SAPIENZA
UNIVERSITÀ DI ROMA

Final Project Report

Earthquake Physics and Machine Learning
Mathematical Sciences for Artificial Intelligence
Department of Mathematics
La Sapienza University of Rome

Machine learning to predict time to failure of lab earthquakes from acoustic emissions

Andrea Gentilini *Leo Vincenzo Petrarca*

Abstract. Earthquake prediction has had a long and controversial history. Despite tremendous effort and occasional glimmers of hope, the results have ultimately been disappointing, leading many to conclude that short-term earthquake prediction is at best infeasible and perhaps impossible. Charles Richter famously stated that "only fools, charlatans, and liars predict earthquakes". Nonetheless, the successful prediction of earthquakes remains one of the holy grails in Earth Sciences. Traditional methods rely on statistical information regarding recurrence intervals, which are not sufficiently accurate. However, with the advent of Machine Learning (ML), the earthquake science community now possesses a new suite of tools to tackle this enduring challenge. In 2017, Rouet-Leduc et al. [1] successfully predicted the time remaining before laboratory-induced earthquakes with a reasonable R^2 value, utilizing statistical features extracted from seismic acoustic data. This study employs ML techniques, which have demonstrated significant potential in recent works, to predict the time-to-failure of laboratory earthquakes using acoustic emission data. Specifically, we develop a novel predictive model trained on the La Sapienza Rock Mechanics Laboratory dataset, with the goal of surpassing the performance of two key benchmarks: the winning model ("The Zoo") from the 2019 Los Alamos National Laboratory Kaggle competition [2], and the convolutional neural network (CNN) model proposed by Rouet-Leduc from the same period. By systematically training and evaluating these models on the La Sapienza dataset, our approach achieves superior performance compared to the state-of-the-art at the time of the Kaggle challenge. This work advances the field by contributing to the broader discussion on the feasibility and limits of earthquake predictability through acoustic emissions.

1. Introduction

Earthquakes are among the most devastating and unpredictable natural phenomena, posing severe risks to human life and infrastructure. Although forecasting earthquakes in natural settings remains an unsolved challenge, laboratory experiments offer controlled conditions to study earthquake-like events and gain deeper insights into fault mechanics. In these experiments, seismic slip events are artificially induced under well-defined parameters to replicate the stick-slip behavior observed in natural faults. Such controlled settings serve as an effective testbed for developing and validating predictive models. Laboratory earthquakes are typically generated by shearing granular materials, such as glass beads or rock powders, within confined apparatuses that simulate fault zones. As shear stress gradually increases, the system undergoes intermittent cycles of slow deformation followed by sudden slip events, which closely mirror the stress accumulation and release cycles observed in natural seismic activity. The progression toward fault failure occurs in distinct phases: an initial quiescent period, followed by a phase of increasing stress accumulation, culminating in a catastrophic slip event. A critical quantity in this process is the time-to-failure (TTF), which refers to the time remaining before the catastrophic slip event occurs. Accurate estimation of TTF is central to both laboratory and real-world earthquake forecasting efforts. During these experiments, continuous acoustic emissions (AE), high-frequency sound waves generated by micro-fractures and internal deformations within the granular material, are recorded. These AE signals possess complex statistical and temporal structures that encode valuable information about the fault state and its progression toward failure. Recent advances have demonstrated that machine learning (ML) techniques can effectively extract predictive features from AE data, uncovering subtle patterns indicative of an impending slip event. For instance, Rouet-Leduc et al. [1] successfully employed ML methods to predict TTF by analyzing statistical features of acoustic signals, showcasing the ability of ML to reveal hidden precursors to fault failure in laboratory.

2. Data Characterization and Analysis

The objective of the Kaggle competition [2] was to predict the time-to-failure (TTF) of upcoming laboratory earthquakes by extracting informative features from portions of continuous acoustic emission (AE) data. The data were collected from experiments conducted in a double-direct shear configuration using a biaxial shear apparatus [3], [4], [5], operating under conditions close to the frictional stability transition. In this regime, laboratory earthquakes display complex metastable slip behavior [6], reflecting the intricate dynamics of fault systems. A schematic of the biaxial shear apparatus is shown in Figure 1.

The Kaggle dataset was provided in a two-channel format: one channel contained the acoustic emission signals recorded in arbitrary units, and the second contained the corresponding TTF values expressed in seconds. The training dataset consisted of a single continuous time segment of AE data that included multiple laboratory earthquake events, with TTF values recorded for each time point. This continuous representation enabled participants to analyze the evolution of AE signals leading up to catastrophic slip events and develop models capable of predicting TTF.

