Attention Network Forecasts Time-to-Failure in Laboratory Shear Experiments

Hope Jasperson1, David C. Bolton2, Paul Johnson1, Robert Guyer3,4, Chris Marone5,6, and Maarten V. de Hoop7

1Rice University, EEPS, Houston, TX, USA, 2University of Texas Institute for Geophysics, Austin, TX, USA, 3Los Alamos National Laboratory, Geophysics Group, Los Alamos, NM, USA, 4Department of Physics, University of Nevada, Reno, NV, USA, 5Department of Geosciences, Pennsylvania State University, State College, PA, USA, 6Dipartimento Scienze della Terra, Sapienza Universitá di Roma, Rome, Italy, 7Rice University, CAAM, Houston, TX, USA

Abstract Rocks under stress deform by creep mechanisms that include formation and slip on small-scale internal cracks. Intragravel cracks and slip along grain contacts release energy as elastic waves termed acoustic emissions (AE). AEs are thought to contain predictive information that can be used for fault failure forecasting. Here, we present a method using unsupervised classification and an attention network to forecast labquakes using AE waveform features. Our data were generated in a laboratory setting using a biaxial shearing device with granular fault gouge intended to mimic the conditions of tectonic faults. Here, we analyzed the temporal evolution of AEs generated throughout several hundred laboratory earthquake cycles. We used a Conscience Self-Organizing Map (CSOM) to perform topologically ordered vector quantization based on waveform properties. The resulting map was used to interactively cluster AEs. We examined the clusters over time to identify those with predictive ability. Finally, we used a variety of LSTM and attention-based networks to test the predictive power of the AE clusters. By tracking cumulative waveform features over the seismic cycle, the network is able to forecast the time-to-failure (TTF) of lab earthquakes. Our results show that analyzing the data to isolate predictive signals and using a more sophisticated network architecture are key to robustly forecasting labquakes.

In the future, this method could be applied on tectonic faults to monitor earthquakes and augment early warning systems.

Plain Language Summary Earthquake forecasting is a daunting task, but advances in machine learning can help us toward this goal. In this study, we combine machine learning with expert knowledge about earthquake formation to forecast synthetic earthquakes made in a laboratory. Prior to an earthquake, rocks emit small bursts of energy as they are stressed, which are called acoustic emissions (AE). We identified all the AEs in the raw seismic data from our lab earthquakes and then used a machine-learning algorithm called a Conscience Self-Organizing Map to divide them into groups based on their characteristics. We examined the groups to determine which ones would be useful for earthquake forecasting. Finally, we used the data from the chosen groups along with another machine learning approach (LSTM and attention networks) to forecast the time of future labquakes. Our results show that earthquake forecasting is improved by processing seismic data to find predictive signals and by using machine-learning networks that are specifically made for time series data.

1. Introduction

Earthquake forecasting is an active topic in geoscience research and recent progress in this area has been rapid, thanks in part to developments in machine learning and the availability of laboratory seismic data containing large numbers of labquakes. Quite recently, many researchers have used a variety of machine learning methods in an attempt to forecast labquakes and have had moderate success (e.g., Corbi et al., 2019; Hulbert et al., 2019; Johnson et al., 2021; Rouet-Leduc et al., 2017; Tanyuk et al., 2019). In a recent Kaggle competition hosted by Los Alamos National Lab and the DOE Office of Science, the top scoring team used a combination of gradient-boosted trees, support vector regression, and a convolutional neural network to forecast time-to-failure (TTF) from raw laboratory seismic data for a final mean absolute error (MAE) of 2.26 s (Johnson et al., 2021). This network and others achieve respectable results, but their success is limited because their training procedures lack domain expert knowledge and generally fail to fully utilize the power
of recurrence. In this work, we prepare laboratory seismic data for a forecasting network by identifying predictive signals via catalog creation and clustering. By doing this, we remove noise and spurious signals that make training a machine learning network more difficult. We use these predictive signals to train a variety of recurrent networks and achieve a superior MAE of 1.04 s using an attention network. Attention is meant to mimic cognitive attention by enhancing the important aspects of the input data and muting the remaining aspects. Determining which aspects of the data are important is learned via model training data by gradient descent. Our results show that analyzing the data to isolate predictive signals and using a more sophisticated network architecture are key to robustly forecasting labquakes.

A primary motivation for this study comes from the engineering field of nondestructive testing (NDT). The goal of NDT is to continuously assess the health of a material or machine while it is in operation (Farrar & Worden, 2013). NDT has many practical applications, including identifying when machine parts need to be repaired and forecasting infrastructure failure. NDT techniques have been successfully applied to a wide range of materials, such as reinforced concrete (Calabrese et al., 2013), manufacturing tools (Yen et al., 2013), composite materials (Li et al., 2015), wood (Diakhate et al., 2017), and general structural health monitoring including structures (Farrar & Worden, 2013).

