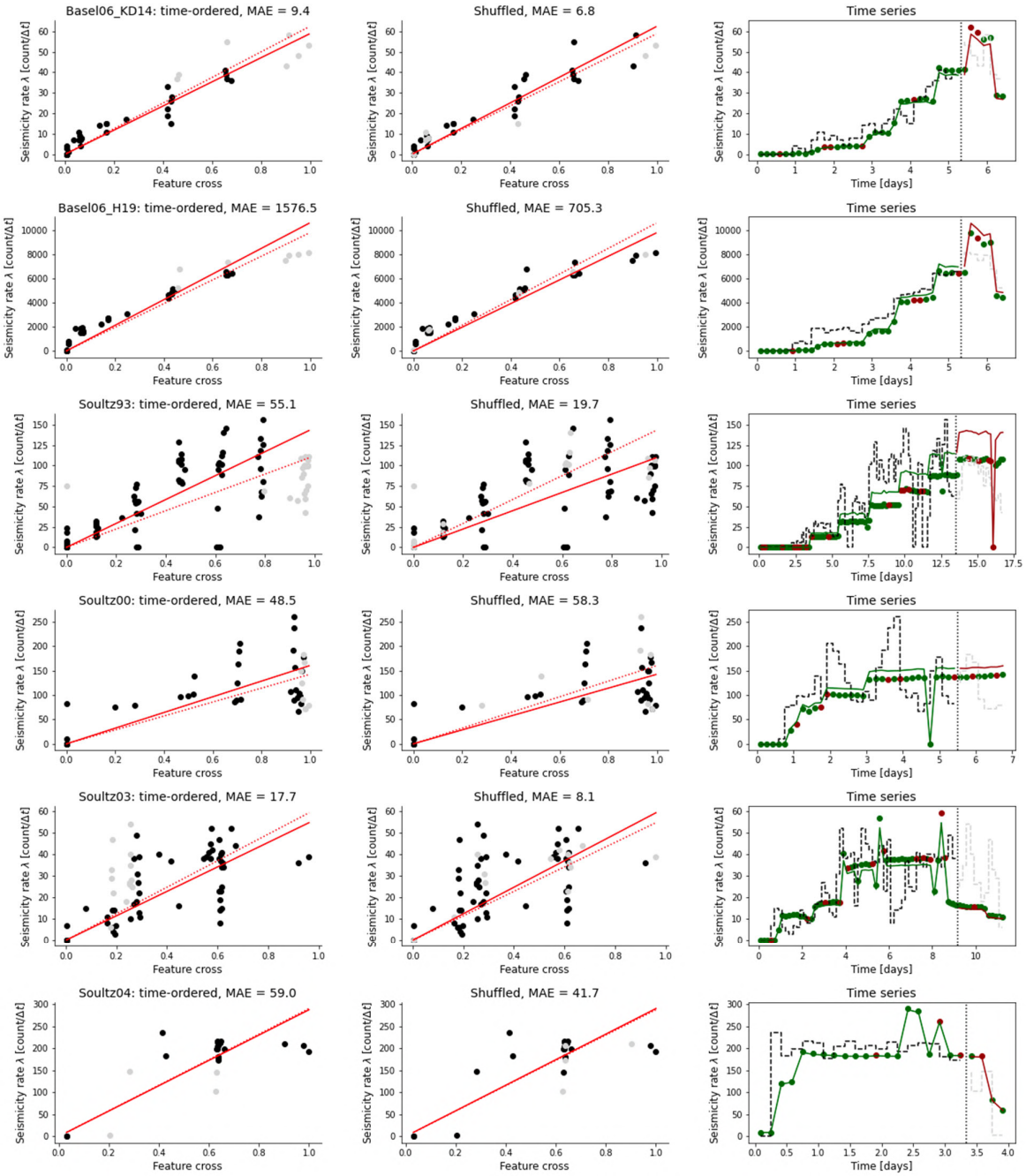

Figure 2a. Results (for data1_files) of MLo on original sample ordering (left), shuffled samples (middle) and for both cases displayed as a time series (right). For left two panels, the matching fit is represented as a solid red line and the fitted points as black dots. Fits (in green) and predictions (in red) are shown on the time series for both ordered data (curves) and shuffled data (points). Results are for `indfeature`= 2 ($\dot{V} \times P$).

** MAE per site per feature x for ordered samples **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Flow rate	8.5	943.0	51.4	43.7	16.2	57.9
Pressure	21.9	2195.3	23.4	44.2	9.3	91.1
Feature cross	9.4	1576.5	55.1	48.5	17.7	59.0

** MAE per site per feature x for shuffled samples **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Flow rate	4.0	708.9	18.9	43.9	9.1	54.1
Pressure	9.3	1240.7	35.9	21.9	8.4	52.9
Feature cross	6.8	705.3	19.7	58.3	8.1	41.7

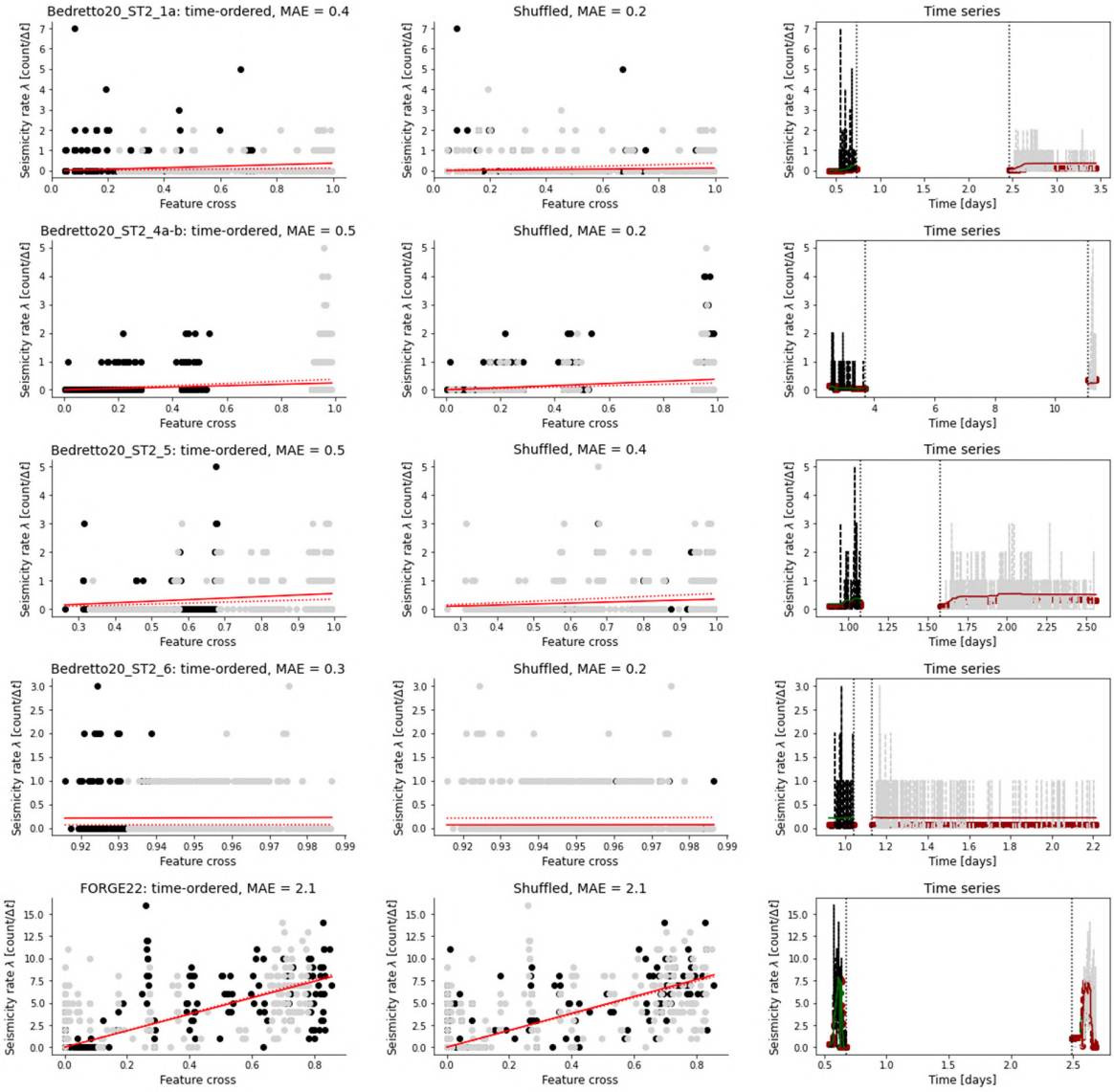

Figure 2b. Results (for data2_files) of ML0 on original sample ordering (left), shuffled samples (middle) and for both cases displayed as a time series (right). The matching fit is represented as a solid red line and the fitted points as black dots. Fits (in green) and predictions (in red) are shown on the time series for both ordered data (curves) and shuffled data (points). Results are for `indfeature`= 2 ($\dot{V}\times P$).

** MAE per site per feature x for ordered samples **

	Bedretto20_ST2_1a	Bedretto20_ST2_4a-b	Bedretto20_ST2_5	Bedretto20_ST2_6	FORGE22
Flow rate	0.4	0.5	0.5	0.3	1.9
Pressure	0.2	0.4	0.4	0.3	3.7
Feature cross	0.4	0.5	0.5	0.3	2.1

** MAE per site per feature x for shuffled samples **

	Bedretto20_ST2_1a	Bedretto20_ST2_4a-b	Bedretto20_ST2_5	Bedretto20_ST2_6	FORGE22
Flow rate	0.2	0.2	0.4	0.2	2.0
Pressure	0.2	0.2	0.4	0.2	2.7
Feature cross	0.2	0.2	0.4	0.2	2.1

We observe for most cases that the MAE is much lower when training is done on shuffled data than on time-ordered data, indicating that **model performance has a tendency to perform worse in an ATLS setting because of the potential change of regime over time** (e.g., flow rate progressively increasing over time).
This is accompanied by a **problem of heteroscedasticity** (i.e., variance function of the variable value), corresponding in most cases to an increase of the variance as the flow rate (or pressure) increases, and so as the injection progresses over time. Although it should not bias the linear regression weight estimation on shuffled data, it can lead to important errors for time-ordered data. Since the error can lead to both over- or under-predictions of the target, it indicates that **non-linear regression (e.g., polynomial regression) should be avoided** as it might extrapolate from noise while the linear relationship is physically well established. Weighted linear regression could reduce the impact of outliers but we didn't find it to systematically improve the results (see commented lines in the code cell). Since we are dealing with count data, a Poisson regression is the next natural model to test, as it can explain the heteroscedasticity (with the variance equal to the rate in theory).
This first model (ML0) showed various levels of performance depending on the considered injection experiment, with no feature (flow rate or pressure) being systematically better and with the tested feature cross performing worse than the individual physical variables in many cases. This suggests that **a multivariate model should be considered** to reflect variations observed cross-experiments.

2.2.2 ML0b - Univariate Poisson regression

Since the target y is a count, the loss is not normally distributed but Poisson distributed. It means that a Poisson regressor (an example of Generalised Linear Model, GLM) is more suitable with the loss increasing with X, as observed above. Moreover it will define y in the range $[0,\infty]$.

In most cases, results are not better or worse with Poisson regression when considering pressure as the main feature (results may be worse for other features). The testing of this model is therefore not further pursued. Figure 3 shows the results for `data1_files` experiments. Results for `data2_files` are not shown since they do not provide any additional insight.

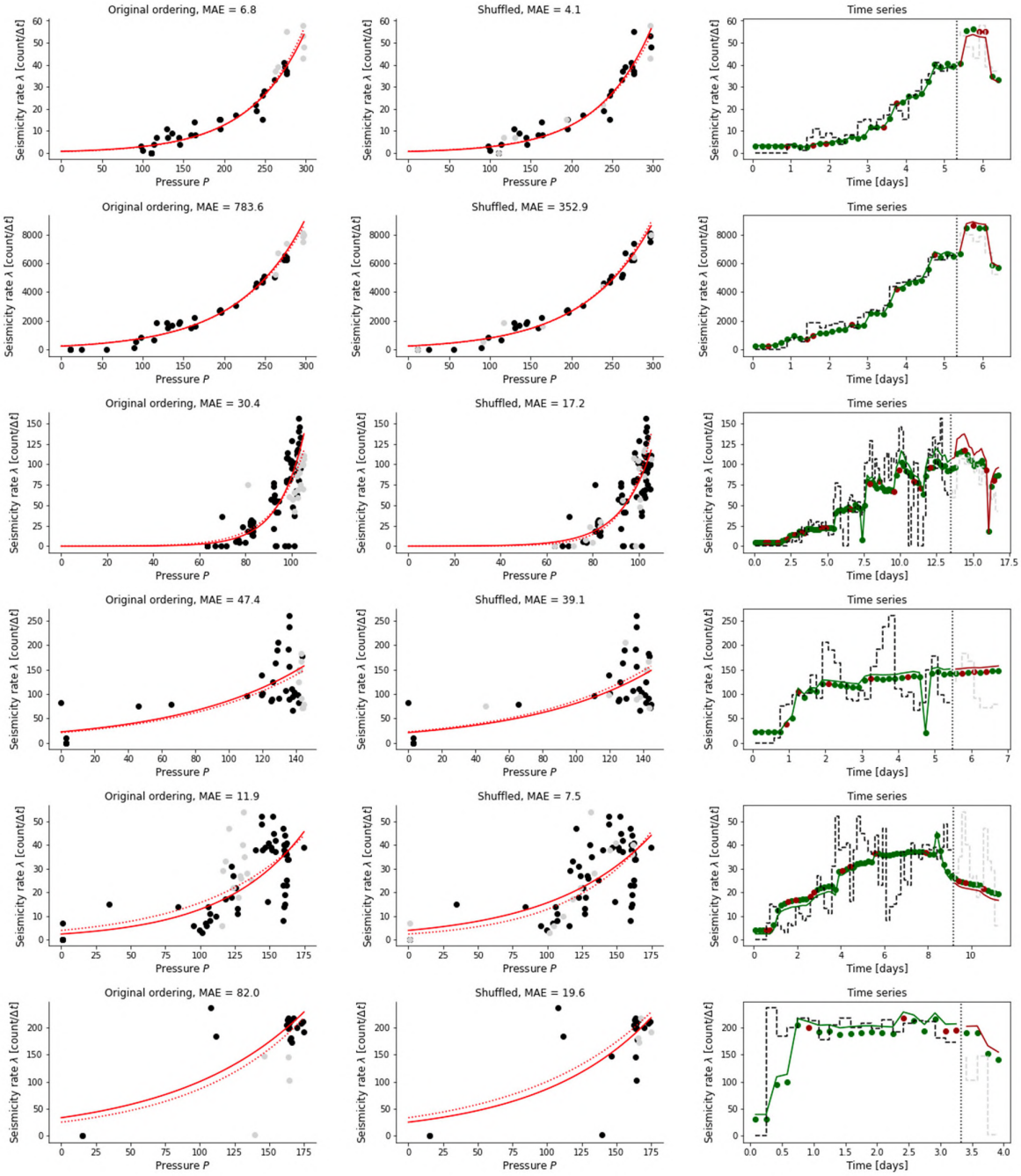

Figure 3. MLob results (for data1_files) for pressure as feature. The Poisson regressor is non-linear as we can observe on the first two columns. The model is better than MLo in the Basel case only. For other sites, and especially for other features, instabilities may be observed.

2.2.3 ML1 - Multivariate linear regression with regularisation (Ridge, Lasso, and Elastic Net)

We now consider regularised multivariate linear regression to investigate how features can be combined to improve forecasting while limiting overfitting by having too many features. We therefore compare multivariate linear regression with Ridge (L2 regularisation), Lasso (L1 regularisation), and Elastic Net (combined L1-L2 regularisation) for all features defined in X. An important part of this subsection consists in testing more engineered features. The main hyperparameter of regularization will be `alpha`, which role will be discussed in Section 2.3.

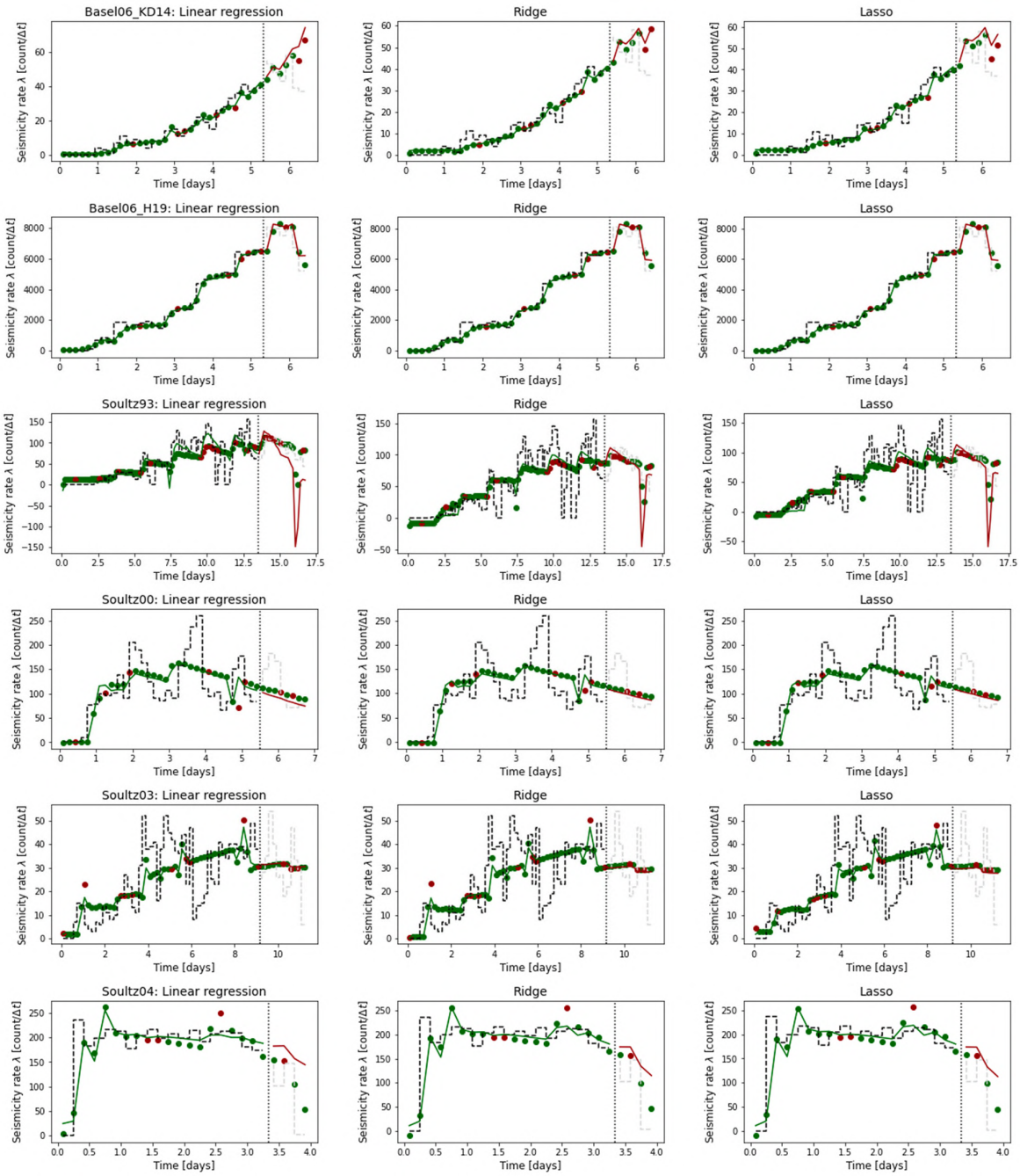

Figure 4a. Results (for data1_files) of three variants of ML1: Multivariate linear regression, Ridge and Lasso (Elastic Net not shown but similar).