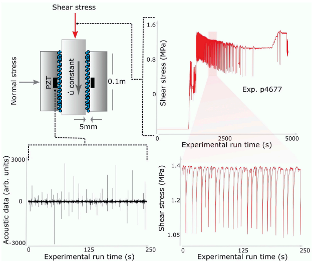


Figure 1: Experimental configuration and data collected, adopted from [3].

Instead, the data utilized in this study were collected at the La Sapienza Rock Mechanics Laboratory using a TiePie oscilloscope, renowned for its precision in capturing high-fidelity electric signals. The dataset was provided in a binary file format and consisted of recordings from six distinct channels: ‘tau’ (shear stress), ‘PU2’, ‘SL2’, ‘SR2’, ‘SR1’, and ‘SL1’. An example is shown in Figure 2. Each channel corresponds to specific sensor outputs, enabling a comprehensive, multi-dimensional analysis of seismic activity.

The oscilloscope was configured with a sampling frequency of $f_s = 6MHz\ (6 \times 10^6\, Hz)$, offering an exceptionally fine temporal resolution essential for detecting subtle precursors to seismic failures. The raw signals were stored in 32-bit floating-point format, which ensures a balance between data precision and storage efficiency. The dataset is organized into 200 chunks, with each chunk representing 1 second of recorded data, resulting in a total of 200 seconds of observations. Given the sampling frequency of $6MHz$, this corresponds to 1.2 billion data points per channel across the six channels, providing a rich and detailed dataset for high-resolution analysis of seismic processes.

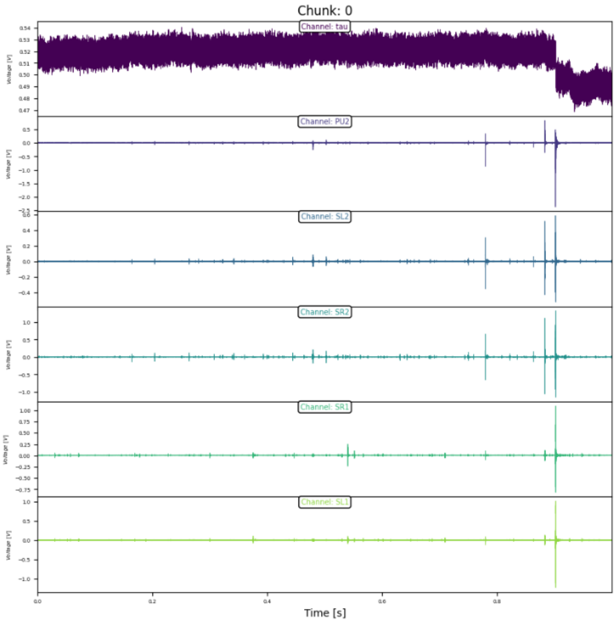


Figure 2: Plot of the six channels of collected data for the first chunk.

3. Data Pre-processing

The data pre-processing pipeline is a crucial step in preparing the raw experimental data for subsequent analysis and feature extraction. Our pre-processing steps are carefully designed to ensure the integrity and usability of the data, enabling the extraction of meaningful features for model training. Each data chunk is extracted from the corresponding binary files. For the construction of our acoustic data array, we needed to select one channel from the available options: ‘PU2’, ‘SL2’, ‘SR2’, ‘SR1’, and ‘SL1’. To identify the most informative channel, we analyzed the overall variance of each channel, under the assumption that the channel with the highest variance contains the most significant information regarding acoustic emissions. Based on this analysis, the channel “PU2” was selected as it exhibited the largest overall variance.

The acoustic data array was then constructed by concatenating the ‘PU2’ signal across all data chunks. The resulting array, obtained from the combined raw data, has a total length of 1.2 billion elements. We then projected all data points, originally stored as 32-bit floating-point values, into a 16-bit integer space. The transformation follows the equation:

$$AU = [\frac{V - V_{min}}{V_{max} - V_{min}} \times (N - 1)]$$

Where:

- $N = 2^{\#targetbits} = 2^{16} = 65536.$
- V_{max} = maximum voltage recorded in the acoustic signal
- V_{min} = minimum voltage recorded in the acoustic signal

Our next objective is to construct a TTF array using the shear stress data. To achieve this, we first downsample the shear stress signal to reduce its size and computational complexity, enabling efficient calculations while preserving the signal’s key characteristics. The target sampling rate for downsampling is set to $f_{s\_\text{red}} = 1000$ Hz (1 kHz). This reduction results in a downsampled array representing the complete shear stress data, with a total length of 200,000 elements. We perform the downsampling using the SciPy library, which provides optimized algorithms for signal processing, numerical integration, and interpolation, among other mathematical operations. The downsampling function utilized includes built-in anti-aliasing filters that prevent signal distortion by mitigating high-frequency artifacts during the resampling process. This ensures that the integrity of the shear stress signal is maintained after downsampling.

