Supporting Information for: Machine Learning predicts laboratory earthquakes

Bertrand Rouet-Leduc[1,2], Claudia Hulbert[1], Nicholas Lubbers[1,3], Kipton Barros[1], Colin Humphreys[2], Paul A. Johnson[4]

[1]Los Alamos National Laboratory, Theoretical Division and CNLS, Los Alamos, New Mexico.
[2]University of Cambridge, Department of Materials Science and Metallurgy, Cambridge CB3 0FS, UK.
[3]Boston University, Department of Physics, Boston Massachusetts.
[4]Los Alamos National Laboratory, Geophysics Group, Los Alamos, New Mexico.

Introduction

The data used are continuous seismic data collected from a laboratory shear experiment. For the training procedure, the recorded shear stress is used to denote where slip events take place. No pre-processing of these data files is used but the data are processed for the machine learning in the manner described below. Seismic data are collected at 330KHz. Only the seismic data are used for the machine learning testing, and only a sequence that the machine learning procedure has not seen before. There are no known imperfections in the data files.

Text S1: Experimental setup

The experimental setup to generate laboratory quakes (Fig. S1) has been discussed extensively *Marone* [1998]; *Niemeijer et al.* [2010]; *Scuderi et al.* [2014]. A three-block assembly with two gouge layers is placed in a bi-axial stress configuration. Two 5mm-thick fault gouge layers are placed between the three blocks, which are held in place by a fixed normal load. The gouge material is comprised of Class IV beads with diameter 105-149 μm. The central block is sheared at constant displacement rate. The two data streams recorded for our purposes here are the shear stress and the acoustic signal. At some time while the gouge material is in a critical shear stress regime, the shear stress abruptly drops, indicating gouge failure. These large drops in shear stress are laboratory quakes (Fig. S1). As applied load progressively increases, the inter-event time (recurrence) of laboratory earthquakes progressively decreases. At smaller applied loads the slips become aperiodic [Johnson et al., 2013]. In all cases, the rate of impulsive precursors accelerates as failure is approached [Johnson et al., 2013]. The acoustic particle acceleration $\ddot{u}$ is measured on the central block and can be readily converted to dynamic strain ϵ used in the ML analysis:

$$\epsilon = \frac{\dot{u}}{c}, \text{ where } \dot{u} = \frac{\ddot{u}}{\omega}, \text{ and } \omega = 2\pi f$$

with $c \approx 700$m/s the average measured wave speed in the granular material, and $f \approx 40.3$kHz. The sampling rate of the acoustic data is 330kHz. We are band-limited by the accelerometer (the frequency response is poor above about 50kHz). Therefore we select one of the system mechanical resonances within this band occurring at 44 kHz. Using this peak we improve our signal to noise ratio. The 'noise' is of very different character when the piston is stopped, and reflects primarily the mechanical resonances of the system. In short, we are certain the signal we analyze is the acoustic signal emanating from the fault, and the electromagnetic and system noise play no role in the predictions.

Corresponding author: Bertrand Rouet-Leduc, bertrandrl@lanl.gov

Text S2: Random Forest overview

Machine learning offers a variety of algorithms suitable for modeling the relationship between an input data (here characterized by features derived from a time window of the acoustic emissions signal) and corresponding output label (here the time remaining before the next failure event, derived from the shear stress signal). We applied a number of different ML algorithms to this problem; here we report results with random forests only, as they led to the best performance, and retain the link to the physics of the experiment. We provide a brief overview of the RF algorithm [Breiman, 2001] used in this study. Details are presented in the Random Forest details and Statistical features sections below.

A trained RF model predicts the time remaining before the next failure, from the features of the input time window. The RF is an ensemble method that makes its prediction by averaging over a set of simple decision trees. The trees are stochastically generated. Although the prediction of an individual decision tree may be quite inaccurate, the errors tend to cancel between trees, and the averaged RF prediction can be remarkably accurate.