** MAE per site for ordered samples when using ML1 **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Linear regression	13.0	517.4	45.3	33.2	9.4	67.6
Ridge	8.9	509.9	21.2	32.6	9.1	56.6
Lasso	9.2	509.8	22.6	32.6	9.1	56.1
ElasticNet	9.2	509.8	22.6	32.6	9.1	56.1

** MAE per site for shuffled samples when using ML1 **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Linear regression	9.5	236.2	22.2	38.5	6.6	31.3
Ridge	7.4	237.8	22.9	36.1	6.4	33.9
Lasso	6.0	237.9	22.5	35.2	6.0	33.8
ElasticNet	6.0	237.9	22.5	35.2	6.0	33.8

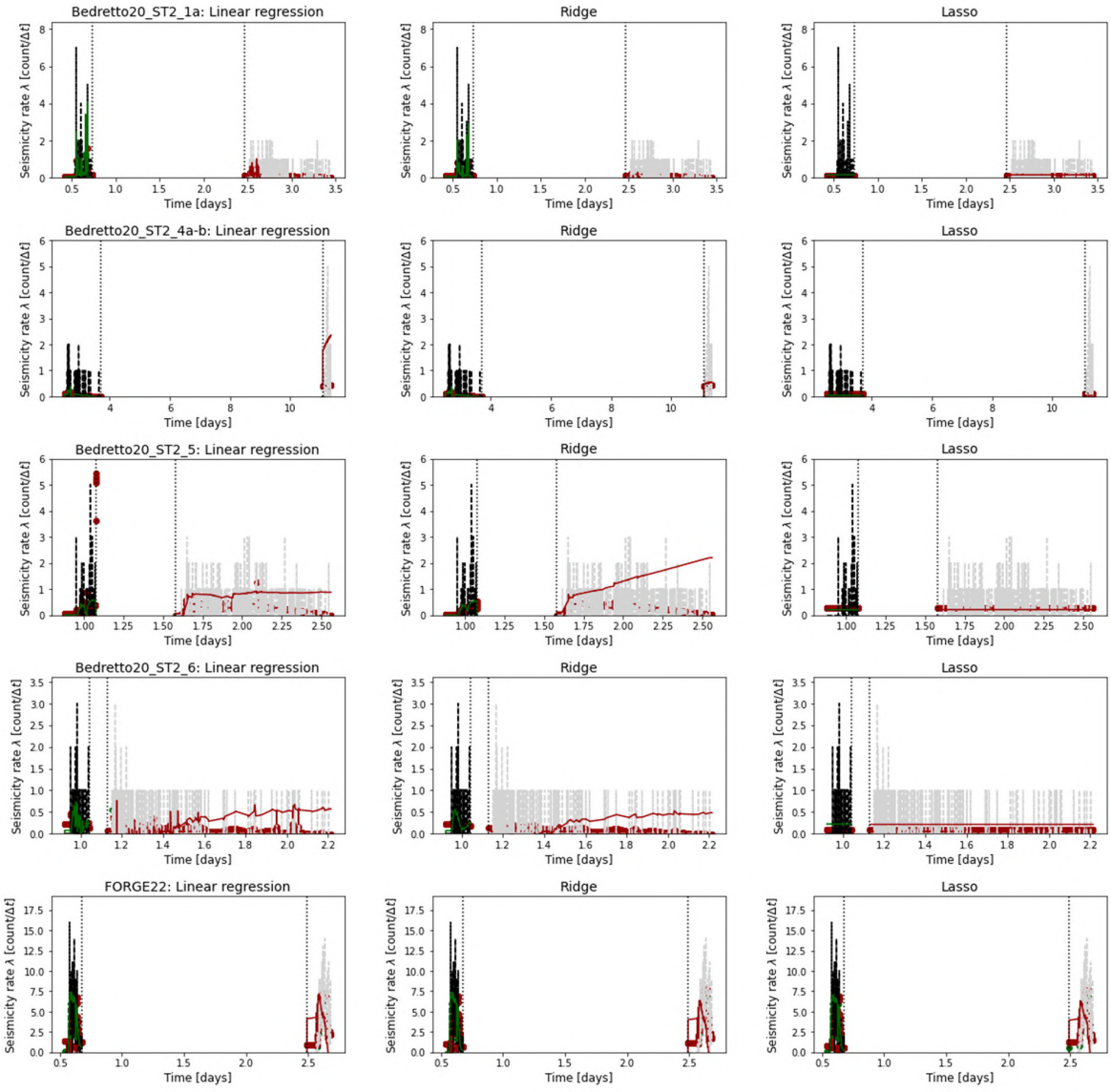

Figure 4b. Results (for data2_files) of three variants of ML1: Multivariate linear regression, Ridge and Lasso (Elastic Net not shown but similar).

Multivariate regression (with or without regularisation, ML1) is systematically better than univariate regression (ML0) for shuffled data. For ordered data and long time horizons, there is however no systematic improvement, with the best model (ML0 with given feature, or ML1) depending on the injection experiment. We observe however that for a short test set range (as in an ATLS), ML1 performs systematically better than ML0 (compared to any individual feature). Moreover, in most of these cases, linear regression and regularisation techniques give similar results. Some anomalies are observed for linear regression, suggesting that **Lasso is a reasonable choice** (since it is simpler than Elastic Net and provides 0 weights on minor features in contrast to Ridge).

Interestingly, the cumulative volume feature can be used as a correcting factor with negative weight, as can be seen for some of the Soultz datasets. This may seem like an ML trick although it might also relate to some real saturation effects.

This role is not systematic. Regarding the feature cross which was shown to perform worse than flow rate and pressure in ML0, this is confirmed in Lasso where the feature is discarded with a zero-weight.
We will add even more features in Section 2.3 when testing Lasso in a pseudo-prospective framework where the evolution of each parameter weight w_i will be investigated.

2.2.4 ML2 - Decision Tree Ensembles

We now test vanilla decision tree ensembles, the Random Forest and AdaBoost regressors, which are some of the most popular ML algorithms. As previously indicated, these standard models cannot extrapolate meaning that they cannot be applied in an ATLS for (X,y) ranges not previously observed. Yet, it remains interesting to observe their performance on shuffled data samples - we only use `data1_files` for illustration purposes. Neural networks will not be considered in this report as preliminary tests were inconclusive, likely due to the too high variance of the model in comparison to the limited data samples available for training (and we know that they also do not extrapolate).

** MAE per site for ordered samples using decision tree ensembles **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Random Forest	10.4	1197.5	18.2	45.8	10.1	68.6
AdaBoost	9.9	1222.5	12.8	46.1	11.6	83.6

** MAE per site for shuffled samples using decision tree ensembles **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
Random Forest	4.5	498.7	16.7	45.7	8.5	22.7
AdaBoost	5.6	509.6	18.2	56.1	8.8	15.9

We notice that the ML2 models perform similarly or slightly worse than linear models when the y_{test} range is within the y_{train} range. Figure 5 also illustrates the intrinsic limitation of ML2, which is the impossibility to extrapolate. This is well visible in the Basel case, leading to much worse performances. Since a usual injection profile will see the flow rate progressively increase over time, decision trees (and ANNs) must be discarded with no need to further explore changes in hyperparameterisation to optimize the results.

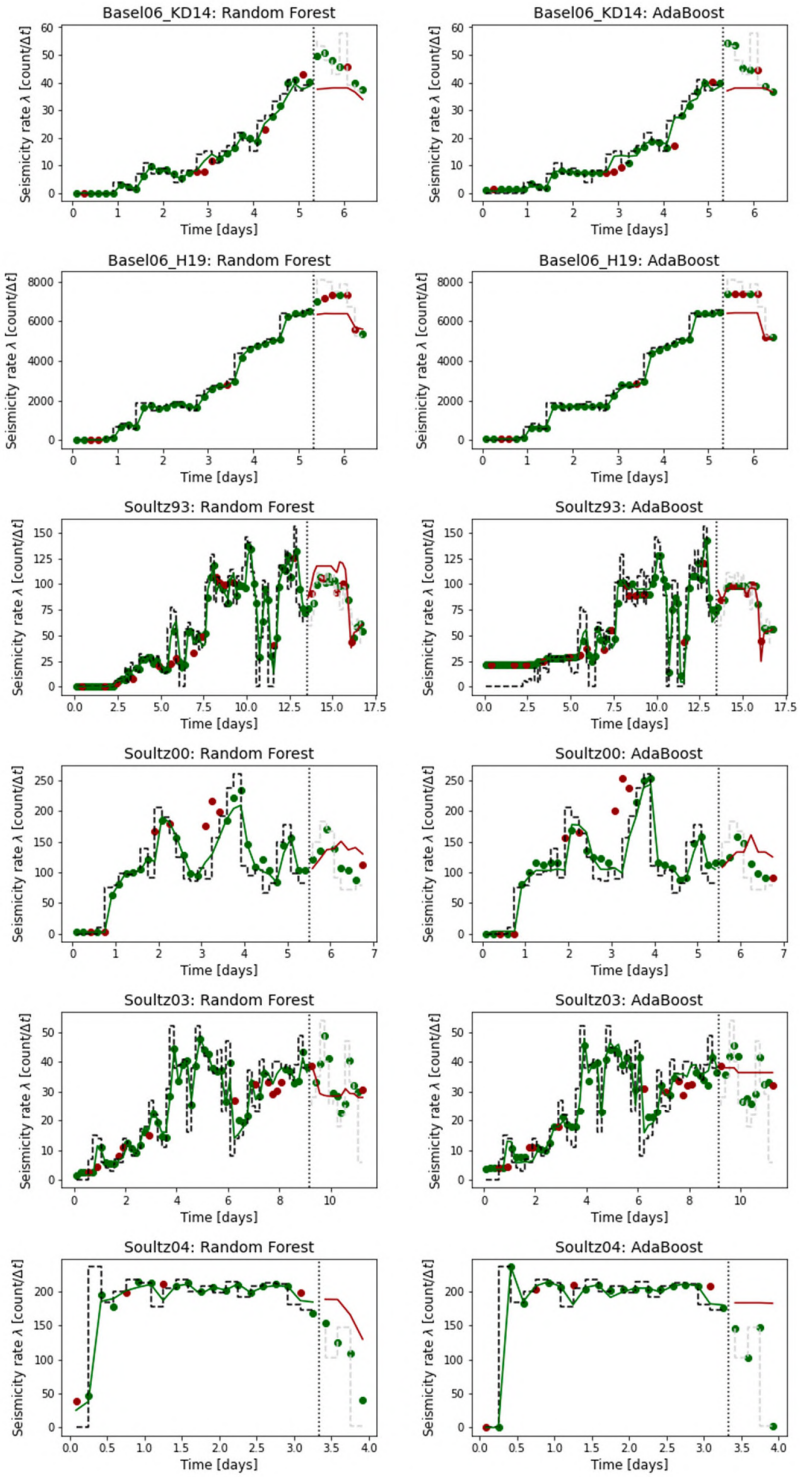

Figure 5. Results of decision tree ensembles (ML2): Random Forest (left) and AdaBoost (right). Notice the impossibility to extrapolate, which is well illustrated in the Basel case.

2.3 Model comparison in a pseudo-prospective framework

In this section (in preamble to Part 3), we compare the ML1 variants with the features previously defined as well as with new ones, such as catalogue based metrics (b-value, max m), other operational metrics (e.g., min/max flow rates and pressures), and also previous y observations. The main difference with the previous testing framework is that we only

have information about the future flow rate $\dot{V}(t_{target})$ and all other information for $t<t_{target}$. As such, here is the relation of each feature to time to avoid any data leakage:

- x_dV: Planned flow rate at in time bin t_{target}
- x_y_prev: Seismicity rate at previous bin
- x_dVmean_prev: Mean flow rate at previous bin
- x_dVmin_prev: Minimum flow rate at previous bin
- x_dVmax_prev: Maximum flow rate at previous bin
- x_Pmean_prev: Mean pressure at previous bin
- x_Pmin_prev: Minimum pressure at previous bin
- x_Pmax_prev: Maximum pressure at previous bin
- x_cumV: Cumulative volume up to previous bin
- x_bval: b-value for entire time series before the target bin
- x_mmax: max m_{max} for entire time series before the target bin

We continue using MAE as performance metric. Probability gain will be used in Part 4.

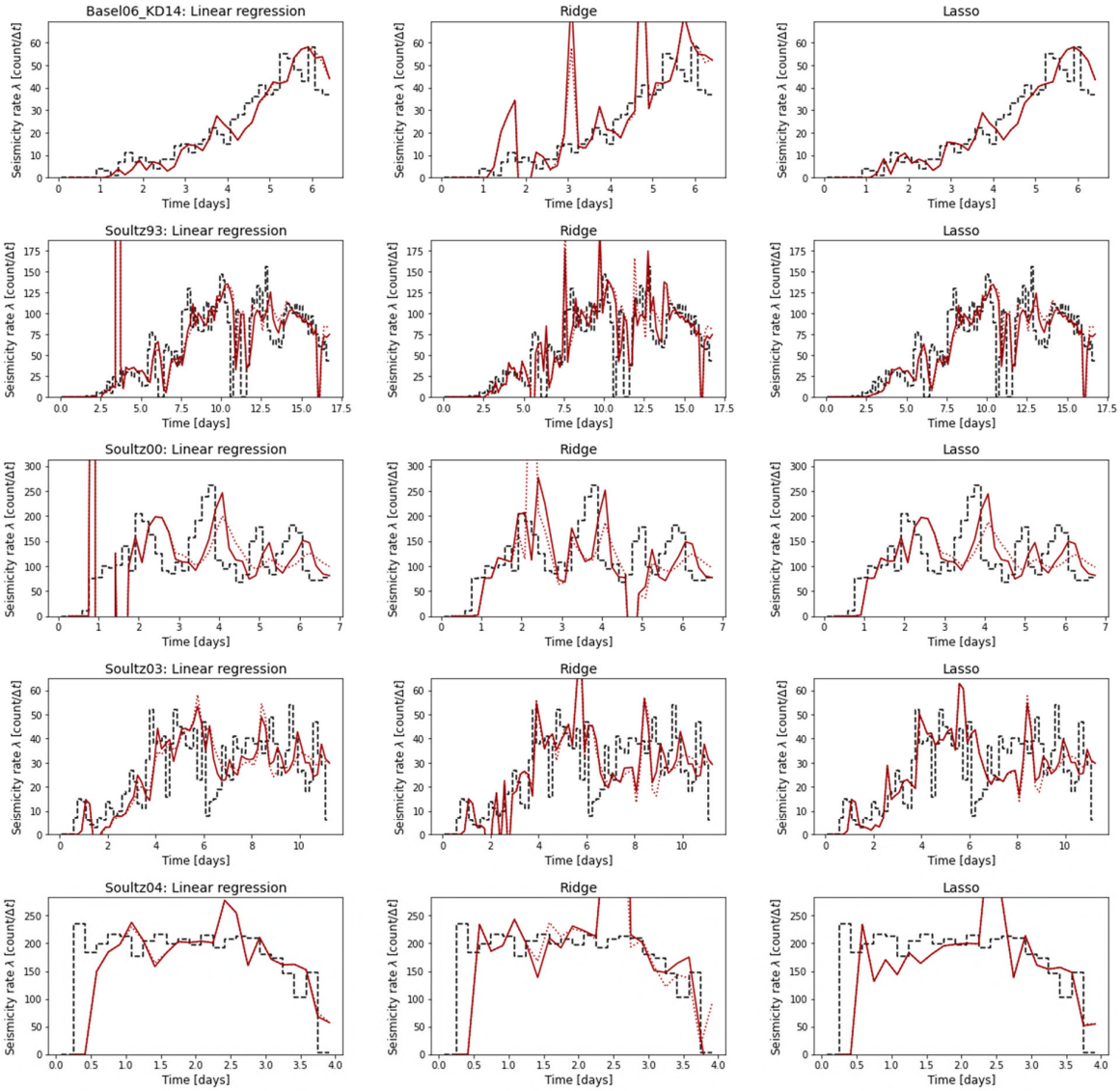

Figure 6a. Results (for data1_files) of three variants of ML1 in a pseudo-prospective setting. Red curves represent the forecast at time t (within Δt bin) with model fit done on $[0, t-\Delta t/2]$. With $y_{t-\Delta t}$ feature (solid curve) and without (dotted curve).

** MAE Results **

	Lin. reg.	Ridge	Lasso	Lin. reg. w. prev. y	Ridge w. prev. y	Lasso w. prev. y
Basel06_KD14	4.2	10.9	4.3	4.4	11.8	4.3
Soultz93	23385.7	23.2	18.0	23385.1	22.9	17.5
Soultz00	20095.8	62.8	43.4	20089.9	50.6	38.1
Soultz03	9.5	10.8	9.5	8.7	10.2	9.0
Soultz04	37.1	65.1	52.0	38.0	104.3	52.1

We observe that the Lasso model is the best forecaster in the pseudo-prospective framework (and will be compared to EM1 in Part 3). Performance is also only improved systematically for Lasso when $y_{t-\Delta t}$ is considered as an additional feature in X. As we can see from Figure 6a, this trick plays a role when anomalous fluctuations occur with the model trying to correct for it, although at the cost of a slight delay.

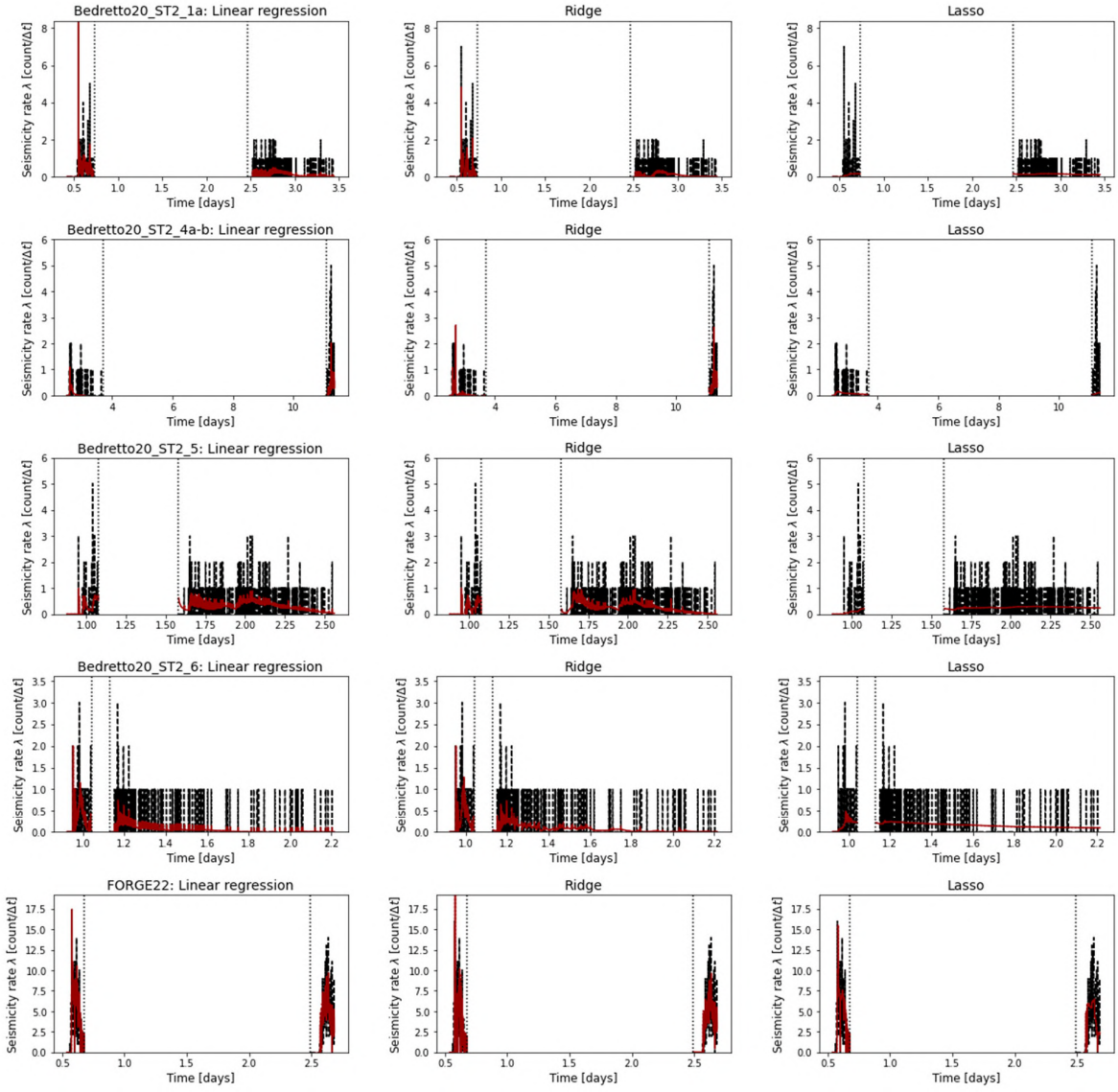

Figure 6b. Results (for data2_files) of three variants of ML1 in a pseudo-prospective setting. Red curves represent the forecast at time t (within Δt bin) with model fit done on $[0, t-\Delta t/2]$. With $y_{t-\Delta t}$ feature (solid curve) and without (dotted curve).