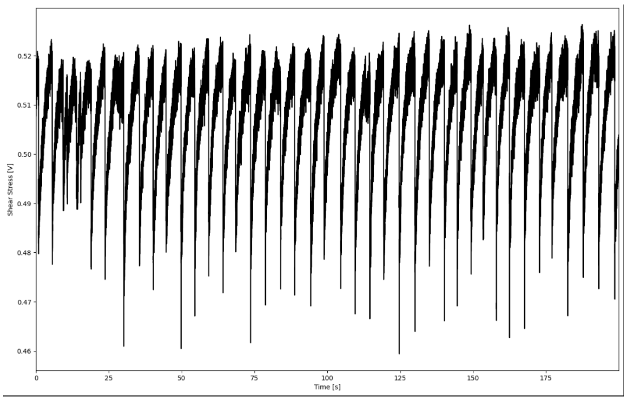


Figure 3: Plot of the raw shear stress.

In order to detect the stress drops, which are essential to build the time-to-failure array, we first need to smooth the shear stress signal using a moving average applied to the downsampled shear stress. The moving average acts as a low-pass filter, reducing high-frequency noise and enabling the identification of significant stress variations more clearly.

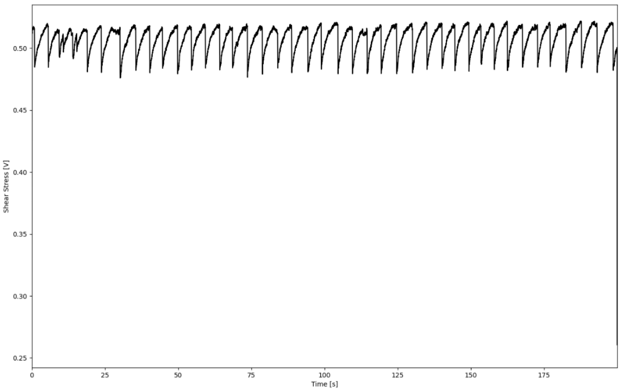


Figure 4: Plot of the denoised shear stress.

Stress drops are typically associated with sudden changes in the stress over time. To capture these variations, we utilize the derivative of the signal. Specifically, we calculate the incremental ratio by considering a fixed step size of step_size $= 30$, which represents the number of points between the values used for computing the difference. This approach is adopted for two main reasons: first, point-by-point computation may introduce excessive noise; second, it helps to avoid detecting multiple peaks corresponding to the same earthquake event. To isolate significant changes in the array, we set a threshold based on the standard deviation of the surrounding values. Only those changes exceeding this threshold are retained, ensuring that we identify meaningful stress drops while filtering out minor fluctuations.

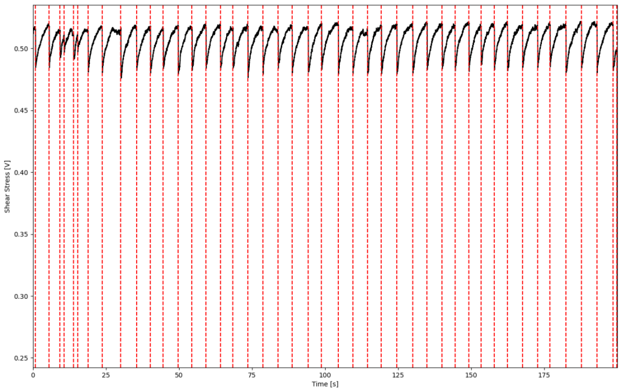


Figure 5: Plot of detected peaks corresponding to stress drops in the denoised shear stress signal.

In total, there are 43 earthquakes.