A single decision tree (pictured in Fig. **1C**) operates as follows. To predict the time to failure for a data point extracted from the time series window, we begin at the root node and work toward the leaves. Each internal node encountered represents a binary decision (a split) on a single feature of the data point. For example, the decision at the root node could be “Is the variance during this time window greater than a given threshold c, for instance c=0.5”? If the answer is "yes", one continues up the left branch. Otherwise, one goes up the right branch. This sequence of decisions continues until a leaf node at the top of the tree has been reached (denoted by open circles in Fig. 1). Each leaf node contains a possible prediction of the tree, i.e. a predicted time remaining before failure (in seconds) for this particular data point. Note that each leaf node corresponds to a specific sequence of yes/no decisions on the features derived from the data point. Thus, a tree represents a map from inputs (acoustic emission features) to output labels (time remaining before failure prediction).

We build our dataset by computing statistical features from local time windows of the acoustic emission signal. Each data point (i.e. local set of statistical features) is associated with a time to failure, which we calculate from the shear stress signal. We then take our entire collection of data points and split it into two disjoint parts: training data and testing data. The training data is used to generate the RF model. The testing data is used to evaluate the performance of the RF model; this constitutes a fair measure of the RF performance because the testing data is independent from the training process (i.e. out of sample performance). It is very important to ensure that testing data does not leak into the training process.

To build the RF, we stochastically generate each decision tree from the training data as follows. The tree is given by a bootstrap resampling of the training data, which induces variation between the trees and mitigates the effect of outliers on the forest. Again, we begin at the root node of the tree and work toward the leaves. To generate each node, we formulate a "yes"/"no" decision (corresponding to a split into left/right branches) operating on the data available at the current node. At each node, we select a random subset of 40 percent of the available features. From the selected features, we construct the decision that best predicts the time to failure. This corresponds to selecting the split that partitions the data available at the current node j into two groups that are maximally dissimilar to each other with respect to time remaining before failure. Choosing a split corresponds to determine the feature X_m (among the 40% of feature selected) and the associated threshold c used to partition the data into these two subsets.

The criteria used for this purpose is the maximum reduction in (empirical) variance between the data available at the current node, and the two subsets of data partitioned by the split. More specifically, with j the current node of the tree, S_j the subset of data available at the current node, N_j the number of data points in S_j, $N_{j,\mathrm{L}}$ and $N_{j,\mathrm{R}}$ the number of data points in the left and right subsets $S_{j,\mathrm{L}}$ and $S_{j,\mathrm{R}}$ generated by the split, the criterion for a possible split s writes:

$$\Delta\mathrm{Var}(s, j) = \mathrm{Var}(S_j) - \frac{N_{j,L}}{N_j}\mathrm{Var}(S_{j,L}) - \frac{N_{j,R}}{N_j}\mathrm{Var}(S_{j,R}) \tag{1}$$

The split selected is the split that maximizes this variance reduction criterion. This criterion ensures that the data within each of the two subsets generated by the split is as homogeneous as possible, while these two subsets are as heterogeneous as possible one from another.

The data partitioned according to the split generates two branches. On each branch, the process repeats recursively, generating the entire tree structure. Decisions are generated until the number of data points at a node has reached a minimum size, at which point the tree constructs a leaf node. The tree assigns a prediction to each leaf node by taking the average time remaining before failure for the data that falls into that leaf. Following the nodes of a tree up to the final leaf gives the prediction of the tree for all the data that falls into this leaf. See the documentation on decision trees associated with *Pedregosa et al.* [2011] for more details and illustrations. The forest then averages the predictions of all the trees to make its final predictions. The random forest model is built once the sequence of splits has been identified (and therefore all the trees are constructed) on the training set. This model is then applied (i.e. by fixing the trees' structure) to a new dataset never analyzed before, the testing set. According to which leaves these new datapoints fall into, the model is able to generate predictions - without ever seeing the actual times to failure corresponding to this new dataset. The prediction associated with a data point falling in a given leaf corresponds to the average calculated in the training set for that particular leaf. These predictions on the testing set are shown in Figure 2 and Figure 1d.