** MAE Results **

	Lin. reg.	Ridge	Lasso	Lin. reg. w. prev. y	Ridge w. prev. y	Lasso w. prev. y
Bedretto20_ST2_1a	0.2	0.2	0.2	0.2	0.2	0.2
Bedretto20_ST2_4a-b	0.2	0.2	0.2	0.2	0.2	0.2
Bedretto20_ST2_5	0.4	0.3	0.4	0.4	0.3	0.4
Bedretto20_ST2_6	0.2	0.1	0.2	0.2	0.1	0.2
FORGE22	1.6	2.0	1.6	1.4	2.0	1.6

Note how the FORGE22 dataset is rich in seismicity in a very short time window. Hence, we used a time bin significantly shorter than for Bedretto with a forecast per minute instead of per hour. Figure 7 shows the Lasso forecast on FORGE22, zooming into the 2 injection periods:

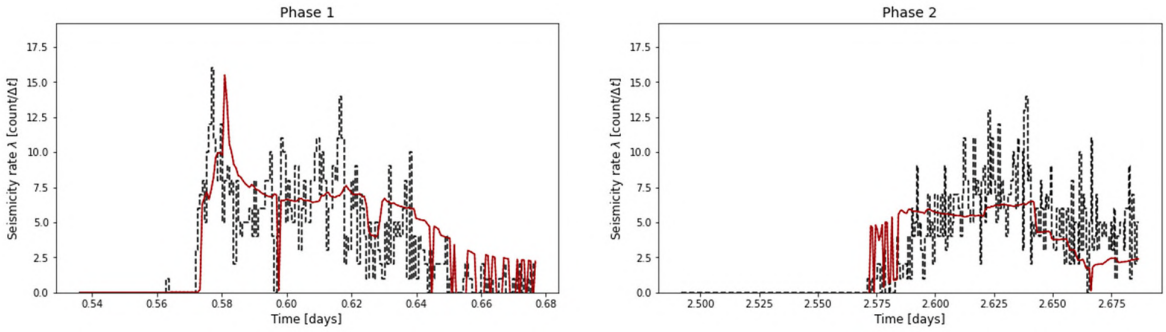

Figure 7. Lasso forecast on FORGE22, zooming into the 2 injection periods.

Figure 8 shows the evolution of the relative weight of each Lasso feature. It shows that several features play a role in forecasting the seismicity rate in Lasso and that their relative weights change over time with many differences observed between different injection experiments. **This proves that ML is relevant for an ATLS**, learning from a wide variety of features.

Despite the high variability observed in respective weights, some general patterns can be observed:

- The planned flow rate is the main feature as it is the only information into the future (basis of EM1).
- Previous flow rates and pressures play a role intermittently.
- Cumulative volume (positive and negative variants) seem to be used to calibrate the model and deviate from the linear relationship between flow rate and seismicity rate.
- The previously observed seismicity rate is used as feature when anomalies are observed with the seismicity rate deviates from what is expected from the injection profile.
- No role of catalogue-based features.

Below we report a table providing the list of non-zero weight features (x) per experiment. Features which are given a 0-weight by Lasso are: the feature cross $\dot{V}\times P$ (already removed), mean pressure (for high alpha, e.g., 100 instead of 10), and both catalogue-based features, i.e. the b-value and m_{max}.

	dVplanned	yprev	dVmean	dVmin	dVmax	Pmean	Pmin	Pmax	cumV	-cumV	mmax	bval
Basel06_KD14	x				x		x	x	x	x		
Soultz93	x	x	x	x	x		x	x	x	x		
Soultz00	x	x	x	x	x		x		x	x		
Soultz03	x	x	x	x	x	x	x	x	x	x		
Soultz04	x	x	x		x	x	x			x		

	dVplanned	yprev	dVmean	dVmin	dVmax	Pmean	Pmin	Pmax	cumV	-cumV	mmax	bval
Bedretto20_ST2_1a												
Bedretto20_ST2_4a-b												
Bedretto20_ST2_5								x				
Bedretto20_ST2_6						x		x				
FORGE22		x		x	x	x	x	x	x	x		

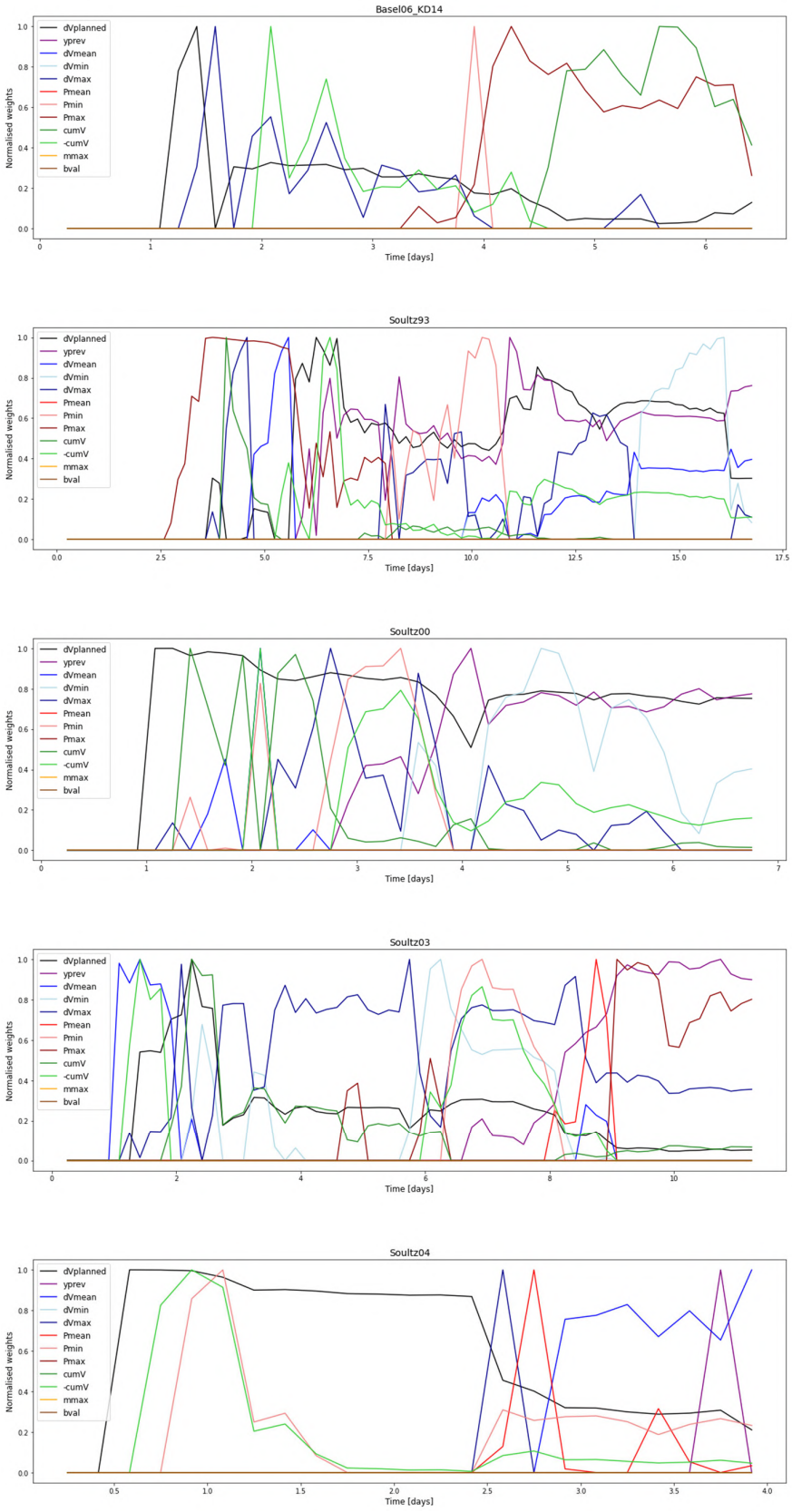

Figure 8a. Evolution of the relative role of each feature over time (for data1_files).

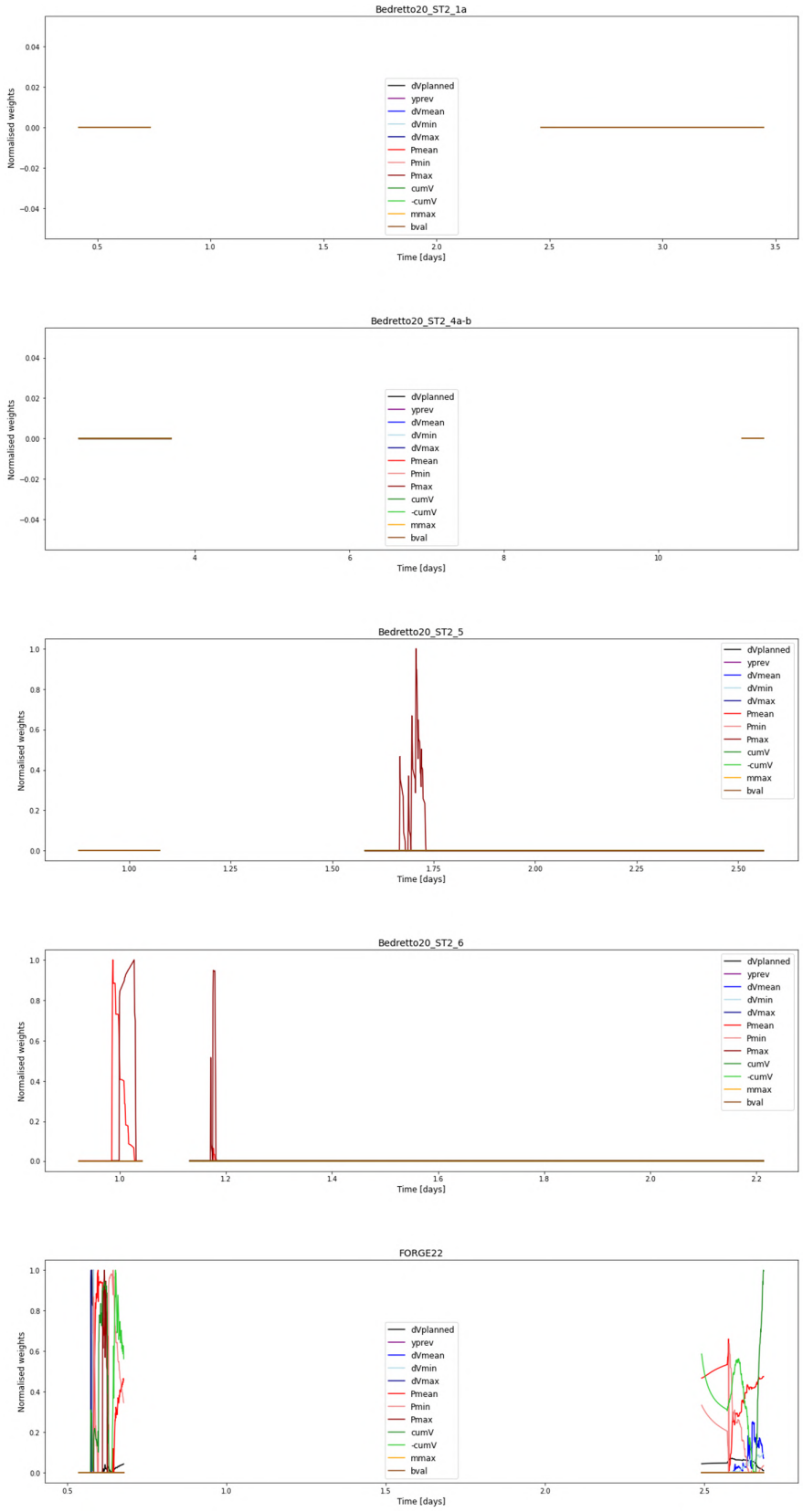

Figure 8b. Evolution of the relative role of each feature over time (for data2_files).

We here observe that Lasso is the best model and that seismicity-based features are given a zero-weight. Interestingly, increasing alpha to 100 (instead of 10) makes Lasso too conservative for short and small datasets (0-weight for all features, with final result simply the intercept, i.e., the mean of previous observations). Therefore, **we choose to fix alpha to 10 for both types of data** (`data1_files` + `data2_files`).

2.4 Add-ons to ML1

2.4.1 Maximum magnitude forecasting

$Mmax$ forecasting is derivative as it depends on the rate forecast and the b-value. Now that a well-performing ML model has been developed to compete against EM1, we can define the Lasso prediction y as input for $Mmax$ assessment in the next time bin. We rerun the previous pseudo-prospective test now adding $Mmax$ calculation based on the definition given by Broccardo et al. (2017).

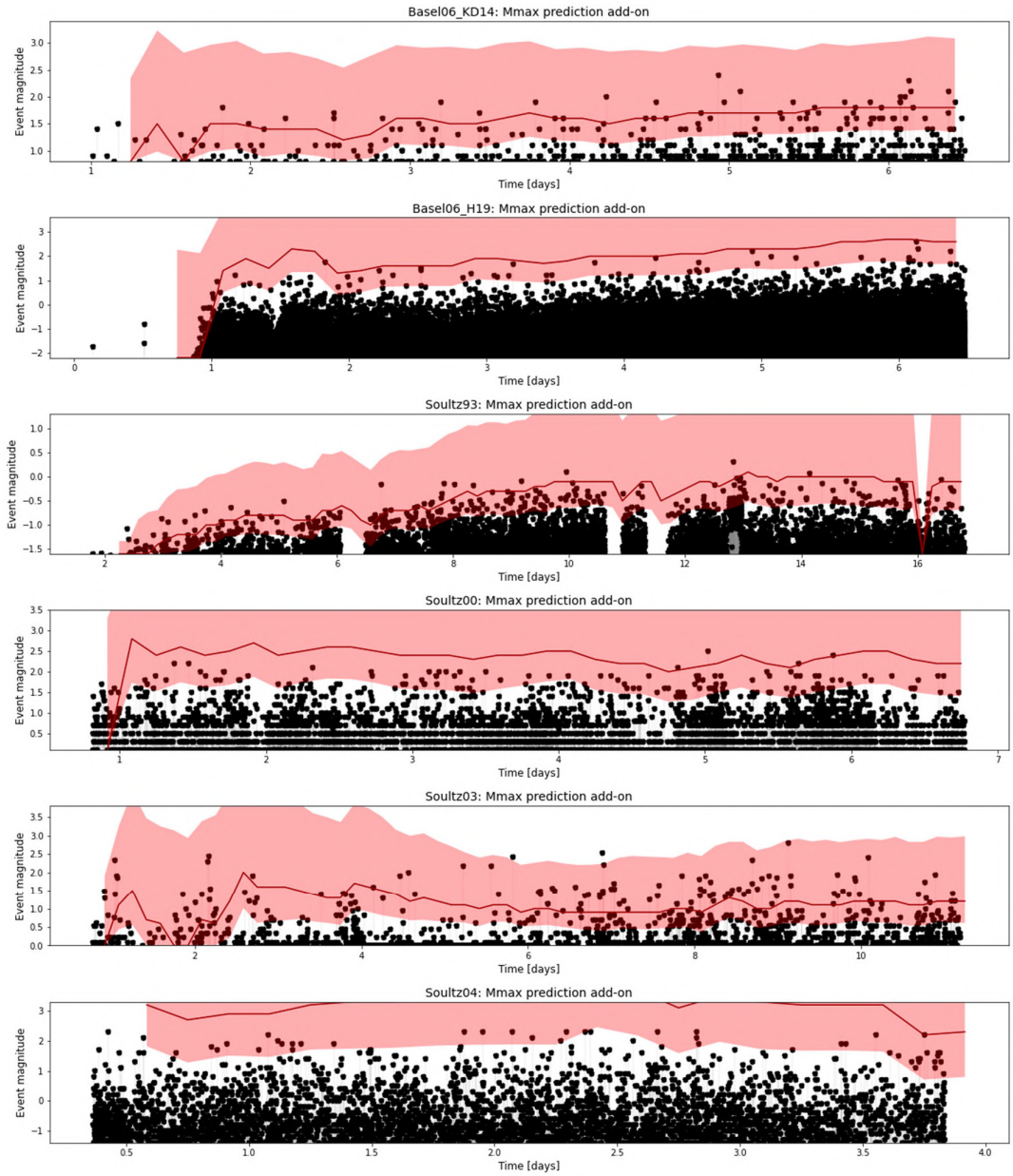

Figure 9a. $Mmax$ forecast (for data1_files) using ML1_Lasso in a pseudo-prospective setting. Red curves represent the forecast at time t (within Δt bin) with model fit done on $[0, t-\Delta t/2]$. The solid dark red curve represents the forecasted $Mmax$ mode and the lighter ribbon the 99% confidence interval.

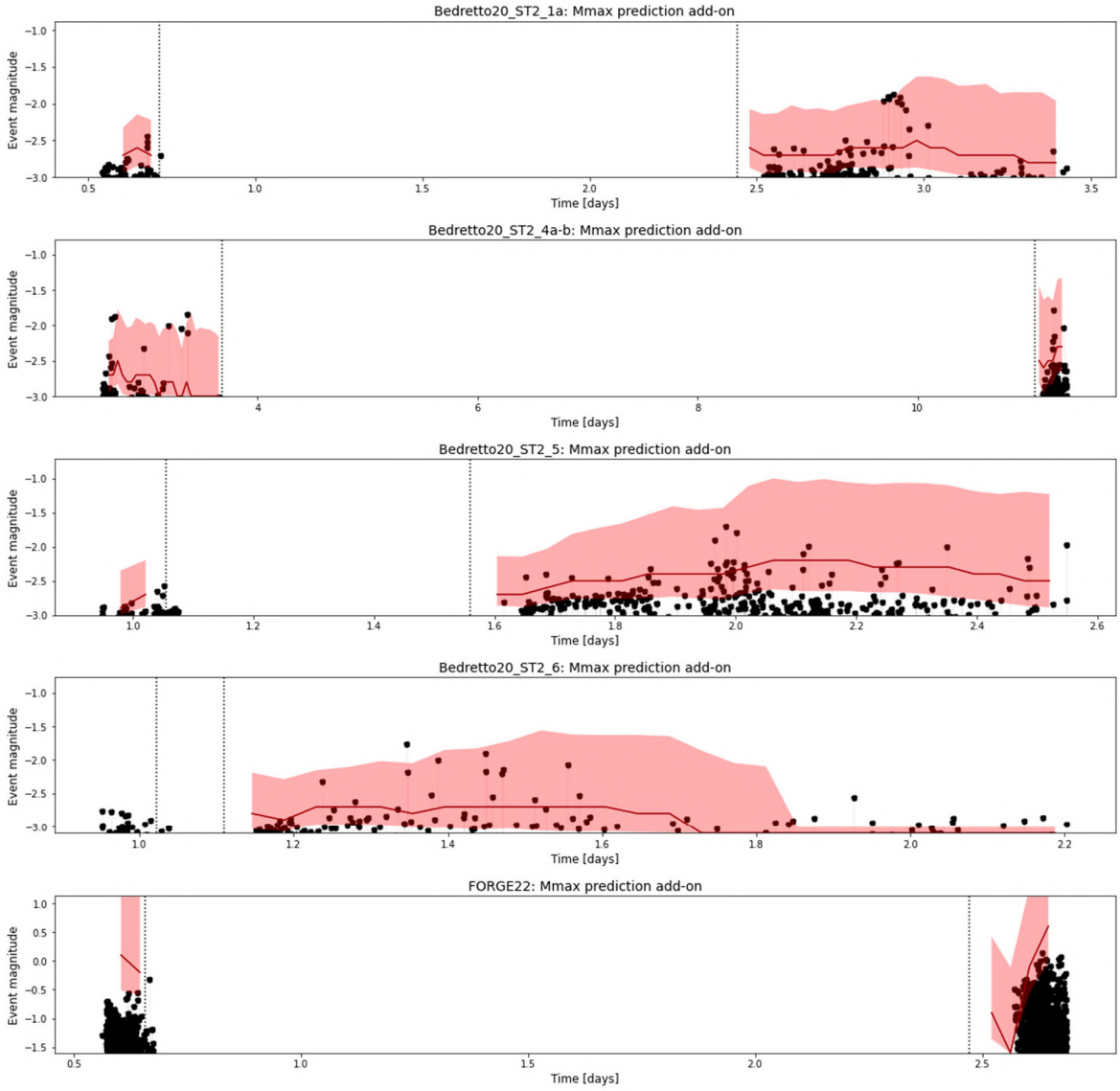

Figure 9b. $Mmax$ forecast (for data2_files) using ML1_Lasso in a pseudo-prospective setting. Red curves represent the forecast at time t (within Δt bin) with model fit done on [0,$t-\Delta t/2$]. The solid dark green curve represents the forecasted $Mmax$ mode and the lighter ribbon the 99% confidence interval.