Our objective is to construct the TTF array by leveraging the detected stress drops in the shear stress signal. The time-to-failure at each specific index is defined as the time remaining until the next significant stress drop (earthquake). The shear stress signal exhibits distinct drops corresponding to earthquake-like events, marking the critical moments of fault failure. By systematically identifying these drops within the stress signal, we compute the time difference between the current position in the data and the closest subsequent event. This time difference, scaled by the inverse of the sampling frequency, allows us to obtain the TTF in seconds. The resulting TTF array has the same shape as the denoised shear stress signal, that is, $200,000$ elements. For the final values of the signal, where no subsequent stress drop (earthquake) is observed, the TTF is set to infinity.

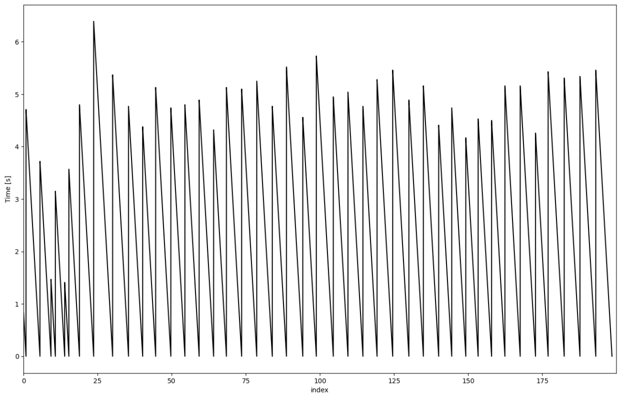


Figure 6: Plot of the time to failure.

We now aim to upsample TTF array back to $6MHz$ to ensure that we have the full resolution of the data available for subsequent analysis. This decision is motivated by the fact that, in later steps, we will evaluate the model's performance using reduced dataset (less data). For this reason, it is essential to start with the complete, high-resolution dataset to test the model's full potential. Upsampling usually introduces interpolation errors, which could compromise the quality of the data. However, in this case, we are fortunate: we do not need to interpolate the TTF values because the evolution of the time-to-failure is inherently time-dependent and continuous. Conceptually, upsampling in this context is equivalent to expanding the vector through time, where the values are already known. To perform the upsampling, we want for each point in the downsampled array 6,000 new points, as the relationship between the original sampling frequency (6 MHz) and the downsampled frequency (1 kHz) is given by:

$$\frac{6\,\text{MHz}}{1\,\text{kHz}} = 6,000.$$

To achieve this upsampling, linear interpolation is applied to create the new points at a higher sampling resolution, ensuring a smooth transition between consecutive elements of the downsampled TTF array. Each pair of consecutive values in the downsampled TTF array is interpolated with evenly spaced increments, effectively reconstructing the signal at a finer temporal scale.

Additionally, we address edge cases by removing the infinite values from the time-to-failure array. Correspondingly, we ensure that the matching last elements are also removed from the acoustic data array. This step is critical to maintaining consistency, as we require aligned time-to-failure values and acoustic emission signals to construct the training dataset for the machine learning model.

4. Dataset Creation and Feature Engineering

The input data consists of the time-to-failure and acoustic data arrays, both of length $1,191,240,001$. We begin by determining the appropriate number of complete earthquake events to include in our training set, allocating approximately 80% of the data for training and the remaining 20% for testing. The training size is carefully chosen to ensure cuts align with full earthquake events. To handle and pre-process the data, we utilize the Pandas library to construct and manage data frames for both the training and testing sets. To ensure reproducibility of our results, a fixed random seed is established.

The full signal is divided into segments, each of size $150,000$ points (equivalent to 0.025 sec at a sampling frequency of 6 MHz). For each segment, selected features are calculated, and the target value is defined as the time-to-failure at the end of the segment. The resulting segments and extracted features are organized into Pandas data frames for efficient manipulation. While a vast number of features can be derived from the acoustic data, we focus on a selection of 30 commonly used features. To identify the most relevant features for model learning, we perform feature optimization, limiting the subset size to 5 features to balance complexity and performance.

We conduct an iterative feature selection process, in which the model is trained multiple times using various subsets of these candidate features. This process reveals that the optimal model performance is achieved when trained on the following four features:

1. Number of peaks in the signal.
2. 20th percentile of the signal's standard deviation over a sliding window.
3. Mean (over time) of the 4th Mel-Frequency Cepstral Coefficient (MFCC).
4. Mean (over time) of the 18th Mel-Frequency Cepstral Coefficient (MFCC).

Regarding the dataframes structure, for each segment, the selected features are extracted and stored in data frames corresponding to both the training and testing sets. Each segment generates a single row in the resulting data frame, with the following six columns:

1. Peak count.
2. Rolling standard deviation (20th percentile).
3. Mean of the 4th MFCC coefficient.
4. Mean of the 18th MFCC coefficient.
5. Start index of the segment (relative to the entire dataset).
6. Time-to-failure at the end of the segment.