Note the predictions never reach zero due to the discretization in time of the problem imposed by the moving window approach. In particular, we do not consider the time windows during which a failure occurred (neither for training nor for testing) because they would bias the prediction: at the moment failure takes place, all the statistical features are several orders of magnitude higher than the rest of the time. Moreover, we only care here about what happens leading up to failure, not at failure itself. This problem vanishes with smaller windows, at the cost of increased computation.

Text S3: Random Forest details

For this work, we used the scikit-learn implementation of the random forests *Pedregosa et al.* [2011], which implements the algorithm of *Breiman* [2001]. We compute regularization hyper-parameters by grid search based on a 3-fold cross-validation. The minimum number of samples to generate a split was 30. The minimum number of samples on a leaf was 30. The maximum number of features to consider when making a split was 40 (out of 100 features). The forest size was 1000 trees. The performance of the random forest is not sensitive to this choice of hyper-parameters: changing any hyperparameter by a factor of 2 typically affects the R2 performance by only a few percent.

To create a model that uncovers the physics of shear failure, we make predictions using moving time windows applied to the data. Each window is 1.8s, which is small compared to the time between fault gouge failures (8s on average). The offset between windows is 0.18s, meaning that consecutive time windows overlap by 90 percent. We characterize the acoustical signal in each window by a set of $\approx$ 50 statistical features (detailed in section “Statistical feature”). Each window is further split in two, and the features are computed for each sub-window to form one data point x_i, totaling $\approx$ 100 statistical features. We then label the data point x_i according to the time remaining until the next gouge failure, y_i, determined from the stress signal. The machine learning dataset is then $D_n = \{(x_i, y_i)_{i=1...n}\}$. Figure S4 shows the random forest learned from two features, the normalized second (variance) and fourth central moments (kurtosis). We use a 50/50 split of the full time series data into two contiguous pieces for use as training and testing data, respectively. Contiguity of these pieces is important to minimize contamination of the training data with information about the test data, which can arise due to temporal correlations in the full time series.

Text S4: Statistical features

We compute many statistical features within each time window for use by the random forest. We then perform feature selection through recursive feature elimination (RFE) to identify a subset of interesting variables to use in our algorithm. RFE has been shown to be among the best methods for feature selection with RF *Gregorutti et al.* [2017]. Once the ML model is built, random forests enable us to identify the most predictive features.

We provide the decision trees with many features that describe the physical state of the acoustic signal during a time window, and leave them to decide which ones are actually important. The features can be separated into three main categories:

• Signal distribution and energy: we use several higher order moments of the acoustic data to capture the evolution of the signal's energy. Within each time window we compute the signal's mean and higher moments, normalized (variance, skewness, kurtosis), and not normalized (7 features). The nth centered moment of f is given by: $\frac{1}{T}\int(t-\mu)^n f(t)\mathrm{d}t$, with $\mu=\int f(t)\mathrm{d}t$ the mean of f over the time window of length T.

• Precursors: when close to failure, the system enters a critical state and often emits strong bursts of acoustic emissions. We rely on different percentiles and thresholds to monitor this precursory activity during the considered time window, along with the minimum and maximum strain amplitude during a time window (18+10+2=30 features). We use the 1st to 9th and 91th to 99th percentiles, by increments of 1%. Our thresholds measure the fraction of time that the acoustic signal spends over a threshold value f_0, given by: $\frac{1}{T}\int\Theta(f(t)-f_0)\mathrm{d}t$, with $\Theta(x)$ the Heaviside step function ($\Theta(x)=0$ if $x<0$ and $\Theta(x)=1$ otherwise). This feature reflects previous analysis of the same experimental apparatus (*Johnson et al.* [2013]). We use the strain thresholds $f_0=10^{-9}$, 5×10^{-9}, 10^{-8}, 5×10^{-8}, 10^{-7}, and their opposite for negative strains.