We here verified that we can integrate $Mmax$ prediction in ML1_Lasso by simply using the statistical definition of $Mmax$ (Broccardo et al., 2017). Inputs here become the ML1_Lasso rate forecast y_{target} and the b-value previously observed.

2.4.2 Spatial considerations

Weh ave so far discarded the spatial component in seismicity rate forecasting. This aspect is of secondary importance in hazard assessment in view of the limited spatial extent of seismicity clouds. It remains however interesting to understand the evolution of the cloud. Due to the amount of data needed to observe any reliable spatial pattern, only `data1_files` will be considered. Of those, we test `'Basel06_KD14'` and `'Soultz93'` for which EW and NS distances from the well are directly given. We will first look at snapshots of the seismicity clouds at different times and characterise them by defining a set of features that describes a simple bivariate Gaussian distribution:

- Bivariate Gaussian distribution means (μx, μy) (i.e., centroid coordinates)
- Bivariate Gaussian distribution co-variance ((σ_{xx}^2, σ_{xy}^2), (σ_{yx}^2, σ_{yy}^2))

The method could be applied in three dimensions at the cost of lower reliability.

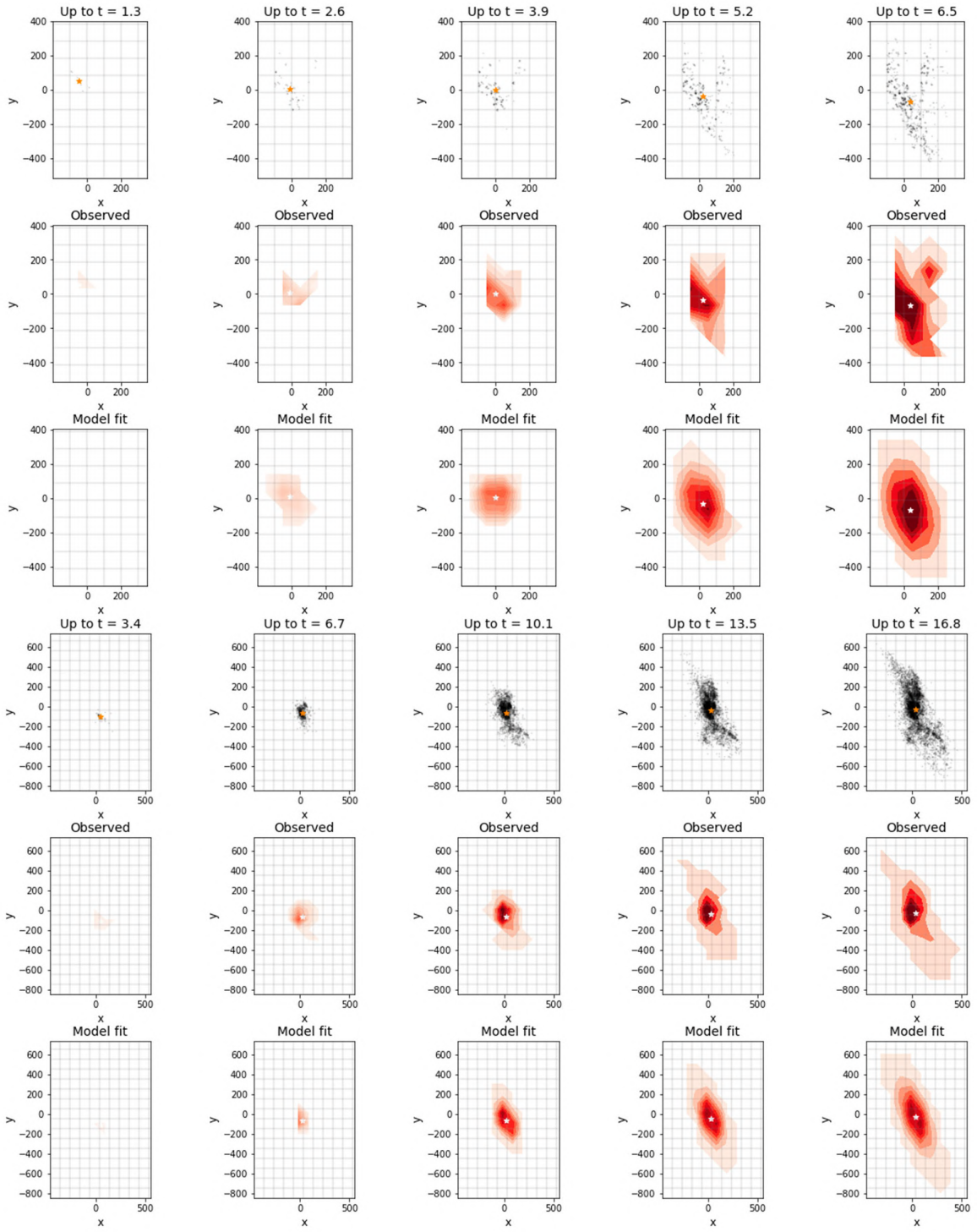

Figure 10. Snapshots of the seismicity clouds at different times in days. Top three rows are for 'Basel06_KD14', whereas the bottom three rows show the results for 'Soultz93'. Color represents the density of the seismicity.

A bivariate Gaussian model seems reasonable since it can model the anisotropy of the cloud. Considering more than one cloud is outside the scope of this study (yet, it could simply be treated by increasing the number of components in GaussianMixture, here fixed to 1). Let us now investigate how those different spatial features evolve through time.

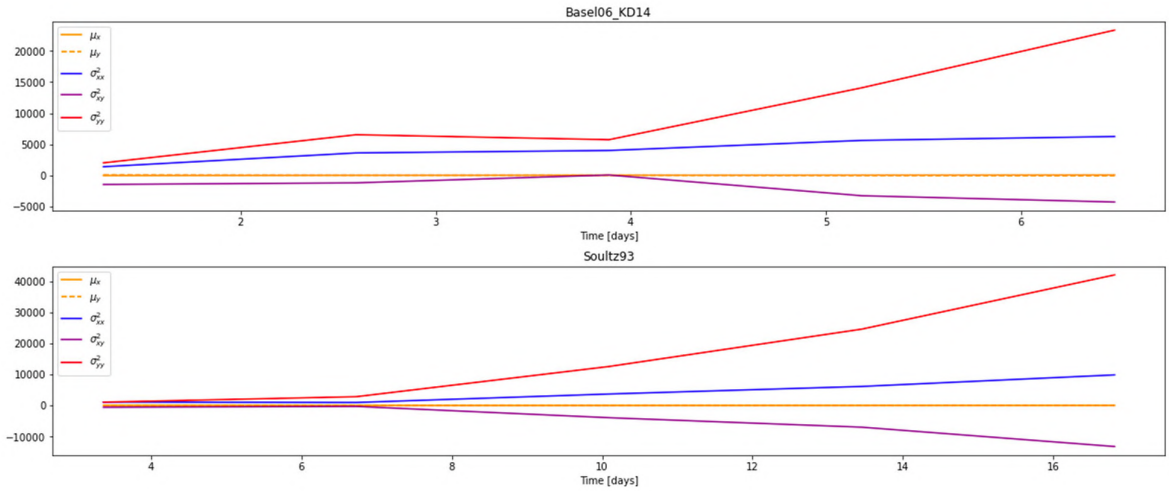

Figure 11. Evolution of spatial features with time: μ is the centroid of the cloud, the sigmas are the standard deviations of the spatial extent.

From Figure 11 we can notice the general N-S expension represented by σ_{yy} but also its acceleration over time. As such, we will keep linear regression for the centroid forecast but 2nd-order polynomial regression for the other cloud characteristics.

We will now rerun the ML1_Lasso pseudo-prospective forecast for the two sites, estimate the cloud statistics on the training set, and predict the future cloud spatial distribution at t_{target} (within Δt bin) from (non-)linear regression extrapolation and event density constrained by the predicated rate y_{target} (Figure 12).

A visual inspection suggests that a spatial add-on to ML1 is feasible. No statistical test is proposed at this stage of the project for performance assessment against other potential models.

2.5 Conclusions

The present analysis confirmed the importance of using Machine Learning for the forecasting of induced seismicity. The flexibility of ML indeed permits to weigh a large number of features function of variations over time and across experiments, which can show great deviations from the simple linear relationship between flow rate and induced seismicity rate. Since the linear relationship remains the main pattern, models from the Generalized Linear Model (GLM) family were naturally the ones to test. They are particularly adapted to the ATLS situation since, in contrast to decision tree ensembles or neural nets, they can: (i) extrapolate, (ii) keep the linear regression high bias to avoid overfitting when data remain scarce, and (iii) deal with a dozen of features or more while regularizing to again avoid overfitting. Over the wide range of experiments considered, **it appears that Lasso is the best model,** and is thus the model selected for comparison with EM1 in Part 3 (also in Part 4 for testing on physics-based simulations). Changing the regularization hyperparameter α plays a role for small datasets with α=100 constraining Lasso too much in this case. Results suggest that α=10 is reasonable for any type of data. We also showed that $Mmax$ and seismicity cloud forecasting can be simple adds-on to ML1.

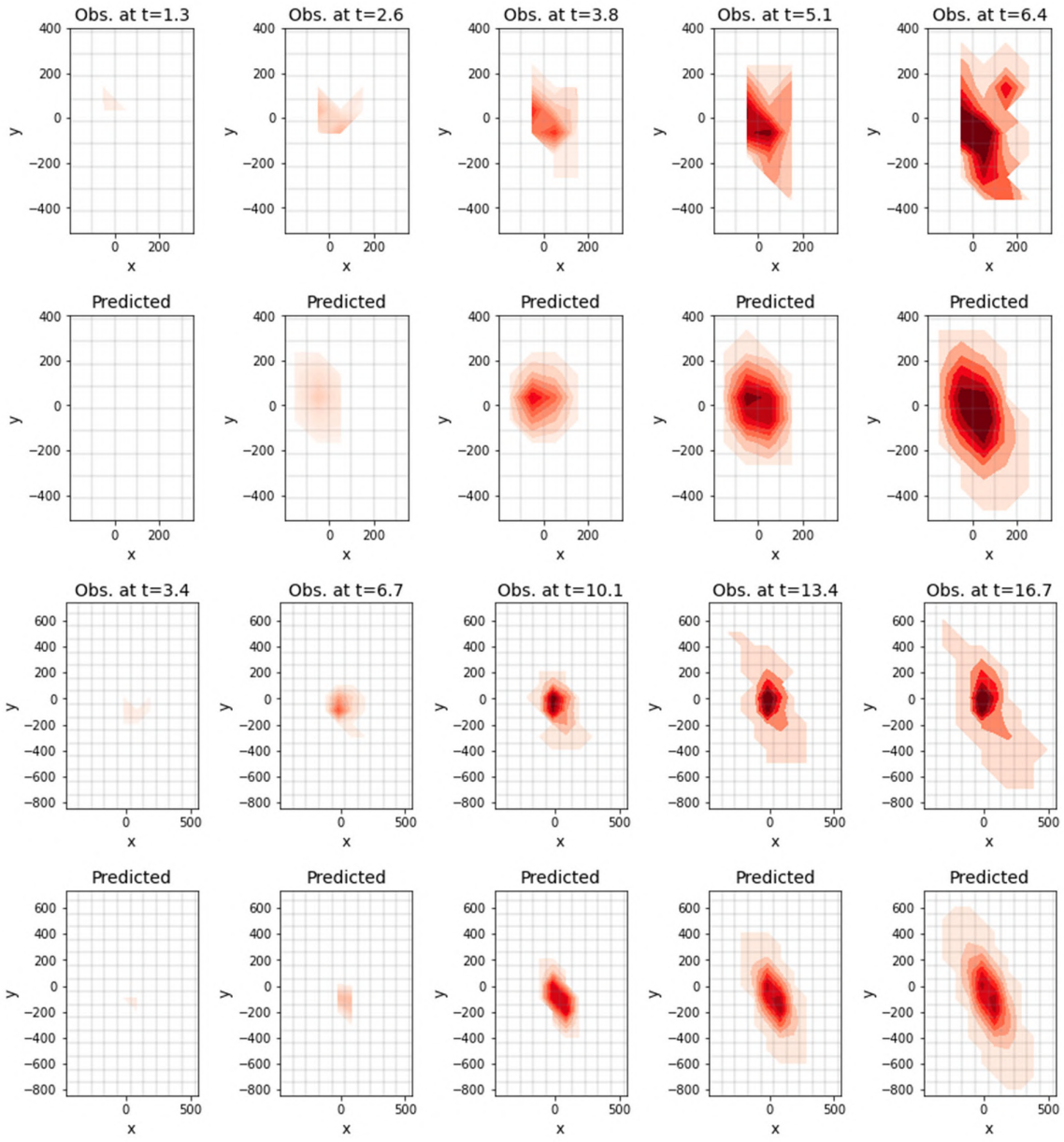

Figure 12. Cloud statistics on the training set, and prediction of the future cloud spatial distribution at t_{target} (within Δt bin) from (non-) linear regression extrapolation.

3. **Part 3:** ML1 model vs. EM1 in testbench environment

In this report, the best ML model found in Part 2, the Lasso model (ML1), is compared to the EM1 model in a standard pseudo-prospective framework. A forecast is done for each one of the injection experiments listed in Table 2 of Part 2. In comparison to the preliminary pseudo-prospective forecast done in Section 2.3 of Part 2, we here:

- Add uncertainty estimates using confidence intervals
- Use probability gain instead of MAE
- Compare ML1 (i.e., Lasso) to EM1
- Reframe ML1 for longer time horizons

We also reformat EM1 so that EM1_MLE is a special case of EM1_BH with uniform prior instead of Beta prior.

3.1 Data preparation

This section is identical to section 2.1 of Part 2. For each injection dataset, we import the raw data and define the array `data` with variables:

- `data[0,:]`: Earthquake's time `t` in decimal days from day of start of injection
- `data[1,:]`: Earthquake's magnitude `m`
- `data[2,:]`: Flow rate `dV` in m³/day (m³ to match a_{fb} unit, /day to match ATLS timeseries unit)
- `data[3,:]`: Cumulative volume `cumV` in m³
- `data[4,:]`: Pressure `P` in bars

Flow rate, cumulative volume and pressure are interpolated at the earthquake occurrence times. Note that we make the distinction between two types of injections because of different interpolation structures and different ML training/testing strategies:

- `data1_files`: Long individual continuous injections (Basel and Soultz).
- `data2_files`: Short successive injections (Bedretto Underground Lab, FORGE)

3.2 EM1 reformatting and testing

We here reproduce the EM1 model(s). However, in contrast to previous analyses, we consider EM1_MLE and EM1_BH to be the same linear model with same NHPP likelihood function but with different priors, EM1_MLE having a uniform prior for all parameters. The model is first restructured to have the same `fit` and `predict` functions as in scikit-learn to improve comparability with the ML models of Part 2. For further comparison with Part 2, we will test EM1 on the same training set - test set split done there in a first stage.

3.2.1 EM1 versus Lasso in standard training-test split

Before comparing the performance of Lasso versus EM1 in a pseudo-prospective forecast framework, let us first compare them using the same training set - test set splits done in Section 2.2 of Part 2.

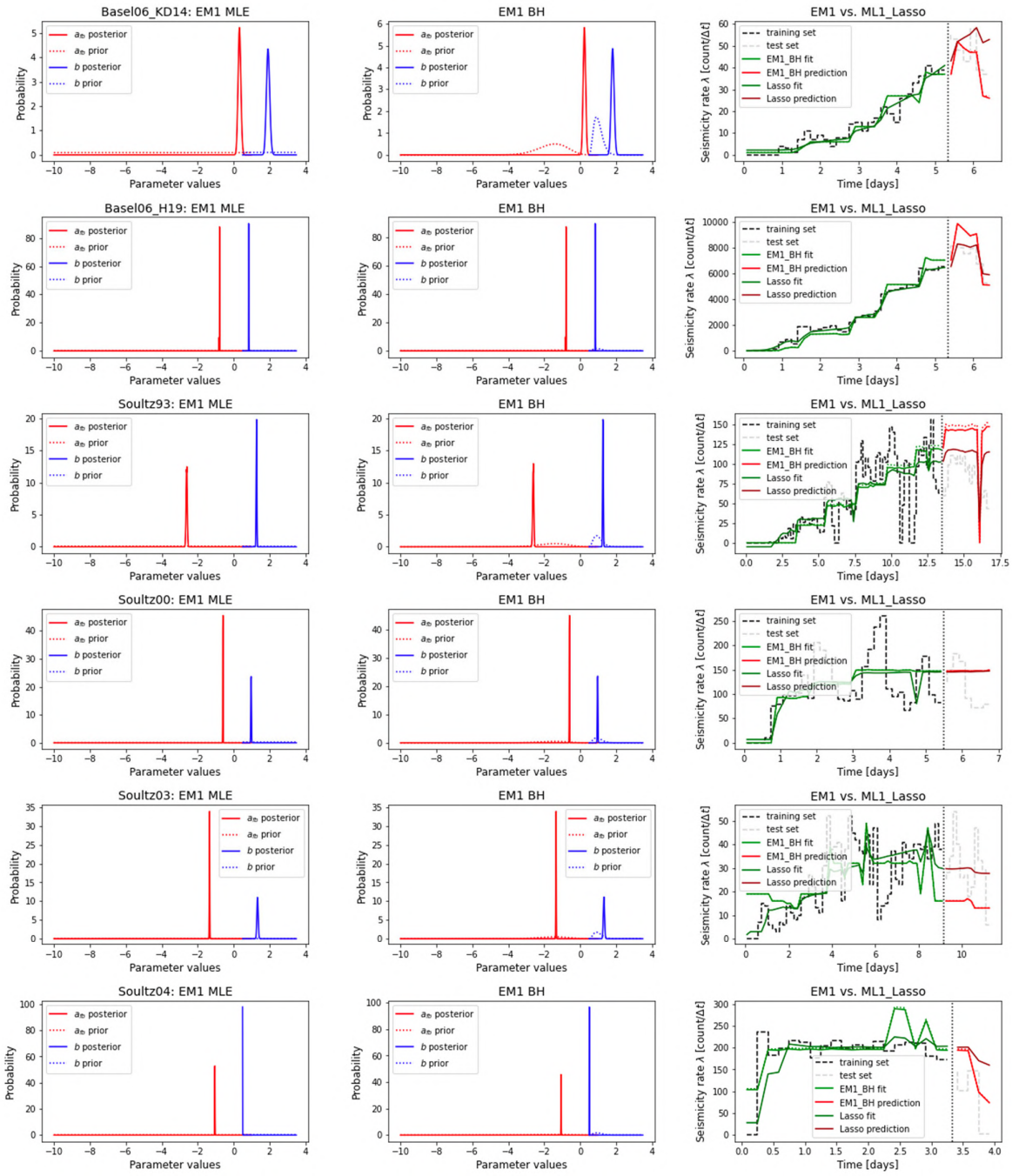

Figure 1a. Comparison of EM1 and ML1 (Lasso, Ridge) forecasting for data1_files using a standard training set-test set split (Lasso features the same as in Deliverable 3, Section 2.2). Green and red used for training and testing, respectively, for EM1 with EM1_BH solid and EM1_MLE dotted lines. Dark green and dark red used for training and testing, respectively, for ML1 with Lasso solid and Ridge dotted lines.