This structured approach to dataset creation and feature engineering ensures a consistent and reproducible framework for feature extraction, dataset preparation, and model evaluation.

5. Training

The model selected for training is a LightGBM regressor, an ensemble method in the boosting family [7]. We use 3-fold cross-validation to evaluate the model's performance during training. The dataset is split into approximately 80% for training and 20% for testing, ensuring that the splits align with complete earthquake events.

We perform an exhaustive search over a predefined hyperparameter space to optimize the model's performance. The selected hyperparameters include the number of boosting iterations, learning rate, maximum tree depth, and minimum data per leaf. Early stopping is applied with a patience of 50 iterations to prevent overfitting, based on the validation loss.

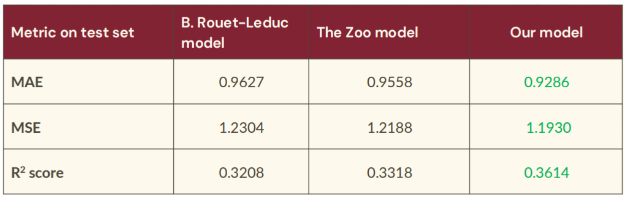
The final model is evaluated on the independent test set to estimate its generalization performance. The performance is quantified using the following metrics:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R^2

6. Results

The results demonstrate the effectiveness of our proposed model in predicting the time-to-failure variable. The performance of the model is evaluated using three standard metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the coefficient of determination R^2. These metrics provide a quantitative assessment of the accuracy and reliability of the predictions.

We compare the performance of our model against two state-of-the-art models available at the time of the Kaggle competition: the B. Rouet-Leduc model and the Zoo model. The results are summarized in Figure 7 , where our model achieves better performance across all three metrics.



Metric on test set	B. Rouet-Leduc model	The Zoo model	Our model
MAE	0.9627	0.9558	0.9286
MSE	1.2304	1.2188	1.1930
R^2 score	0.3208	0.3318	0.3614

Figure 7: Comparison of our model's results against the Zoo model and the B.Rouet-Leduc model.

The improvement in performance is further validated by visually comparing our model's predictions with the ground truth and the Zoo model's predictions. As shown in Figure 8, our predictions align more closely with the true values, particularly in capturing the key variations that lead to failure events.

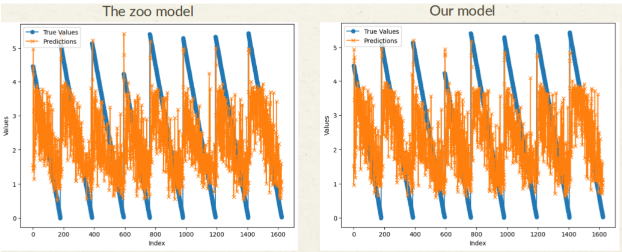
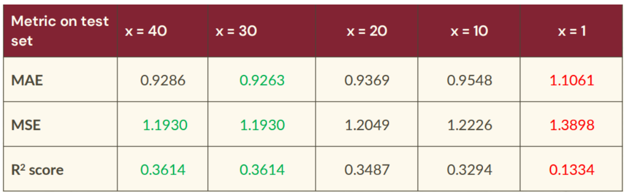


Figure 8: Comparison of our model's predictions with ground truth and the Zoo model.

To evaluate the efficiency of the model while maintaining its performance on the testing metrics, we aim to determine the minimum amount of data necessary to achieve comparable results. A conventional approach for resource reduction involves downsampling the original input signal, such as Time-to-Failure (TTF) and acoustic data. However, in this study, we explore an alternative strategy: reducing the size of the training dataset by selectively removing specific segments. The key question we address is whether segment removal can effectively minimize the training data volume without compromising the model's accuracy or generalization performance. For this reason, we analyze the impact of two different strategies for reducing the dataset size: introducing holes in the dataset (sampling at regular intervals, i.e. selecting x samples every y samples) and cutting the last portion of the dataset. For both approaches, the total amount of data analyzed are the same: 100%, 75%, 50%, 25%, and 2.5%. Results are summarized in Figure 9 and Figure 10. In Figure 9, we analyze the approach of selecting x samples every y samples, with $y = 40$.