• Time correlation features: we build several features based on Fourier analysis and auto-correlation functions (5 features). These features that we initially thought would be critical, are deemed to be of very low importance by the random forest models we built, and only very marginally improve the predictions. The Fourier transform-based features are the integral of the power spectrum over narrow frequency bands. We use the frequency bands $(a,b)=\{(19.65,20.65),(39.8,40.8),(80.1,81.1)\}$ in kHz, with the corresponding feature given by: $\int_a^b\hat{f}(\omega)\mathrm{d}\omega$, with $\hat{f}$ the Fourier transform: $\hat{f}(\omega)=\int \mathrm{f(t)e^{-2\pi i\omega t}dt}$. The autocorrelation is $\frac{\mathrm{E}[(\epsilon_t-\bar{\epsilon})(\epsilon_{t-h}-\bar{\epsilon})]}{(Var)(\epsilon_t)}$, with $\bar{\epsilon}$ the mean of the strain time series within a time window, and the time scale $h=\frac{1}{41.25\mathrm{kHz}}$. The correlation runs over the values of t within the considered time window. The partial autocorrelation function on the raw discrete time series ϵ_t is given by: $r(h)=\mathrm{Corr}(\epsilon_t,\epsilon_{t-h}|\epsilon_{t-1},...,\epsilon_{t-h+1})$, with the same timescale h.

We computed these features on both the AE signal ϵ and its first finite difference, given at a time t_i by: $\frac{\epsilon(t_{i+1})-\epsilon(t_i)}{t_{i+1}-t_i}\approx\frac{\mathrm{d}\epsilon}{\mathrm{d}t}(t_i)$. Forests that analyze only the derivative of the dynamic strain have a slight performance advantage, and so results reported here use only features from the derivative signal. As mentioned in the previous section, each time window of the acoustic signal is further split in two, and the features above are computed for each sub-window to create one data point. This gives the algorithm a notion of the short term evolution of the signal emitted by the gouge. The total number of features for each data point is therefore $(7+30+5)\times2=84$, constructed from a 1.8s piece of acoustic emission.

The importance of a feature is given by the mean decrease in impurity that it brought to the random forest model. For a regression problem, the impurity is the empirical variance of the data. Each split in the decision trees is made by minimizing the variance of the resulting two subsets of data. The sum of the variance of the two subsets is then lower than the variance of the data before the split. This decrease in variance is the decrease in 'impurity' (Eq. 1). The 'mean decrease impurity' (MDI) of a feature X_m is then the average drop in variance of the data for the splits (s, j) of all the decision trees T of the random forest that used this feature (*Louppe et al.* [2013]):

$$\text{Importance}(X_m) = \text{MDI}(X_m) = \frac{1}{N_\text{T}} \sum_{T} \sum_{(s,j)\in T: v(s,j)=X_m} \frac{N_j}{N} \Delta\text{Var}(s, j) \tag{2}$$

with j a node, (s, j) the split at node j, X_m the m^{th} variable (feature) of the input vector, $v(s, j)$ the variable used to make the split (s, j), N_T the number of trees in the random forest, N_j the number of data points left at node j, and N the total number of data points in the database. Note that this measure of 'impurity' is the variance of the data in feature space, not to be confused with the variance of the acoustic signal, which is a feature in itself.