** MAE results **

	Basel06_KD14	Basel06_H19	Soultz93	Soultz00	Soultz03	Soultz04
EM1_MLE	8.0	1131.6	63.4	44.6	16.2	66.5
EM1_BH	8.3	1131.6	58.4	44.6	16.2	65.0
ML1_Lasso	8.5	502.3	30.6	44.5	9.1	83.8

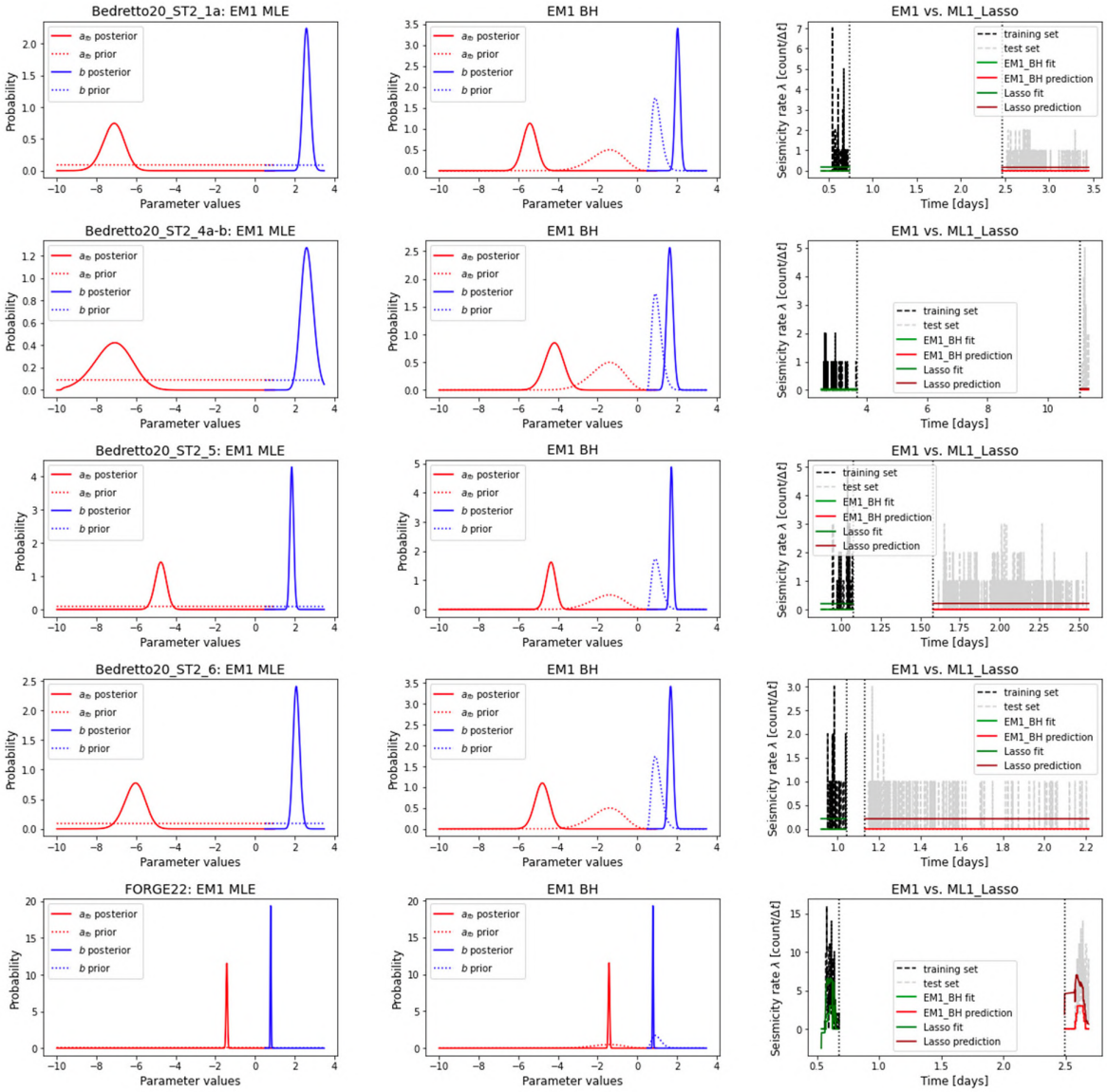

Figure 1b. Comparison of EM1 and ML1 (Lasso, Ridge) forecasting for data2_files using the first sub-injection as training set and second sub-injection as test set (ML1 features the same as in Deliverable 3, Section 2.2). Green and red used for training and testing, respectively, for EM1 with EM1_BH solid and EM1_MLE dotted lines. Dark green and dark red used for training and testing, respectively, for ML1 with Lasso solid and Ridge dotted lines.

** MAE results **

	Bedretto20_ST2_1a	Bedretto20_ST2_4a-b	Bedretto20_ST2_5	Bedretto20_ST2_6	FORGE22
EM1_MLE	0.1	0.4	0.2	0.1	2.1
EM1_BH	0.1	0.4	0.2	0.1	2.1
ML1_Lasso	0.2	0.4	0.4	0.3	3.7

We do not observe any model performing systematically better when using a standard training set-test set split and the same set of features of ML1 as used in Section 2.2.3 of Part 2. Depending on the site, EM1_MLE, EM1_BH or ML1_Lasso can be the best model.

An interesting side-observation is that the Beta prior in EM1_BH does not necessarily improve the forecast compared to an uniformed prior (EM1_MLE).

3.3 Model comparison in pseudo-prospective forecast¶

3.3.1 Rate forecast in the next Δt bin

We now compare EM1 and ML1_Lasso in a pseudo-prospective framework, using log-likelihood and probability gain as performance metrics.

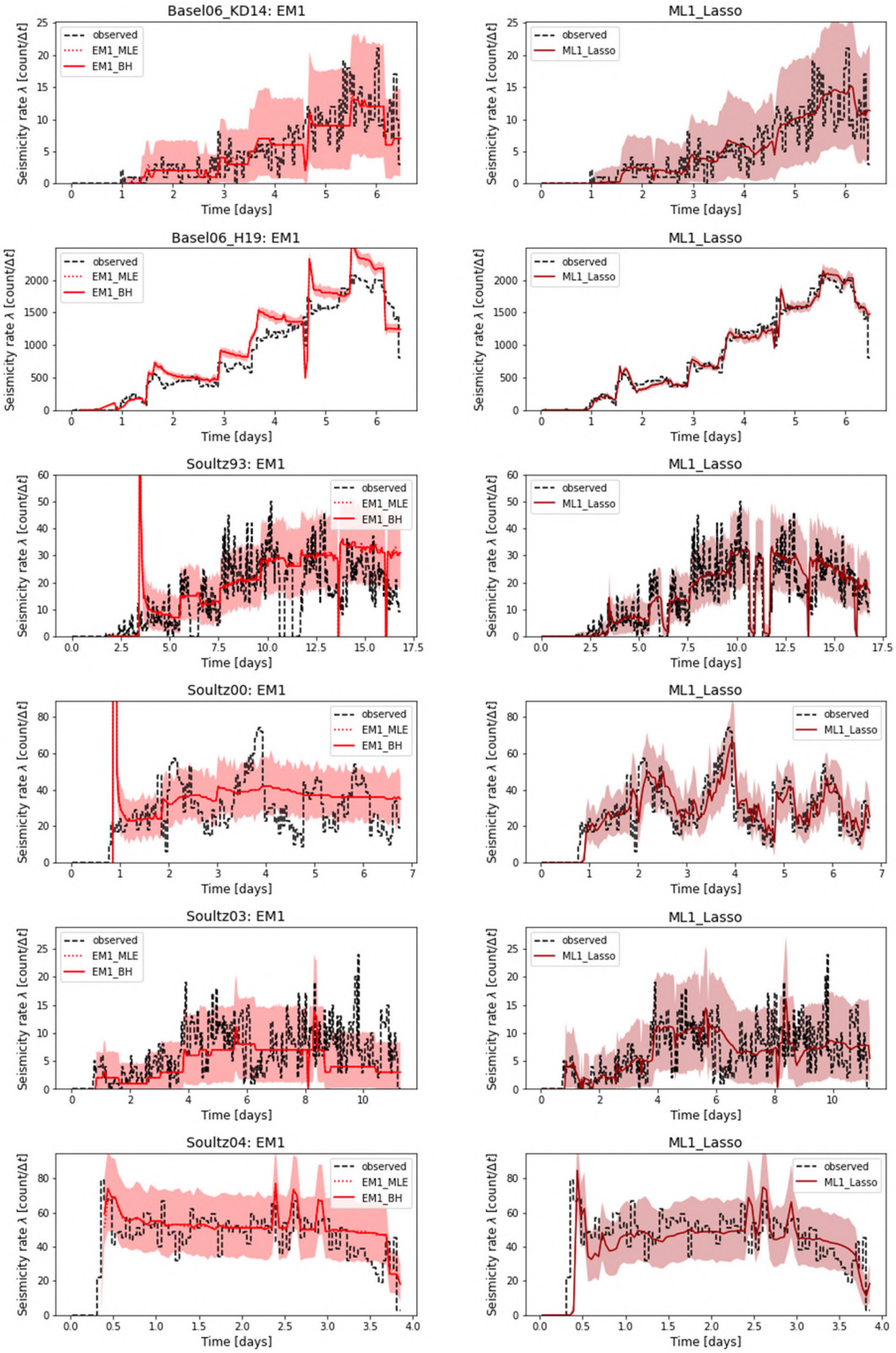

Figure 2a. Pseudo-prospective forecast (for data1_files) with forecast realised for t_{target} and the training for $t<t_{target}$.

We finally provide plots similar to Figs. 3 and 6 of the DEEP deliverable 3.1:

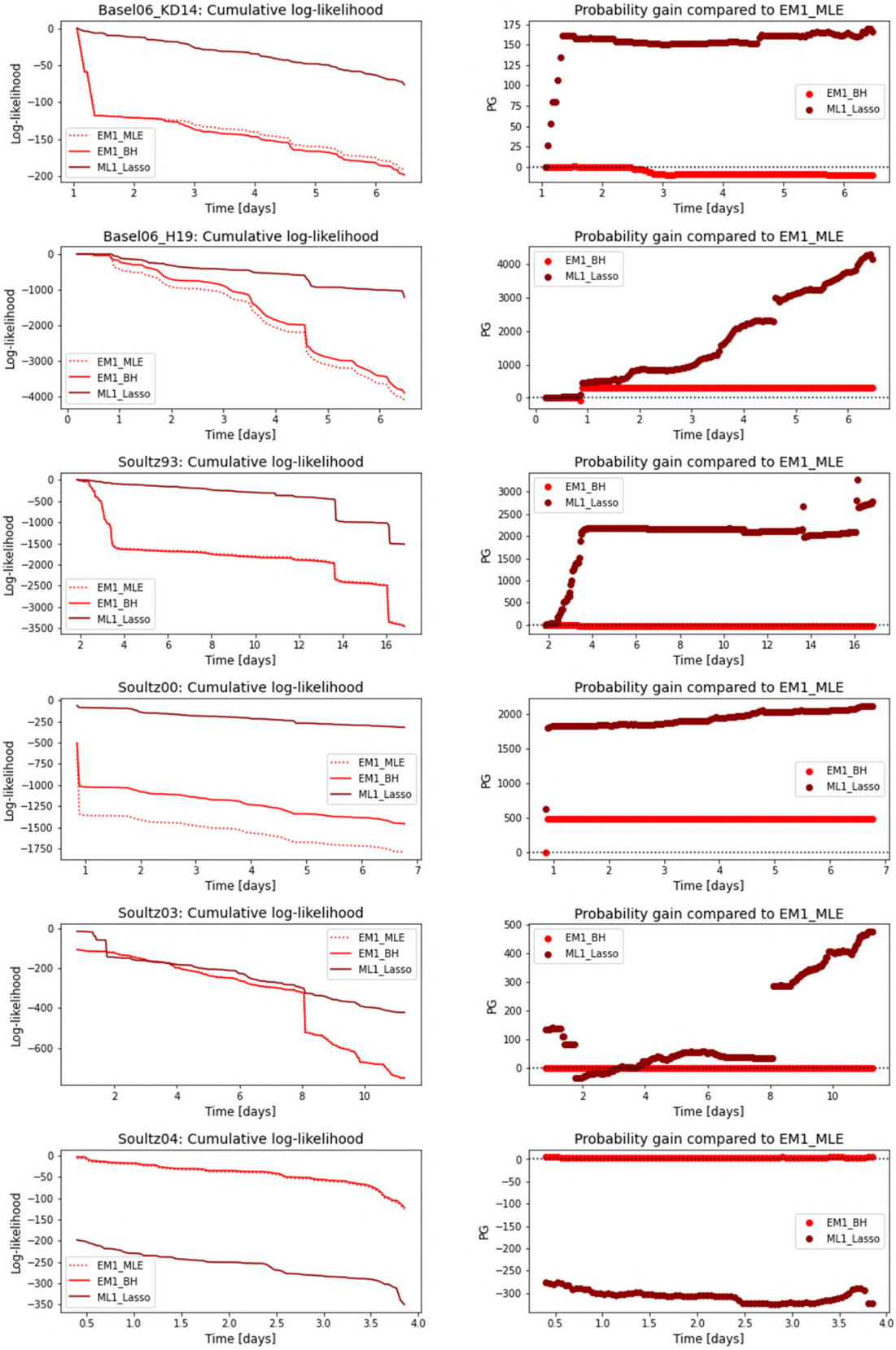

Figure 3a. From left to right: Column 1) Cumulative value of the log-likleihood. Column 2) Cumulative plot of the punctual probability gain (PG) The PG is calculated against the EM1_MLE model.

We then rerun everything for data2_files:

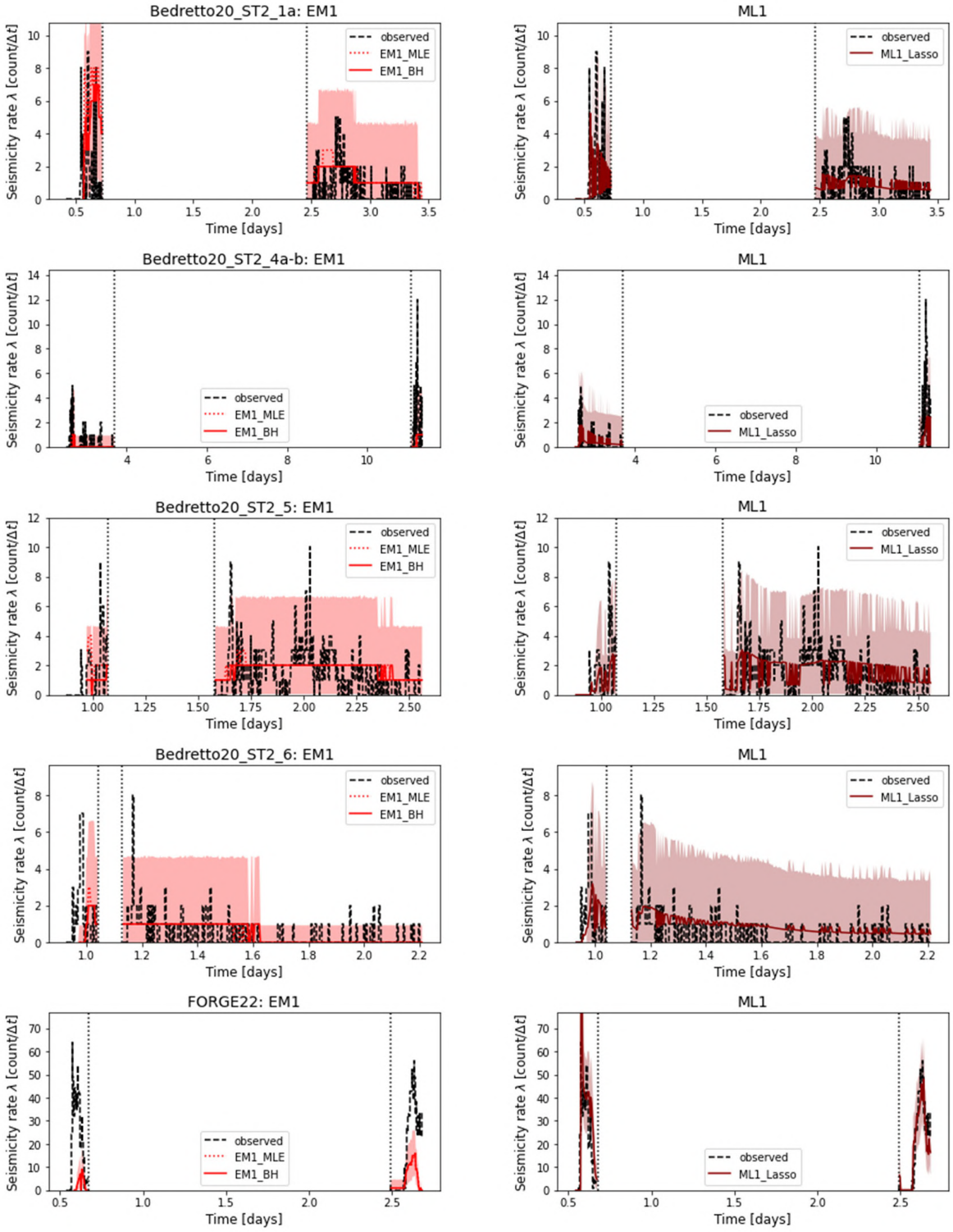

Figure 2b. Pseudo-prospective forecast (for data2_files) with forecast realised for t_{target} and the training for $t<t_{target}$.

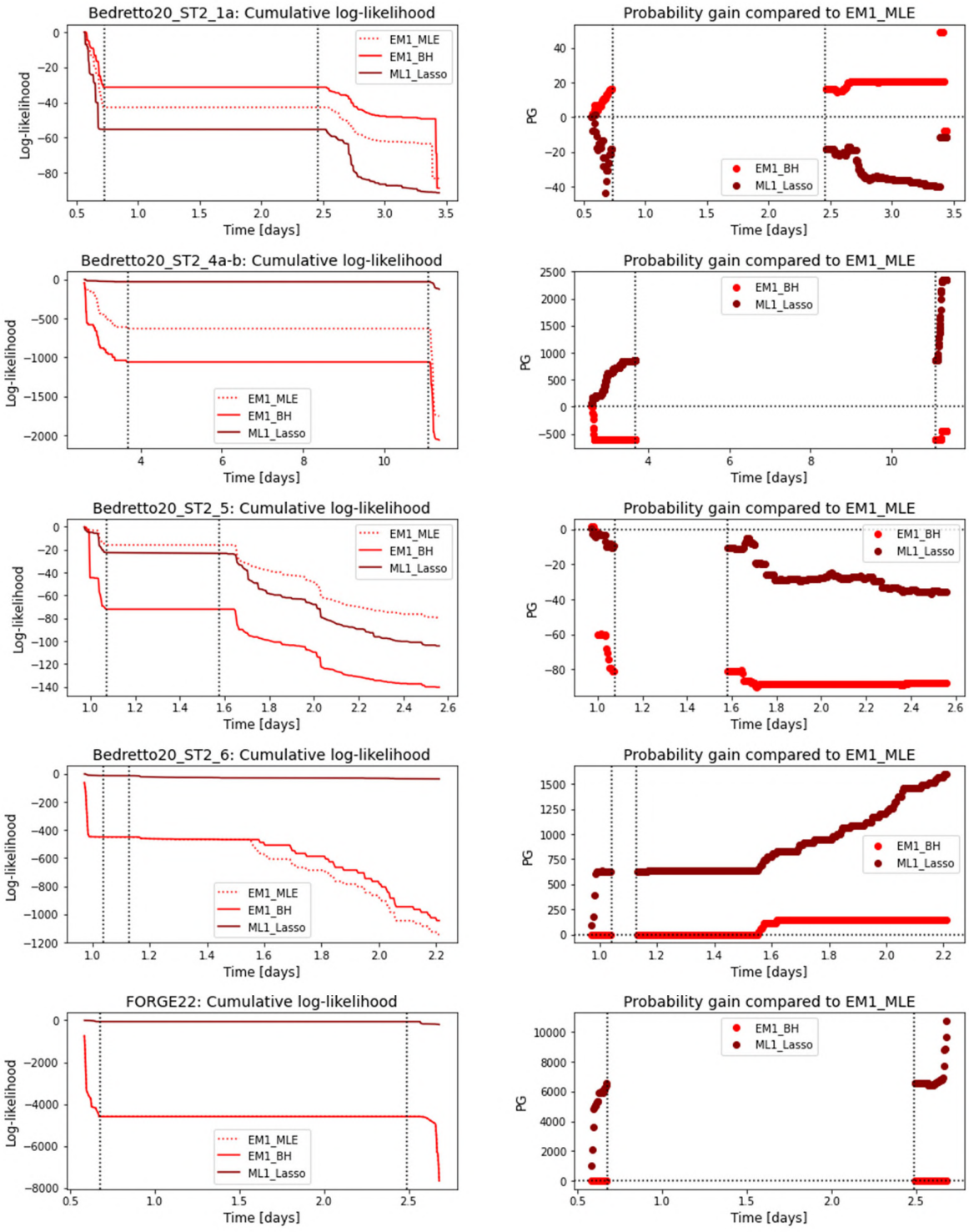

Figure 3b. From left to right: Column 1) Cumulative value of the log-likleihood. Column 2) Cumulative plot of the punctual probability gain (PG) The PG is calculated against the EM1_MLE model.