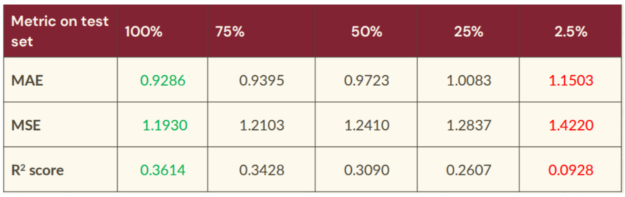
When introducing holes in the dataset, performance remain stable (and get even better) up to $\frac{3}{4}$ of data, with a mean absolute error (MAE) of 0.9263, a mean squared error (MSE) of 1.1930, and an R^2 of 0.3614. Beyond this point, performance worsen, particularly for $\frac{1}{40}$ of data, where the MAE increase to 1.1061, the MSE to 1.3898, and the R^2 drops to 0.1334. In contrast, reducing the dataset by cutting its final portion results in a steady decline in performance as the dataset size decreases. At 75%, the MAE increased to 0.9395 and the MSE to 1.2103, while R^2 dropped to 0.3428. Performance get significantly worse at 2.5%, with the MAE reaching 1.1503, the MSE 1.4220, and R^2 dropping to 0.0928.

The results indicate that introducing holes in the dataset preserves performance better than reducing its size by cutting. This behavior can be seen by comparing the results at equal levels of data reduction.



Metric on test set	x = 40	x = 30	x = 20	x = 10	x = 1
MAE	0.9286	0.9263	0.9369	0.9548	1.1061
MSE	1.1930	1.1930	1.2049	1.2226	1.3898
R² score	0.3614	0.3614	0.3487	0.3294	0.1334

Figure 9: Model performance with holes in the dataset at varying sampling intervals.



Metric on test set	100%	75%	50%	25%	2.5%
MAE	0.9286	0.9395	0.9723	1.0083	1.1503
MSE	1.1930	1.2103	1.2410	1.2837	1.4220
R² score	0.3614	0.3428	0.3090	0.2607	0.0928

Figure 10: Model performance with progressively smaller dataset sizes.

7. Conclusions

In this work, we demonstrated the application of machine learning techniques to predict the time-to-failure of laboratory-induced earthquakes using acoustic emission data. Our model, trained on the La Sapienza Rock Mechanics Laboratory dataset, outperformed state-of-the-art benchmarks as of 2019, including the B. Rouet-Leduc model and the winning model from the 2019 Los Alamos National Laboratory Kaggle competition. By implementing a rigorous preprocessing pipeline and systematic feature engineering, we successfully extracted informative features from the acoustic data, achieving strong predictive performance.

We analyzed the impact of dataset reduction using two strategies: introducing holes in the data and removing the final portion of the dataset. The results demonstrated that introducing holes preserves predictive performance significantly better than reducing the dataset size, emphasizing the critical role of maintaining temporal coverage when analyzing time-dependent signals.

References

[1] B. Rouet-Leduc, C. Hulbert, N. Lubbers, K. Barros,C.J Humphreys, P.A Johnson *Machine learning predicts laboratory earthquakes*, Geophysical Research Letters (2017).

[2] P.A. Johnson, B. Rouet-Leduc, L.J. Pyrak-Nolte, G.C. Beroza, C.J. Marone, C. Hulbert, A. Howard, P. Singer, D. Gordeev, D. Karaflos, C.J. Levinson, P. Pfeiffer, K.M. Puk, W. Reade, *Laboratory earthquake forecasting: A machine learning competition*, Proc. Natl. Acad. Sci. U.S.A. 118 (5) e2011362118, https://doi.org/10.1073/pnas.2011362118 (2021).

[3] P. A. Johnson et al., *Acoustic emission and microslip precursors to stick-slip failure in sheared granular material* Geophys. Res. Lett. 40, 5627–5631 (2013)

[4] J. Dieterich, G. Conrad, *Effect of humidity on time- and velocity-dependent friction in rocks*, J. Geophys. Res. 89, 4196–4202 (1984)

[5] M. M. Scuderi, C. Marone, E. Tinti, G. Di Stefano, C. Collettini, *Precursory changes in seismic velocity for the spectrum of earthquake failure modes*, Nat. Geosci. 9, 695,2016.

[6] J. R. Leeman, C. Marone, D. M. Saffer, *Frictional mechanics of slow earthquakes*, J. Geophys. Res. Solid Earth 123, 7931–7949 (2018)

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, *LightGBM: A highly efficient gradient boosting decision tree*, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.