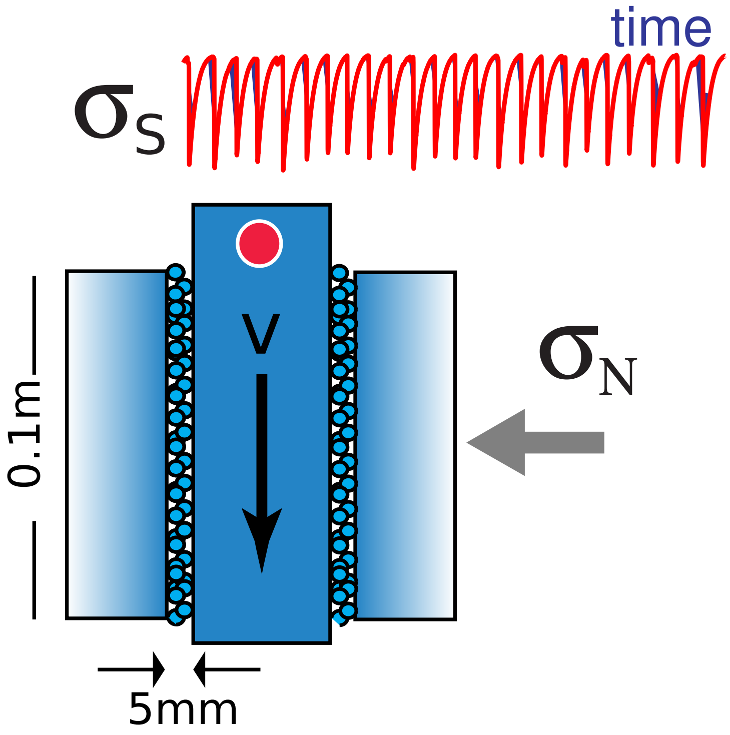

Figure 1. Bi-axial shearing device. A slider block with fault gouge layers on either side is loaded by a constant stress σ_N. The slider is driven downward at constant displacement rate v, inducing slide-slip behavior (as seen in the shear stress σ_S). An accelerometer continuously records the acoustic emission at a sampling rate exceeding 330kHz

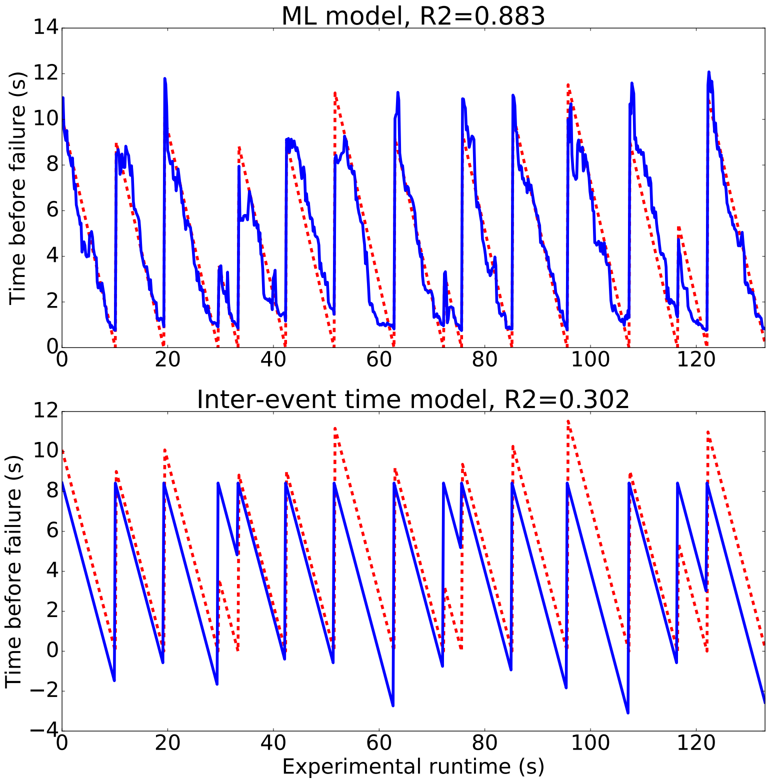

Figure 2. Top: the machine learning model. Bottom: a model that uses the average inter-event time of the training data to make predictions. Its performance is terrible: at an $R^2 = 0.3$ it is not much better than a straight line, as it completely misses the outliers. The dashed red curves are the real times remaining before failure, and the blue solid curves are the predictions from the models.