FORGE22 case: The FORGE22 dataset is rich in seismicity in a very short time window. Hence, we used a time bin significantly smaller than for Bedretto with a forecast per minute instead of per hour. In Figure 4 we show the Lasso forecast on FORGE22, zooming into the 2 injection periods.

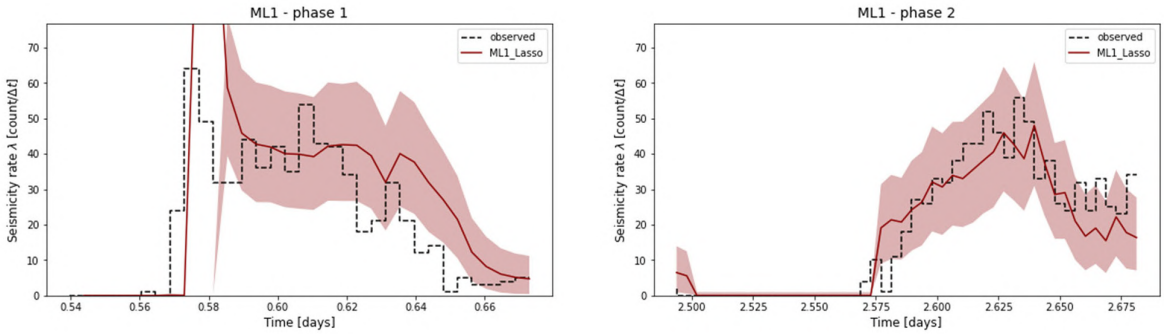

Figure 4. Lasso forecast on FORGE22, zooming into the 2 injection periods.

We observed that, in terms of log-likelihood, **the Lasso model (ML1_Lasso) performs better than EM1 for pseudo-prospective forecasting in most cases for small Δt. For longer Δt, Lasso performs better in about half of the cases**. This suggests that implementing an ML strategy within an ATLS is worthwhile exploring in the near future.

3.3.2 ML1 reframing for longer time horizon

One important advantage of EM1 is its ability to forecast the seismicity rate in any future time bin Δt since it is solely controlled by the planned flow rate $\dot{V}$ in the future. Lasso, in addition to this feature, employs features observed in the previous bin or in all previous bins. It therefore becomes ill-defined to make prediction on time horizons longer than one Δt bin. To correct for this problem, a "multi-Lasso" model is proposed with each "sub-Lasso" predicting a different bin in the future:

- ML1_t0: predicts $y(t_{next})$
- ML1_t1: predicts $y(t_{next}+1\times\Delta t)$
- ML1_t2: predicts $y(t_{next}+2\times\Delta t)$
- ML1_t3: predicts $y(t_{next}+3\times\Delta t)$
- ML1_t4: predicts $y(t_{next}+4\times\Delta t)$

This is tested on `data1_files` for illustration purposes.

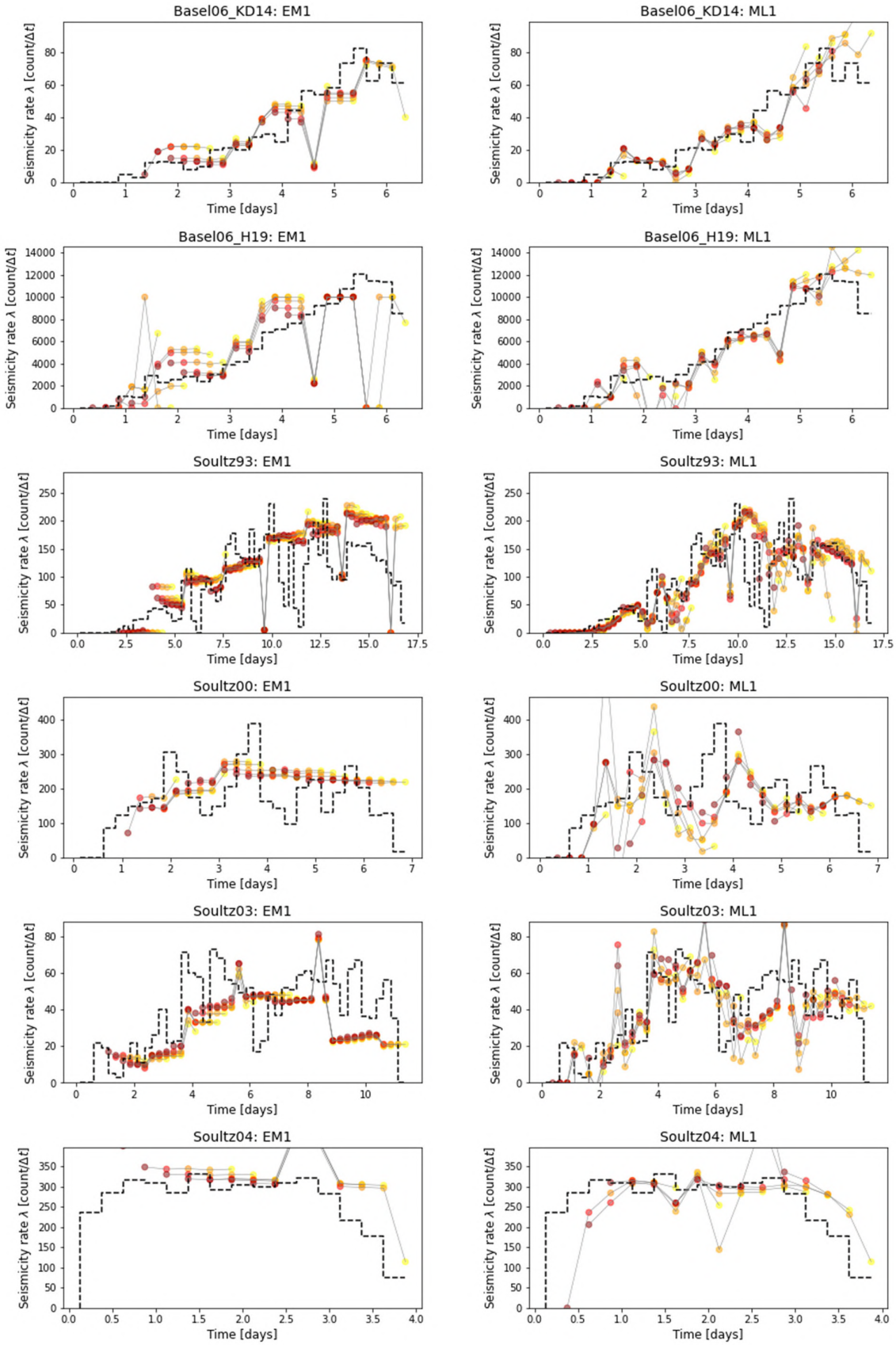

Figure 5. Comparison between EM1 (left) and Multi-Lasso model (right) for different bins in the future.

We observe that the multi-Lasso works properly and can forecast in longer time horizons. Results remain better than EM1 in general. We show below that as the time horizon increases, the Lasso model increases the relative weight of the planned flow rate in most cases. This is for example well illustrated in the Soultz 2000 scenario for dt = 1/24 (Figure 6).

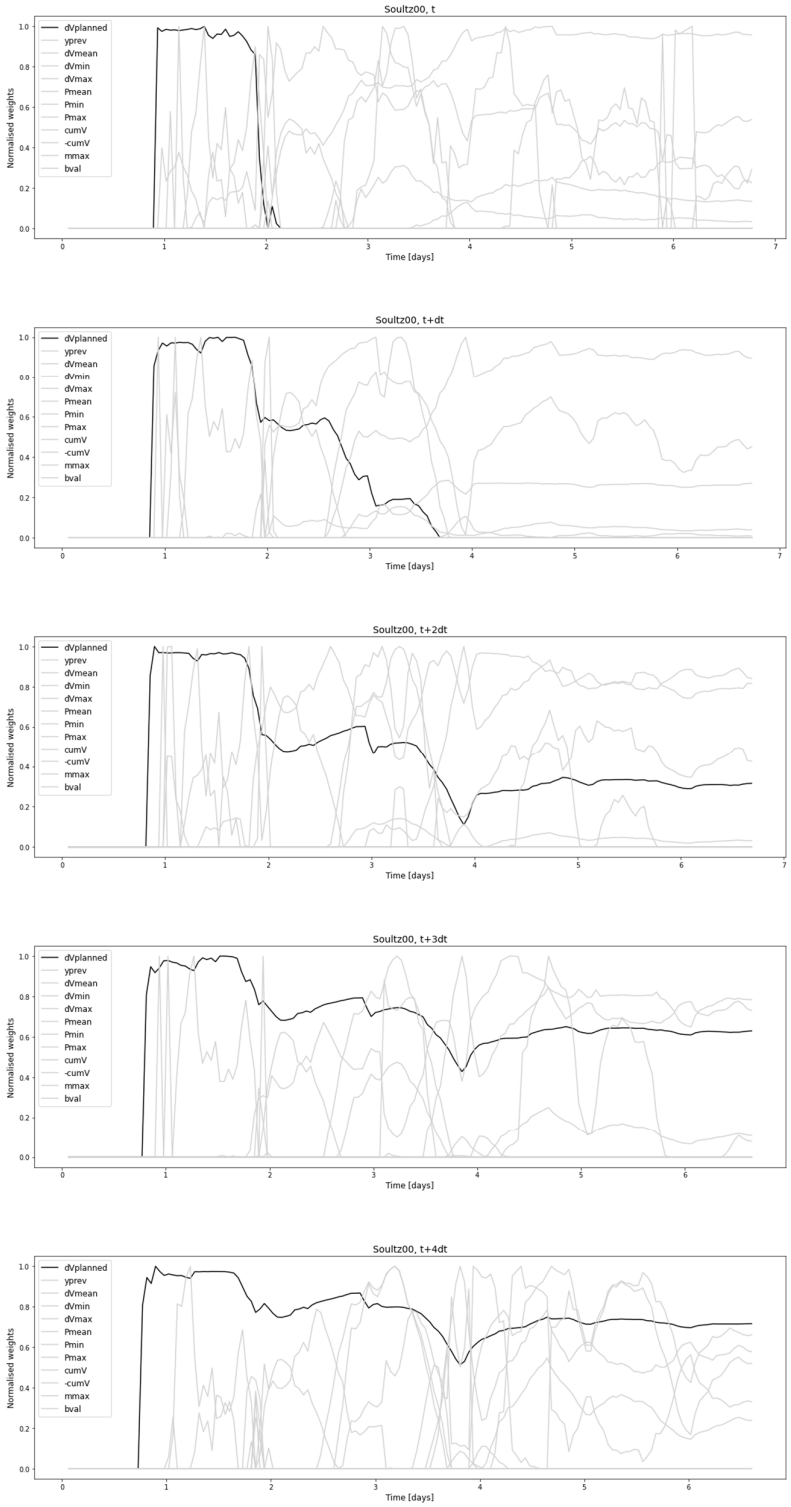

Figure 6. Evolution of the relative role of each feature over time (for Soultz00). Black thick line shows the planned flow rate as the time horizons increases.

4. **Part 4:** Machine Learning model training and testing on simulations

In this section, the ML models evaluated on real induced seismicity data in Parts 2-3 are now tested on physics-based simulations. We will take advantage of the transparency of ML1_Lasso to study the evolution of feature weights over time and their potential changes in different scenarios. Seismicity cloud forecasting will also be investigated in more depth to explore the role of the presence or not of a fault.

4.1 Data Loading, preparation, and visualisation

We first load and visualise all the data. For improved readability, half of the simulations will be analysed at a time (group = Variable_Permeability and group = Fixed_Permeability). In the following, for illustration purposes we only show results of the ML models for the Variable_Permeability group.

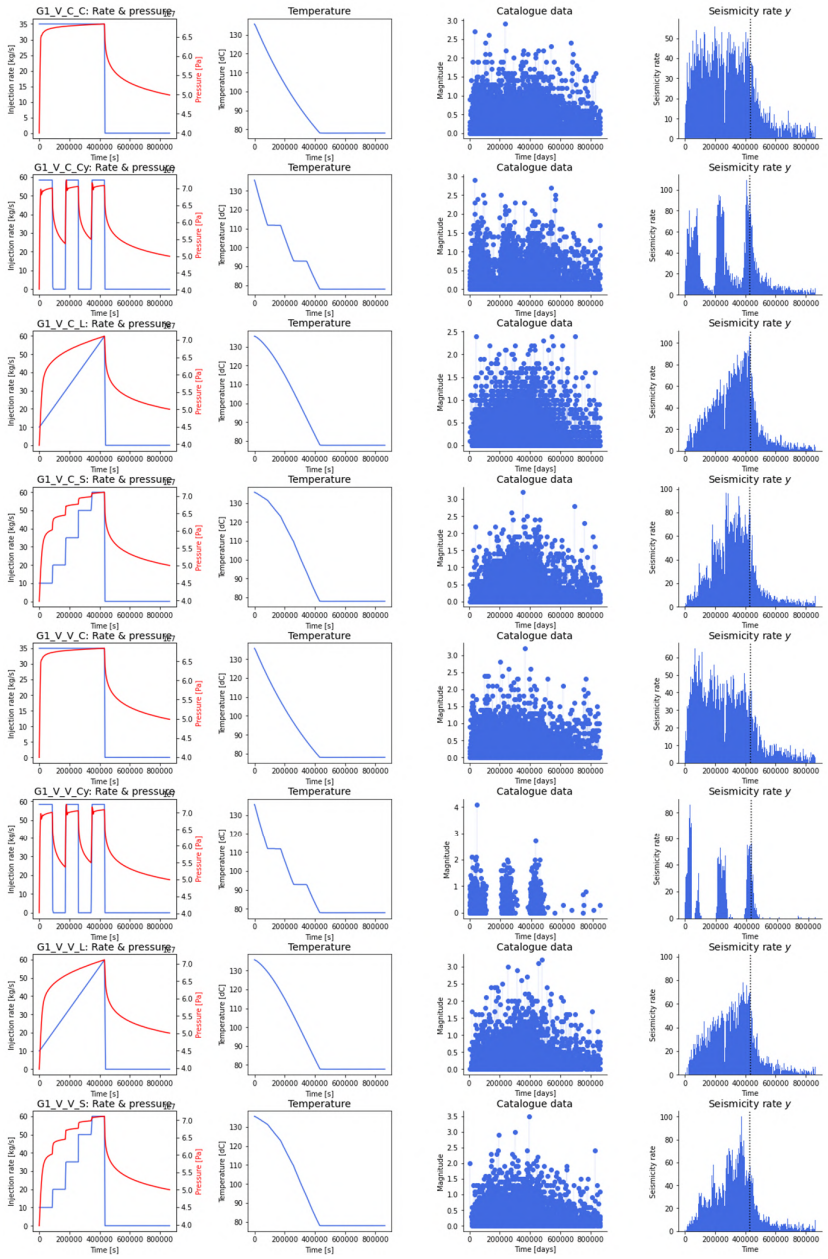

Figure 1a. Simulation data for **homogeneous medium (G1).** Each row shows a simulation obtained with different initial parameters: G1 = homogeneous; V= Variable Permeability/ Variable b value (if in 3rd position in the acronym), C = No Repeaters/ Constant (if in 4th position in the acronym), Cy = Cyclic, L= Linear, S= step; G2 = Fault.

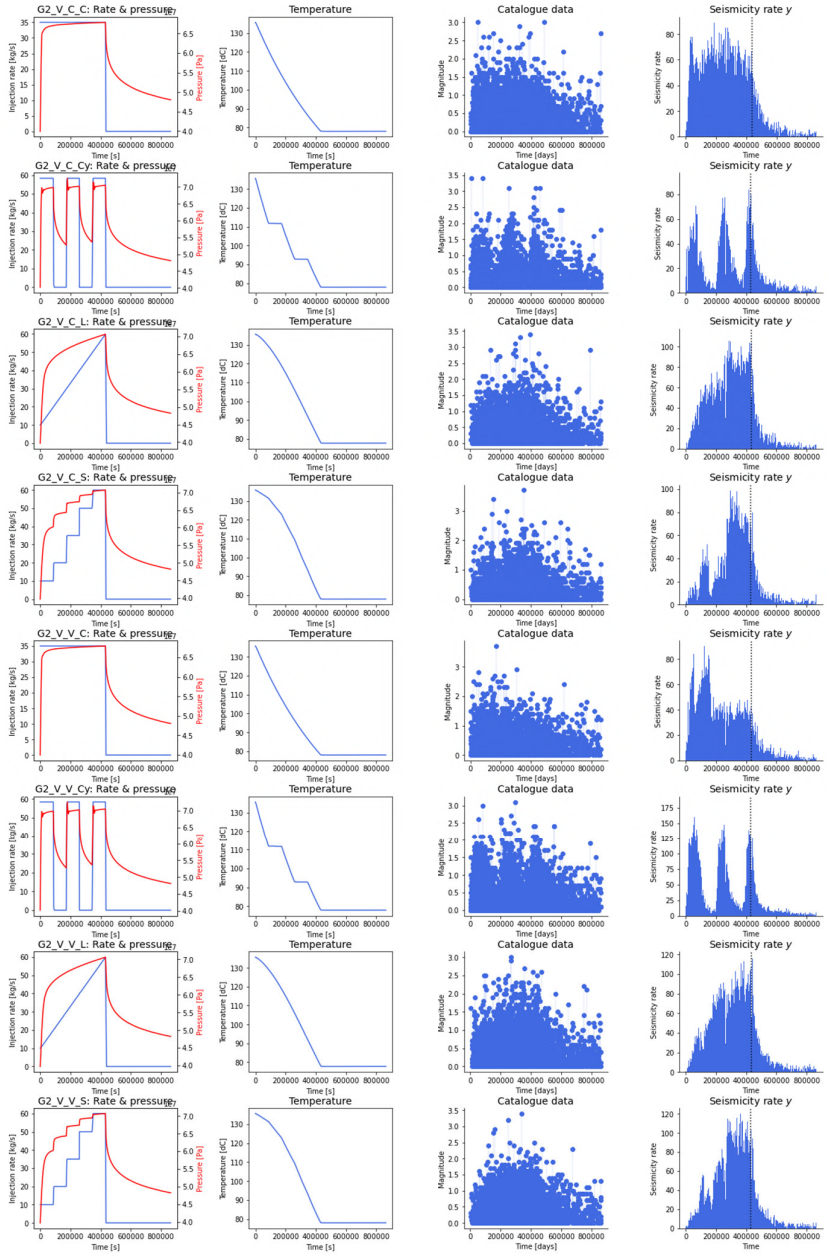

Figure 1b. Simulation data considering the **presence of a fault (G2).** Each row shows a simulation obtained for different initial parameters: G1 = homogeneous; V= Variable Permeability/ Variable b value (if in 3rd position in the acronym), C = No Repeaters/ Constant (if in 4th position in the acronym), Cy = Cyclic, L= Linear, S= step; G2 = Fault.

Note that temperature is an additional potential feature (compared to real data) in the case of physics-based simulations (but not gas saturation nor enthalpy - no information provided in those data fields).

Due to the cyclical scenario, the data preparation for ML must be modified compared to Parts 2-3 (i.e., no simple cutoff at a specific shutin time). The result of the process is shown in Figure 2. Similarly to ML applications to real data, only phases of non-zero flow rates are considered for the present analysis.

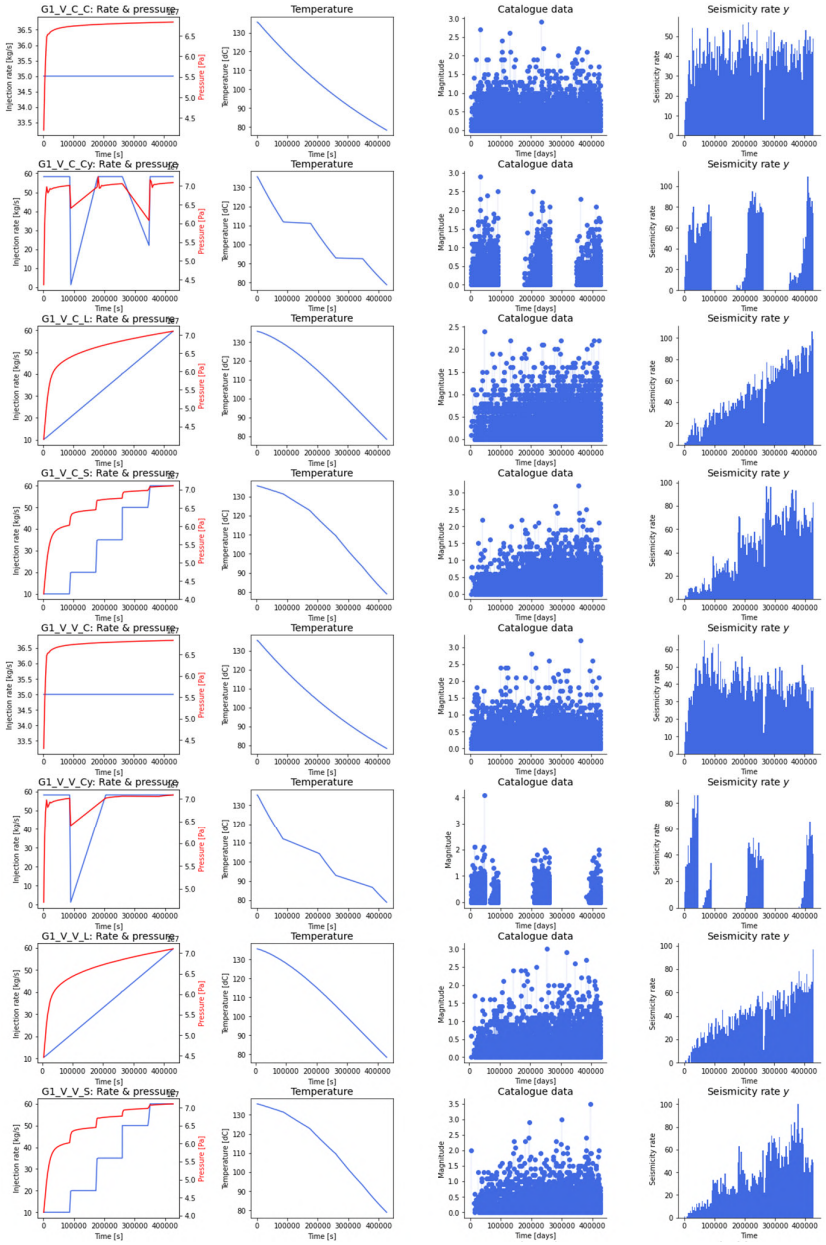

Figure 2a. Data preparation (simulations group G1).

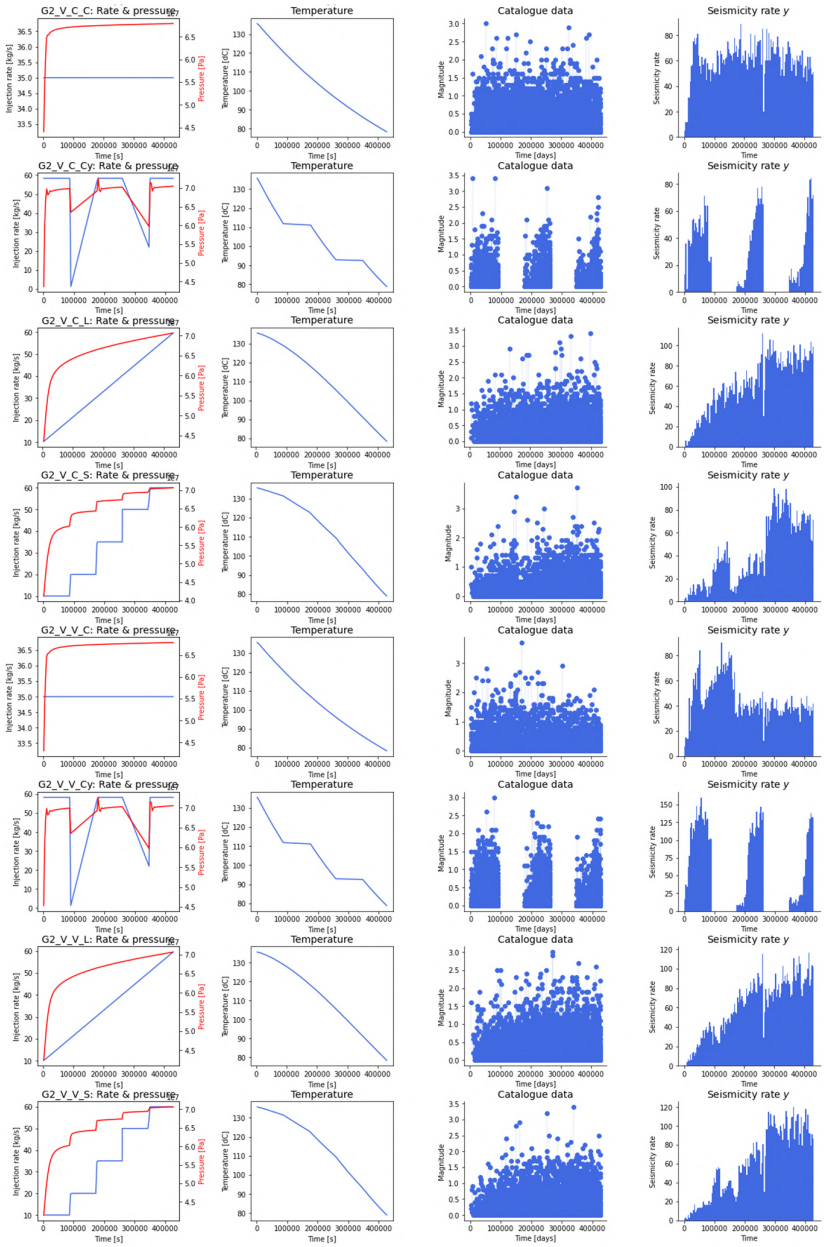

Figure 2b. Data preparation (simulations group G2)

4.2 Lasso model application

4.2.1 Lasso pseudo-prospective forecasting

We now apply ML1_Lasso on simulations. We can expect similar results as for real induced seismicity.

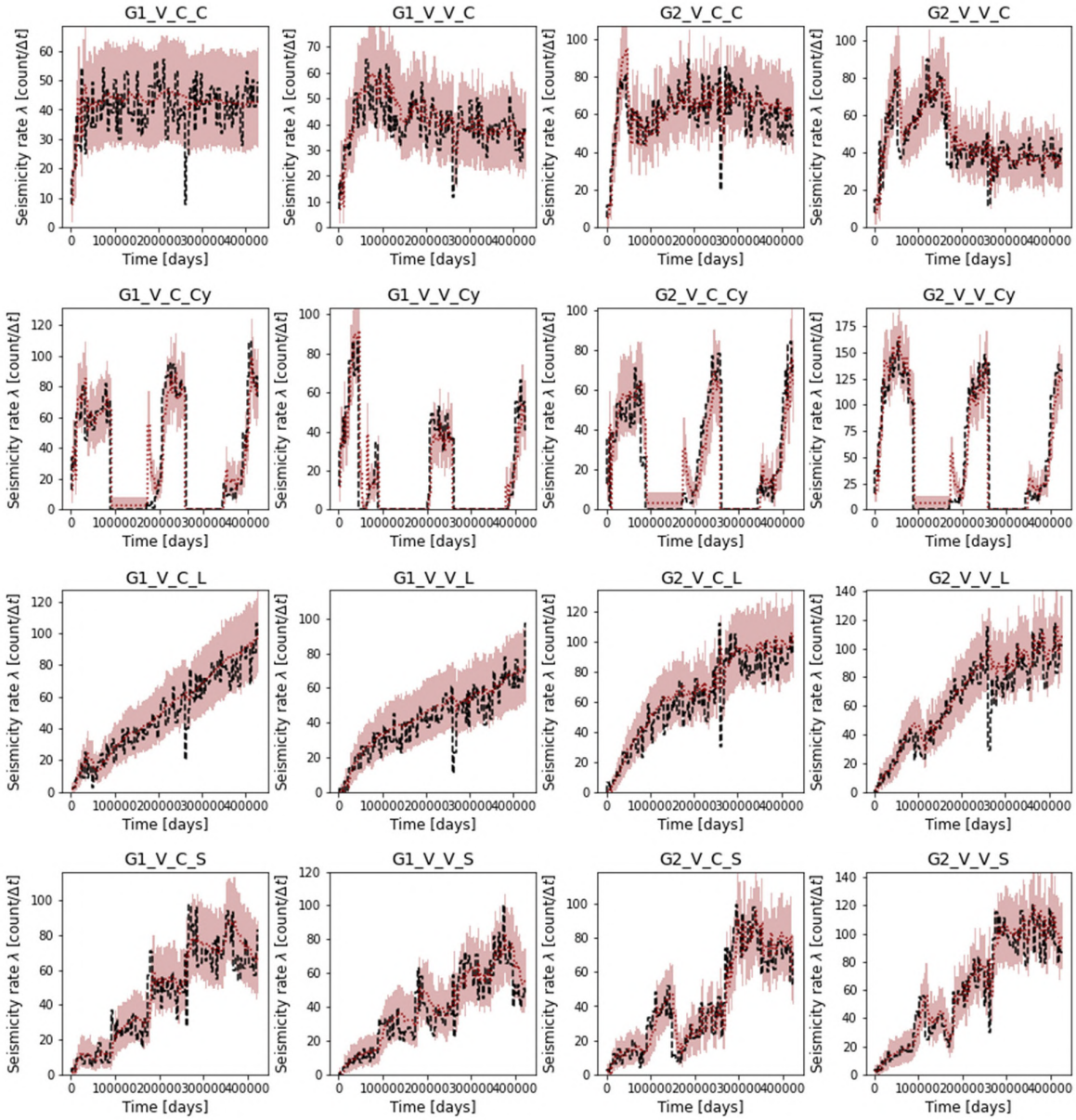

Figure 3. Pseudo-prospective forecast (for all simulation group, Variable Permeability)

4.2.2 Lasso weight analysis¶

An interesting aspect of Lasso is its transparency. It allows us to see which features play a significant role in a physics-based simulation (Figure 4)

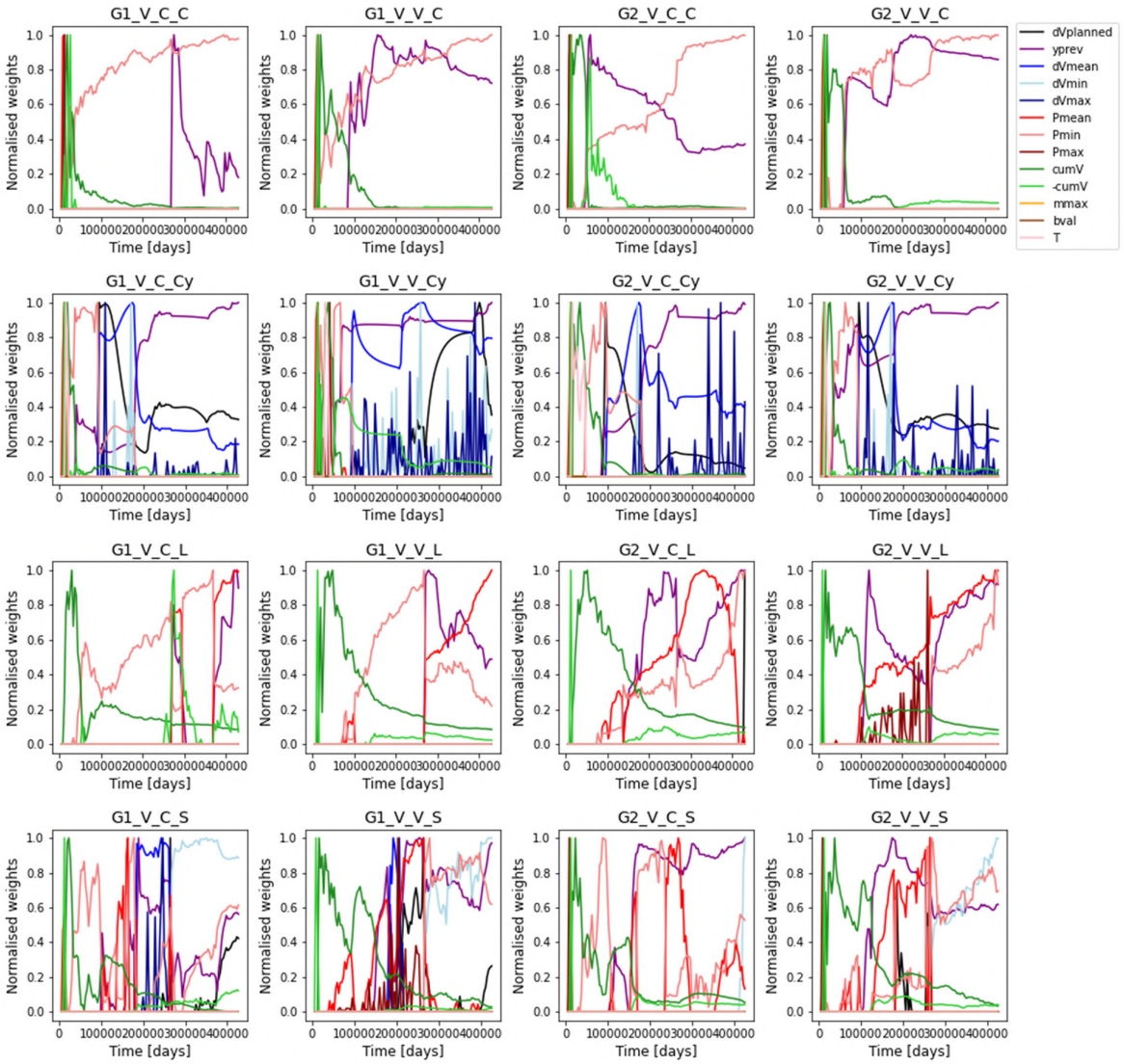

Figure 4. Evolution of the relative role of each feature over time (for all simulations groups, Variable Permeability)

- For the variable permeability: We observe that Lasso weight evolutions change significantly between constant, cyclic, linear and step. However for each one of these conditions, the same patterns remain when there is a fault or not, and when the b-value varies or not. This indicates that the injection profile controls the dynamics of the seismicity at first order.
- For the fixed permeability (here not shown): We observe similar patterns as for the variable permeability, except (i) for the cyclic cases and (ii) that much more variability is observed per injection type for fault/no-fault, and b-value variation/or not.

Summary tables are generated below to show which features have a non-zero weight per simulation.

	dVplanned	yprev	dVmean	dVmin	dVmax	Pmean	Pmin	Pmax	cumV	-cumV	mmax	bval	T
G1_V_C_C		x				x	x	x	x	x			
G1_V_C_Cy	x	x	x	x	x	x	x	x	x	x			
G1_V_C_L		x				x	x		x	x			
G1_V_C_S	x	x	x	x	x	x	x		x	x			
G1_V_V_C		x				x	x	x	x	x			
G1_V_V_Cy	x	x	x	x	x	x	x	x	x	x			
G1_V_V_L		x				x	x		x	x			
G1_V_V_S	x	x	x	x	x	x	x	x	x	x			
G2_V_C_C		x				x	x	x	x	x			
G2_V_C_Cy	x	x	x	x	x	x	x	x	x	x			
G2_V_C_L	x	x				x	x		x	x			
G2_V_C_S		x		x		x	x	x	x	x			
G2_V_V_C		x				x	x	x	x	x			
G2_V_V_Cy	x	x	x	x	x	x	x	x	x	x			
G2_V_V_L		x				x	x	x	x	x			
G2_V_V_S	x	x		x		x	x	x	x	x			

Results may significantly differ for other time bins dt. It would seem preferable to first reformulate the Lasso model as a point-process model (as EM1) to remove the role of this hyperparameter, before interpreting the Lasso weight patterns. This remains however outside the scope of the present notebook. As a side note, if yprev is removed, the temperature T plays a role for the cyclical case as an indicator of delay between flow increase and seismicity occurrence.

4.2.3 Seismicity cloud forecasting

It is interesting to compare the seismicity cloud predicted for G1 (no fault) and G2 (fault) scenarios. We will first map the observed clouds (Figure 5) and for one G1 (no-fault) scenario and one G2 (fault) scenario, let us look at the evolution of the seismicity cloud through time (Figure 6).

We note that the possibility for two bivariate Gaussians in the fault case is worth exploring. We will therefore modify the codes of Part 2 to account for this possibility by taking advantage of the Gaussian Mixture Model (GMM) which is already coded there. We can also notice that the non-linear regression used previously to model the seismicity cloud expansion remains valid in the simple one-cloud scenario. It is likely that it is also valid in the 2-component case, which is likely the combination of two nonlinear regressions (mostly visible on the variable permeability example).

We start with the simple one-cloud forecast (G1), Variable_Permeability case only (Figure 7). Figure 7 verifies that a simple seismicity cloud can be forecasted using the method developed in Part 2.

We then continue with the two-cloud forecast (G2). However, if we apply the same method as above, we notice that the forecasted covariance matrix is generally not positive definite. The problem may also apply to special one-cloud cases, which we would observe for the fixed permeability case for example. We focus on solving this issue specifically for the G2 case, although we will also need to check that it works for the G1/fixed permeability case.

The problem is that forecasting variances and covariances by disjoint models (as previoulsy done) is not guaranteed to be positive definite, which is however a required condition for the multi-variate normal distribution. A solution is to (1) use Cholesky decomposition, (2) do the training and forecast on the Cholesky factors, and (3) reconstruct the forecasted covariance matrix using the reverse operation of the Cholesky decomposition. This is therefore a much more demanding task. From the analysis of the evolution of the Cholesky factors over time, we observe many instabilities in the Cholesky factors which impact the polynomial model. This in turn makes it more difficult to correctly forecast a complex seismicity cloud (Figure 8) when combining multi-component GMM and Cholesky decomposition (especially in the fixed permeability case). Improvements will require a dedicated project. The proposed model can be thus considered as a proof-of-concept.

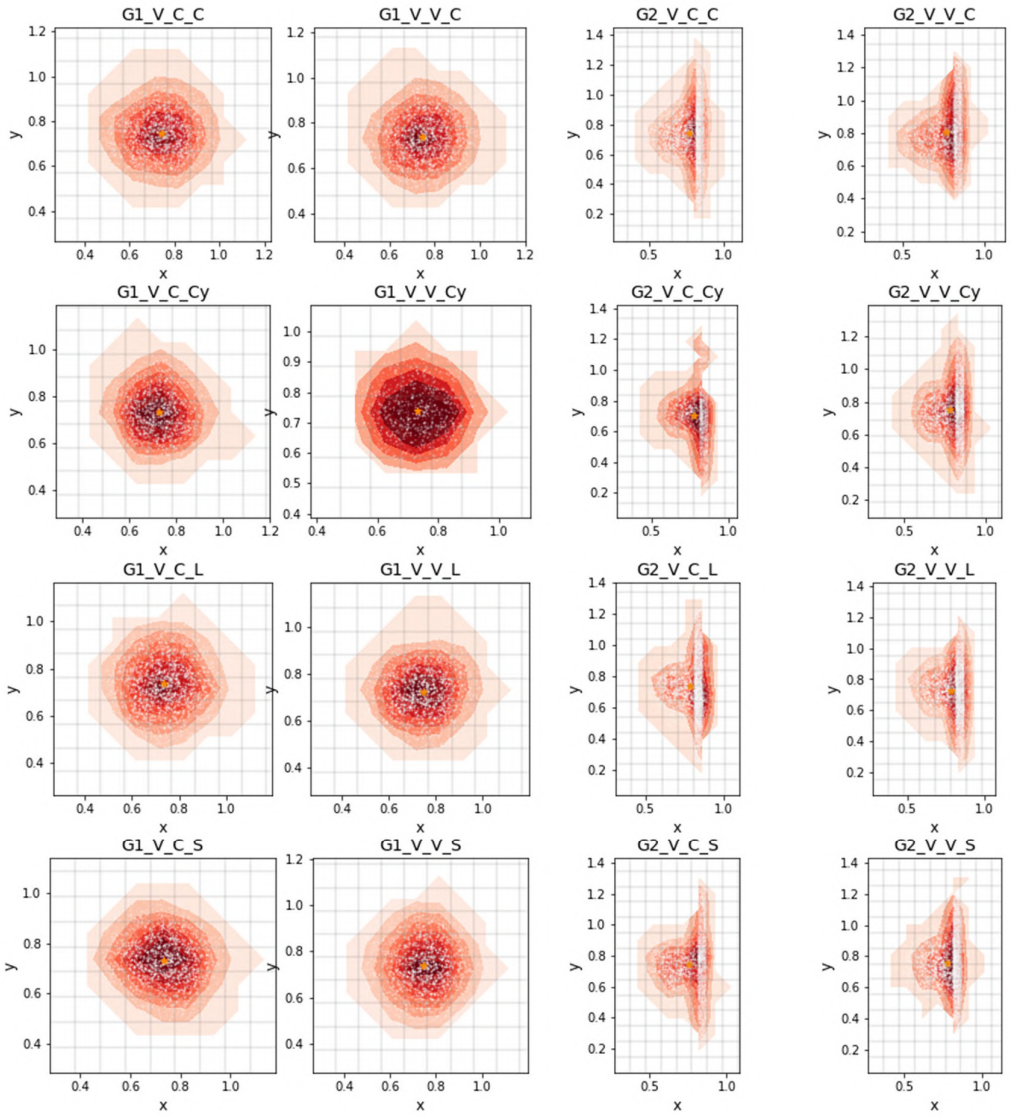

Figure 5. Observed seismicity clouds for G1 (no fault) scenario (two columns on the left) and for G2 (fault) scenario (two columns on the right) for all simulation groups.