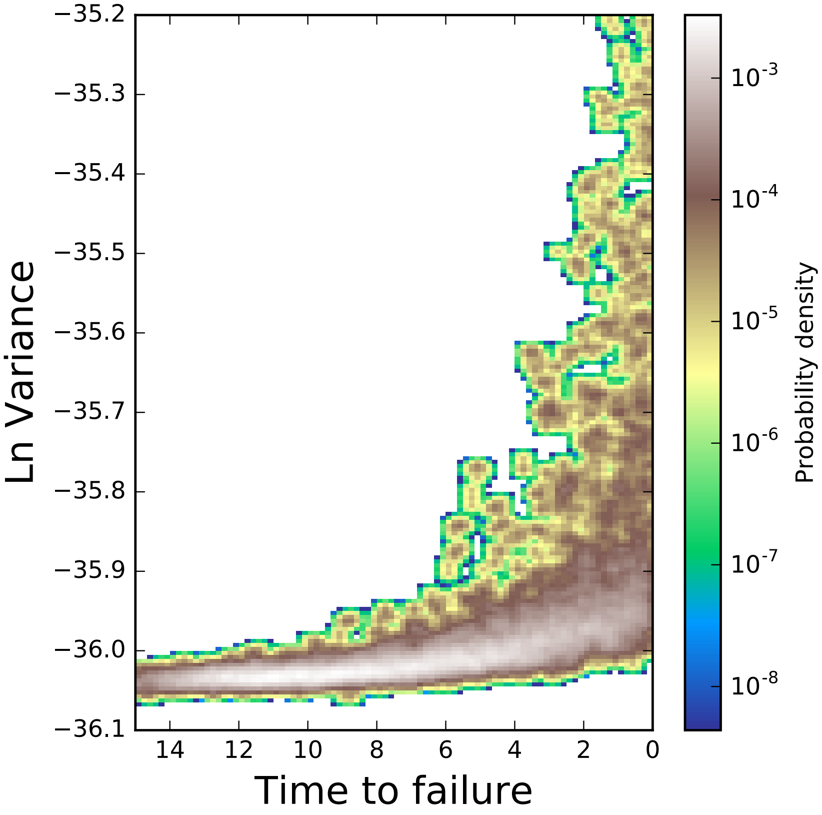

Figure 3. The distribution of data points across the variance and the time to failure, shown as a two dimensional histogram. As failure approaches, the amplitude of the variance grows and fluctuates, providing information about the time to failure.