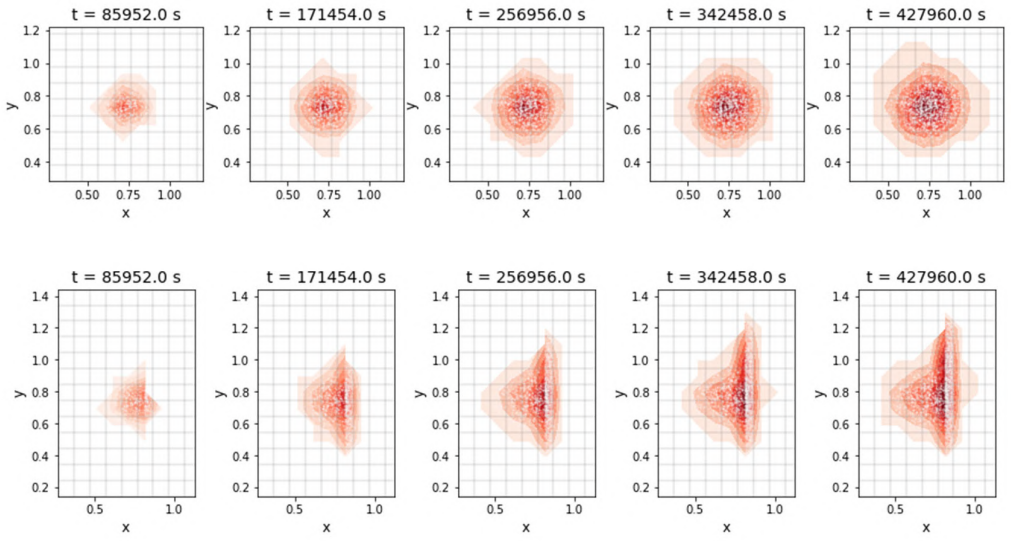

Figure 6. Evolution of the seismicity cloud through time for G1_V_V_C scenario (top) and for G2_V_V_C scenario.

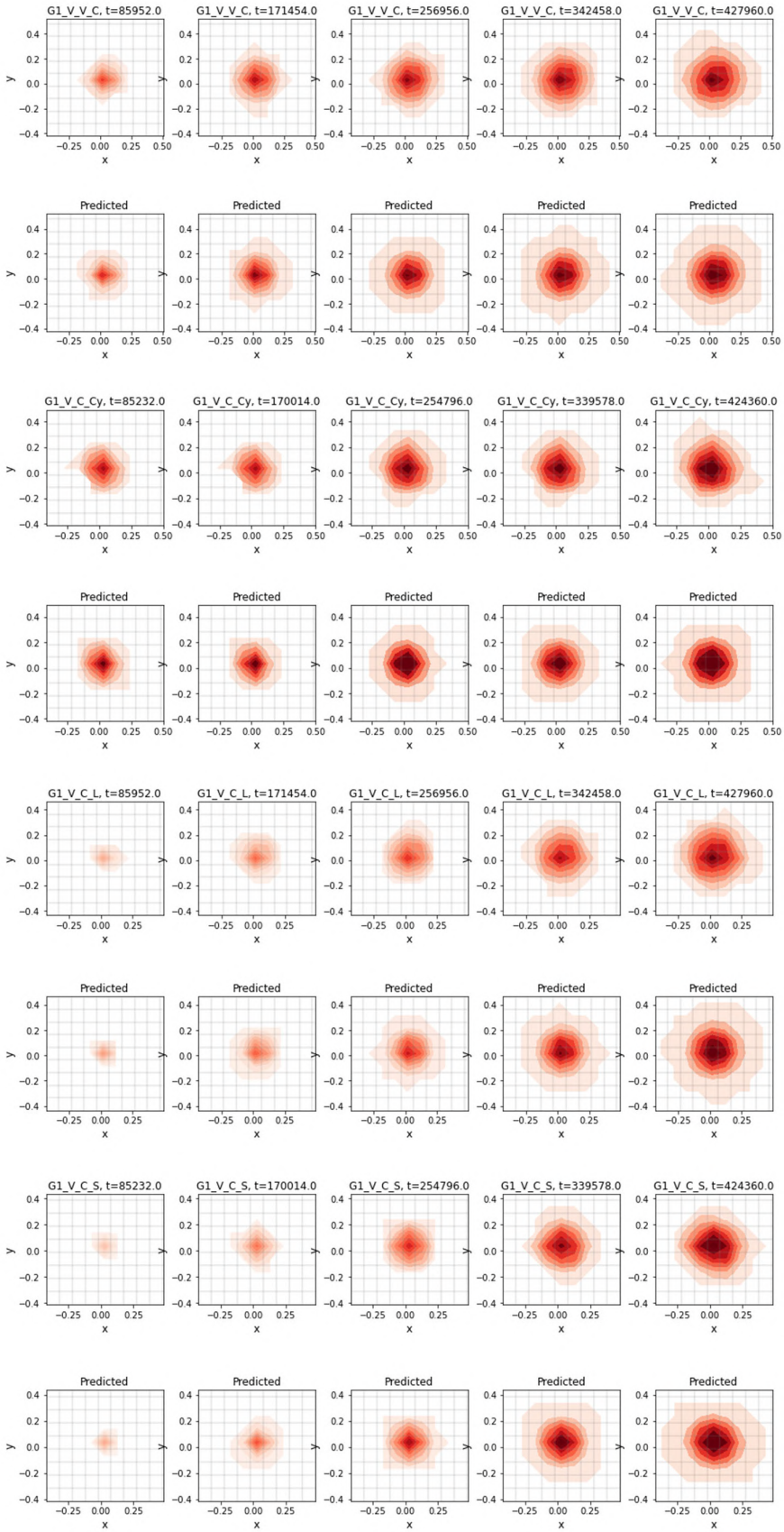

Figure 7. One-cloud forecast (G1), `Variable_Permeability` case only, all simulations group.

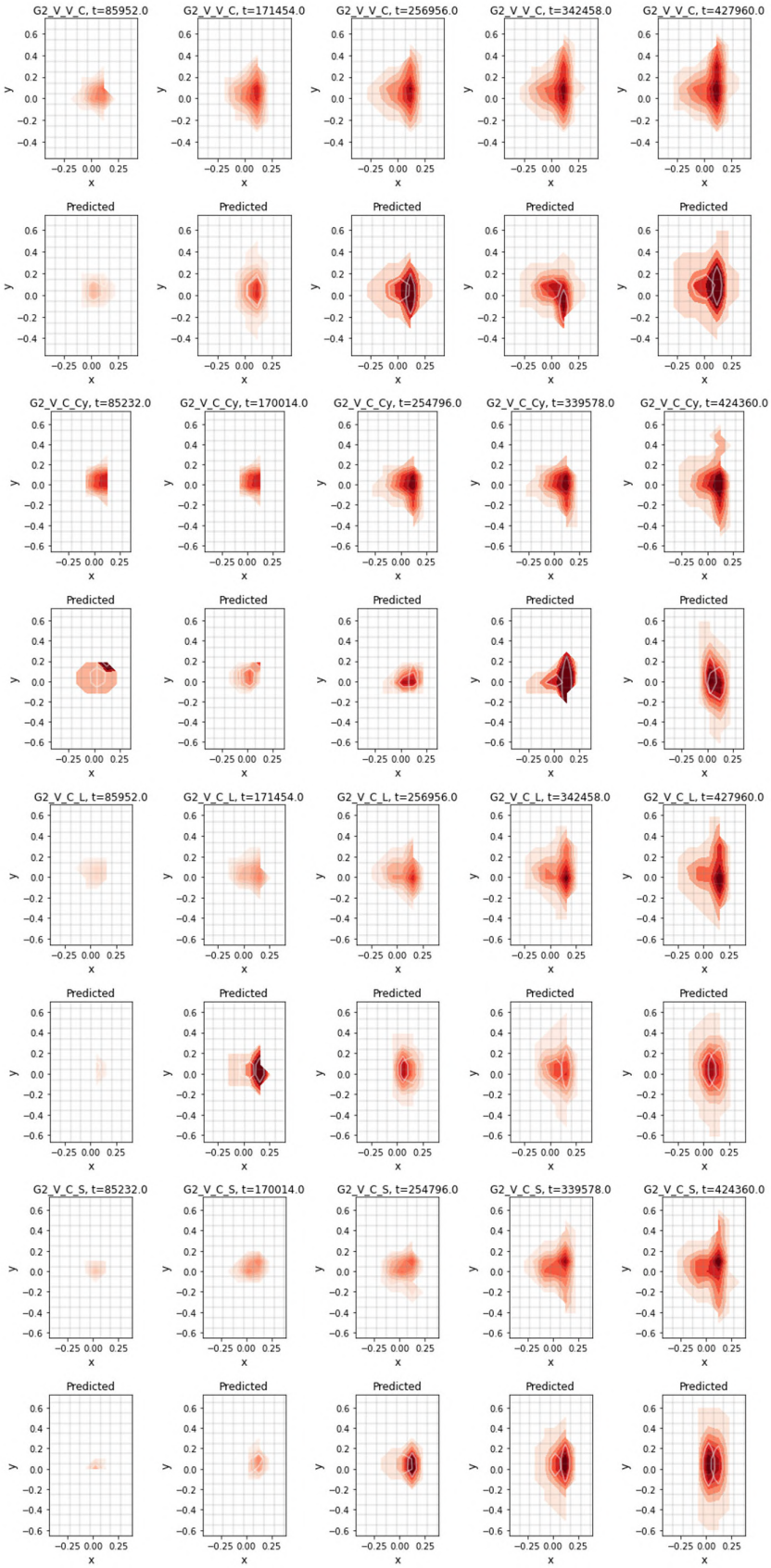

Figure 8. Two-cloud forecast (G2), Variable_Permeability case only, all simulations group.

References List

Asencio-Cortés G, Morales-Esteban A, Shang X, Martínez-Álvarez F (2018), Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure. Computers and Geosciences, 115, 198-210

Atkinson GM, Eaton DW, Igonin N (2020), Developments in understanding seismicity triggered by hydraulic fracturing. Nature Reviews, 1, 264-277

Bachman CE, Wiemer S, Goertz-Allmann BP, Woessner J (2012), Influence of pore-pressure on the event-size distribution of induced earthquakes. Geophys. Res. Lett., 39, L09302

Bergen KJ, Johnson PA, de Hoop MV, Beroza GC (2019), Machine learning for data-driven discovery in solid Earth geoscience. Science, 363,eaau0323

Beroza GC, Segou M, Mousavi SM (2021), Machine learning and earthquake forecasting – next steps. Nature Comm., 12, 4761

Broccardo M, Mignan A, Wiemer S, Stojadinovic B, Giardini D (2017), Hierarchical Bayesian modeling of fluid-induced seismicity. Geophys. Res. Lett., 44, 11,357-11,367

Broccardo M, Mignan A, Grigoli F, Karvounis D, Rinaldi AP et al. (2020), Induced seismicity risk analysis of the hydraulic stimulation of a geothermal well on Geldinganes, Iceland. Nat. Hazards Earth Syst. Sci., 20, 1573-1593

Chen H-J, Chen C-C, Ouillon G, Sornette D (2021), A paradigm for developing earthquake probability forecasts based on geoelectric data. Eur. Phys. J. Special Topics, 230, 381-407

DeVries PMR, Viégas F, Wattenberg M, Meade BJ (2018), Deep learning of aftershock patterns following large earthquakes. Nature, 560, 632-634

Dinske C, Shapiro SA (2013), Seismotectonic state of reservoirs inferred from magnitude distributions of fluid-induced seismicity. J. Seismol., 17, 13-25

Feng Y, Mignan A, Sornette D, Gao K (2022), Investigating injection pressure as a predictor to enhance real-time forecasting of fluid-induced seismicity: A Bayesian model comparison. Preprint available at: https://www.researchgate.net/publication/363289389_Investigating_injection_pressure_as_a_predictor_to_enhance_real-time_forecasting_of_fluid-induced_seismicity_A_Bayesian_model_comparison

Häring MO, Schanz U, Ladner F, Dyer BC (2008), Characterisation oft he Basel 1 enhanced geothermal system. Geothermics, 37, 469-495

He M, Li Q, Li X (2020), Injection-Induced Seismic Risk Management Using Machine Learning Methodology – A Perspective Study. Front. Earth Sci., 8, 227

Herrmann M, Kraft T, Tormann T, Scarabello L, Wiemer S (2019), A Consistent High-Resolution Catalog of Induced Seismicity in Basel Based on Matched Filter Detection and Tailored Post-Processing. J. Geophys. Res.: Solid Earth, 124, 8449–8477

Hicks SP, Goes S, Whittaker AC, Stafford PJ (2021), Multivariate statistical appraisal of regional susceptibility to induced seismicity: Application to the Permian Basin, SW United States. J. Geophys. Res. Solid Earth, 126, e2021JB022768

Hincks T, Aspinall W, Cooke R, Gernon T (2018), Oklahoma's induced seismicity strongly linked to wastewater injection depth. Science, 359, 1251-1255

Holtzman BK, Paté A, Paisley J, Waldhauser F, Repetto D (2018), Machine learning reveals cyclic changes in seismic source spectra in Geysers geothermal field. Sci. Adv., 4, eaao2929

Huang JP, Wang XA, Zhao Y, Xin C, Xiang H (2018), Large earthquake magnitude prediction in Taiwan based on deep learning neural network. Neural Net. World, 2, 149-160

Izadi G, Elsworth D (2015), The influence of thermal-hydraulic-mechanical- and chemical effects on the evolution of permeability, seismicity and heat production in geothermal reservoirs. Geothermics, 53, 385-395

Kong Q, Trugman DT, Ross ZE, Bianco MJ, Meade BJ, Gerstoft P (2018), Machine Learning in Seismology: Turning Data into Insights. Seismol. Res. Lett., 90, 3-14

Kraft T, Deichmann N (2014), High-precision relocation and focal mechanism of the injection-induced seismicity at the Basel EGS. Geothermics, 52, 59-73

Johnson, P. A., Rouet-Leduc, B., Pyrak-Nolte, L. J., Beroza, G. C., et al. (2021). Laboratory earthquake forecasting: A machine learning competition. Proceedings of the National Academy of Sciences, 118 (5), e2011362118}, 10.1073/pnas.2011362118

Limbeck J, Bisdom K, Lanz F, Park T, Barbaro E et al. (2021), Using machine learning for model benchmarking and forecasting of depletion-induced seismicity in the Groningen gas field. Computational Geosciences, 25, 529-551

Martínez-Álvarez F, Reyes J, Morales-Esteban A, Rubio-Escudero C (2013), Determining the best set of seismicity indicators to predict earthquakes. Two case studies: Chile and the Iberian Peninsula. Knowledge-Based Systems, 50, 198-210

Mignan A (2016), Static behaviour of induced seismicity. Nonlin. Processes Geophys., 23, 107-113

Mignan A, Broccardo M, Wiemer S, Giardini D (2017), Induced seismicity closed-form traffic light system for actuarial decision-making during deep fluid injections. Sci. Rep., 7, 13607

Mignan A, Broccardo M (2019), One neuron versus deep learning in aftershock prediction. Nature, 574, E1-E3

Mignan A, Broccardo M (2020), Neural Network Applications in Earthquake Prediction (1994-2019): Meta-Analytic and Statistical Insights on Their Limitations. Seismol. Res. Lett., 91, 2330-2342

Mignan A, Broccardo M, Wang Z (2021), Comprehensive Survey of Seismic Hazard at Geothermal Sites by a Meta-Analysis of the Underground Feedback Activation Parameter a_{fb}. Energies, 14, 7998

Okoroafor ER, Smith CM, Ochie KI, Nwosu CJ, Gudmundsdottir H, Aljubran M(J) (2022), Machine learning in subsurface geothermal energy: Two decades in review. Geothermics, 102, 102401

Pawley S, Schultz R, Playter T, Corlett H, Shipman T, Lyster S, Hauck T (2018), The geological susceptibility of induced earthquakes in the Duvernay play. Geophys. Res. Lett., 45, 1786-1793

Qin Y, Chen T, Ma X, Chen X (2022), Forecasting induced seismicity in Oklahoma using machine learning methods. Sci. Rep., 12, 9319

Rinaldi AP, Passarelli L (2021), Deliverable 3.1 – Extensive comparison of existing forecasting induced seismicity models with existing datasets. DEEP deliverable, 16 pp.

Ross ZE, Meier M-A, Hauksson E (2018), P wave arrival picking and first-motion polarity determination with deep learning. J. Geophys. Res. Solid Earth, 123, 5120-5129

Rouet-Leduc B, Hulbert C, Lubbers N, Barros K, Humphreys CJ, Johnson PA (2017), Machine learning predicts laboratory earthquakes. Geophys. Res. Lett., 44, 9276-9282

Shaheen A, bin Waheed U, Fehler M, Sokol L, Hanafy S (2021), GroningenNet: Deep Learning for Low-Magnitude Earthquake Detection on a Multi-Level Sensor Network. Sensors, 21, 8080

Shapiro SA, Dinske C, Kummerow J (2007), Probability of a given-magnitude earthquake induced by a fluid injection. Geophys. Res. Lett., 34, L22314

Shi P, Grigoli F, Lanza F, Beroza GC, Scarabello L, Wiemer S (2022), MALMI: An Automated Earthquake Detection and Location Workflow Based on Machine Learning and Waveform Migration. Seismol. Res. Lett., 93, 2467-2483

Spada M, Tormann T, Wiemer S, Enescu B (2013), Generic dependence of the frequency-size distribution of earthquakes on depth and its relation to the strength profile of the crust. Geophys. Res. Lett., 40, 1-6

van der Elst NJ, Page MT, Weiser DA, Goebel THW, Hosseini SM (2016), Induced earthquake magnitudes are as large as (statistically) expected. J. Geophys. Res. Solid Earth, 121, 4575-4590

Villiger L, Gischig VS, Doetsch J, Krietsch H, Dutler NO et al. (2020), Influence of reservoir geology on seismic response during decameter-scale hydraulic stimulations in crystalline rock. Solid Earth, 11, 627-655

Wozniakowska P, Eaton DW (2020), Machine learning-based analysis of geological susceptibility to induced seismicity in the Montney Formation, Canada. Geophys. Res. Let., 47, e2020GL089651

Yu P, Dempsey D, Calibugan A, Archer R (2021), Machine learning investigation of injection-seismicity in Rotokawa geothermal field. Proceed. 43rd New Zealand Geothermal Workshop, 23-35 Nov. 2021

Zhang X, Zhang J, Yuan C, Liu S, Chen Z, Li W (2020), Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method. Sci. Rep., 10, 1941

Liability Claim