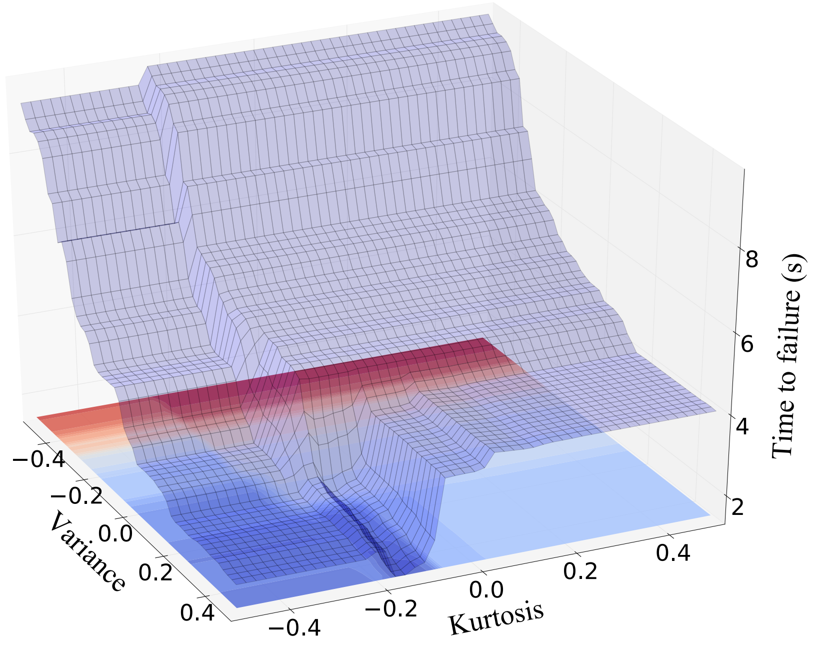

Figure 4. The time remaining before the next failure predicted by a RF model constructed from the second and fourth normalized central moments (variance and kurtosis) of the time signal.

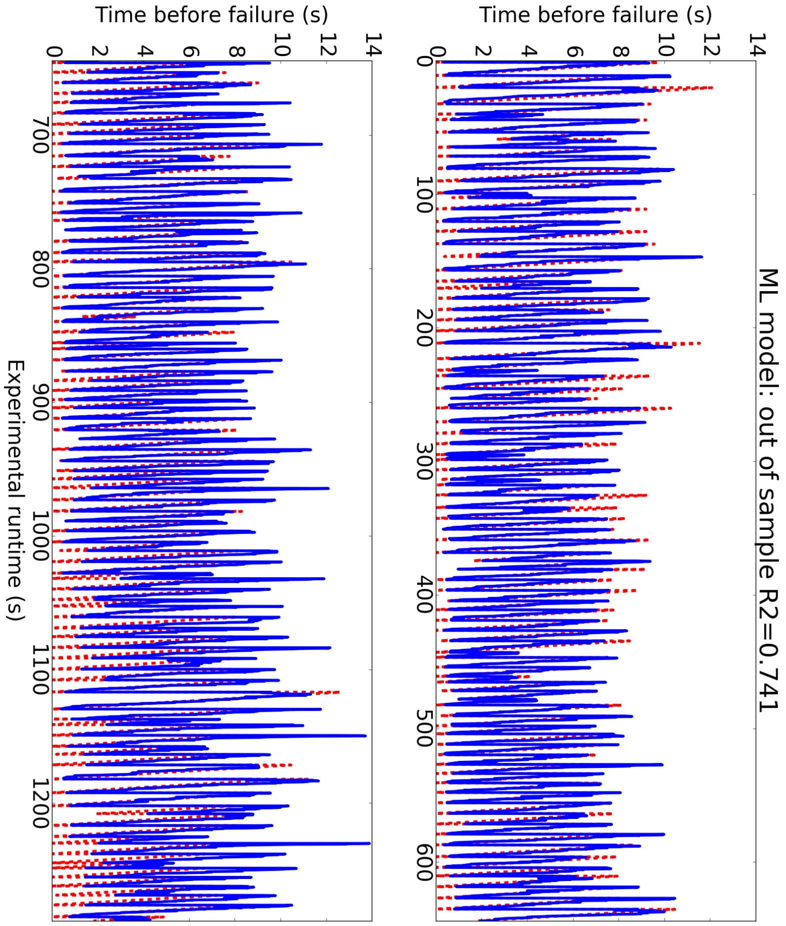

Figure 5. Results of training and testing at two different load levels. The dashed red curve is the real time remaining before failure, and the blue solid curve is the prediction from the ML model. Specifically, training was conducted at 5 MPa and testing took place at 8MPa applied load. The in sample R2 was 0.82 and the out of sample R2 was 0.741. ‘In sample’ refers to the training set, and ‘out of sample’ refers to the testing set. The accuracy is reduced but predictions nonetheless remain highly relevant to new experimental conditions, with clear outliers still being predicted correctly. Where there is a discontinuity of the dashed red curve, or where it does not reach zero, a problem was detected with the acoustic signal, and it was not considered for the analysis (this cleaning takes place before training or testing, when the statistical features are computed).

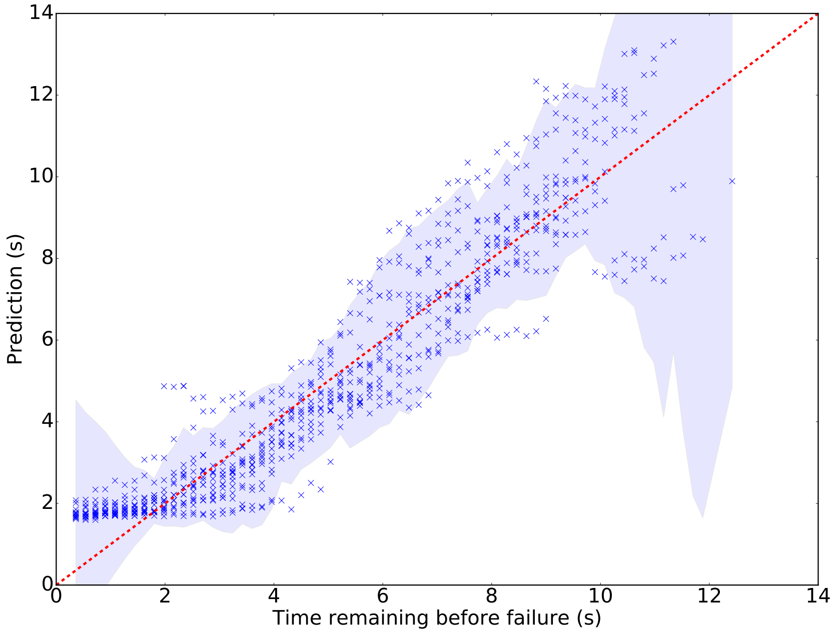

Figure 6. Predictions versus actual time remaining before failure, for all the cycles from the testing set. The data is the same as in Fig. 2 of the main text, but presented differently. Each blue cross is one prediction, and the blue shade is the average prediction ± two times the average absolute error of the model. As failure gets closer, the model gets more accurate, except below 2s, where the window size of 1.8s prevents the model from getting more precise. The true time remaining before failure (x axis) shows bins of 0.18s, again due to the window analysis. Perfect predictions would all be on the red dashed line.

References

Breiman, L. (2001), Random forests, *Machine learning*, *45*(1), 5–32.

Gregorutti, B., B. Michel, and P. Saint-Pierre (2017), Correlation and variable importance in random forests, *Statistics and Computing*, *27*(3), 659–678.

Johnson, P., B. Ferdowsi, B. Kaproth, M. Scuderi, M. Griffa, J. Carmeliet, R. Guyer, P.-Y. Le Bas, D. Trugman, and C. Marone (2013), Acoustic emission and microslip precursors to stick-slip failure in sheared granular material, *Geophysical Research Letters*, *40*(21), 5627–5631.

Louppe, G., L. Wehenkel, A. Sutera, and P. Geurts (2013), Understanding variable importances in forests of randomized trees, pp. 431–439.

Marone, C. (1998), Laboratory-derived friction laws and their application to seismic faulting, *Annual Review of Earth and Planetary Sciences*, *26*(1), 643–696.

Niemeijer, A., C. Marone, and D. Elsworth (2010), Frictional strength and strain weakening in simulated fault gouge: Competition between geometrical weakening and chemical strengthening, *Journal of Geophysical Research: Solid Earth*, *115*(B10), n/a–n/a, doi: 10.1029/2009JB000838, b10207.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011), Scikit-learn: Machine learning in python, *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Scuderi, M. M., B. M. Carpenter, and C. Marone (2014), Physicochemical processes of frictional healing: Effects of water on stick-slip stress drop and friction of granular fault gouge, *Journal of Geophysical Research: Solid Earth*, *119*(5), 4090–4105, doi: 10.1002/2013JB010641, 2013JB010641.