NDT acoustic emissions (AE) monitoring relies on continuously recording acoustic data as the material of interest is stressed. Acoustic data are recorded at least until the material reaches failure. In the lab, the failure process can be sped up by artificially subjecting the material to stress (e.g., Calabrese et al., 2013; de Oliveira & Marques, 2008; Diakhate et al., 2017; Huguet et al., 2002; Li et al., 2015). Once the material has reached failure, discrete acoustic emissions are isolated within the acoustic data. A discrete AE is a finite-duration elastic wave that is produced by the formation of small-scale internal cracks and slip along grain contacts. Thus, AEs provide a record of how the material responds to stress. Next, AEs can be clustered according to damage mechanism using an unsupervised clustering algorithm. This step groups AEs according to the type of crack or deformation by which they were produced. In some studies, transmitted light (e.g., Li et al., 2015) or scanning electron microscopy (e.g., Fallahi et al., 2016) is used to determine the true labels for each cluster. Finally, AE production throughout the failure cycle is analyzed in order to identify temporal patterns and make inferences about the remaining life (RUL) of the material. Some studies have gone further and used acoustic data combined with machine learning methods to forecast RUL with varying degrees of success (e.g., Elforjani & Shanbr, 2018; Farrar & Worden, 2013; Louis et al., 2020; Zheng et al., 2017).

In this study, we follow the example of NDT to develop a deep-learning based forecasting procedure trained and tested on laboratory-generated seismic data. Although the earthquake rupture process is generally different from the material failure studied in NDT, the underlying physics of AE production are thought to be the similar or the same (e.g., Lockner & Byerlee, 1977; Scholz, 1968). In biaxial shear experiments we analyze, grain-to-grain and grain-to-block displacements are the origin of the AE signals (Trugman et al., 2020). Previous studies on similar laboratory datasets sought to identify temporal trends and precursors in acoustic data via clustering (Bolton et al., 2019) and forecast TTF applying decision tree approaches using the continuous AE signal (Rouet-Leduc et al., 2017) and AE catalogs (Lubbers et al., 2018). Recent works monitor material properties more directly by using active-source acoustic data to predict labquakes using both decision tree and deep-learning approaches (Shokouhi et al., 2021; Shreedharan et al., 2021). These works demonstrate that the laboratory seismic signal contains predictive information. A similar approach applied to the phenomenon of “slow slip” in the laboratory shear experiments (Hulbert et al., 2019) and the Cascadia subduction zone, demonstrated that seismic signals in Earth are imprinted with information regarding fault slip rate and upcoming failure (Rouet-Leduc et al., 2019). Here, we apply a different ML approach and identify components of the seismic signal that carry predictive information through clustering and use machine learning to continuously forecast TTF and fault shear stress.

Our forecasting method consists of four steps: catalog creation, clustering, cluster selection, and forecasting via deep learning. In the first step, we create a catalog of AEs as a way to eliminate noise and focus on potentially useful waveforms. This can be thought of as segmentation and a kind of coarse clustering, such as that of Seydoux et al. (2020). Next, we perform a fine grained clustering procedure on the catalog to further categorize the AEs. We introduce a cluster selection process to identify useful clusters and eliminate clusters.
that do not contain predictive information. Finally, we use an agglomerate of recurrent neural networks to forecast TTF and shear stress at the fault using data from the previously selected waveform clusters.

The layout of this paper is as follows. In Section 3, we first test our proposed method on a synthetic benchmark data set. This data set is generated by the brittle-ductile friction model, also known as the “broom model” (Daub et al., 2011). We train an LSTM network on the benchmark data, which successfully forecasts time to failure. With the success of this test, the remainder of the paper is focused on applying this same forecasting procedure to laboratory seismic data, which is described in Section 4. In Section 5, we discuss the procedure used to prepare the AEs for the forecasting network. As in the NDT process, we further refine the catalog by clustering AEs according to waveform features. After analyzing the resulting clusters, we find that only two clusters appear to contain predictive information. In Section 6, we use these two clusters to train a collection of networks to forecast time to failure and shear stress. Finally, in Section 7, we discuss prior work and provide comparisons with our method. Additional comparisons to historical methods are provided in Supporting Information S1.

2. LSTM and Attention Networks

Our forecasting system is based around the use of recurrent networks, particularly LSTM (Hochreiter & Schmidhuber, 1997) and attention networks (Bahdanau et al., 2015). This type of network is well suited to time series forecasting because of its ability to store past information for future use. In Supporting Information S1, we establish that rate and state friction can be written as a recurrent network from first principles, making this a physics-informed approach. For training, we use a regressive seq2seq procedure (Sutskever et al., 2014) in which the input sequence consists of seismic features and the output sequence is the forecasted values for shear stress and TTF.

The LSTM network consists of several unidirectional LSTM cells stacked so that the hidden state from one cell becomes the input vector for the next cell (Figures S16 and S17). A final fully connected layer outputs the time-to-failure and shear stress at the current time step.

Attention networks are similar to LSTM networks, with the addition of a special layer that gives the network improved ability to observe past data trends. These networks consist of an encoder-decoder structure and generally use gated recurrent units (GRUs) rather than LSTM cells due to their faster training time (Figures S19 and S20). During the training process, the network learns to assign a relative importance to each data point, which is used as a weighting metric when making forecasts. For these networks, we use local self-attention where at time step \( t \), the network only looks at a fixed window of previous time steps, in which importances are calculated for each AE.

The input vectors for each network consist of cumulative waveform features. Part of our goal is to train each network only on AE waveforms that contain predictive information. We expect that eliminating nonpredictive signals and noise from the training data will increase network accuracy. Therefore, we choose not to use the raw trace or features calculated from sliding windows. We use cumulative features for training as opposed to the raw values to make it easier for the networks to learn from temporal patterns.

We used several common network modifications in order to speed up training and enhance generalization. We trained with mini-batches wherein the network sees multiple quakes at once rather than sequentially. Weight updates are performed after each mini-batch, and thus occur multiple times per epoch. This has the benefit of speeding up training and balancing the training loss between hypersensitivity (i.e., updating after each quake) and insensitivity (i.e., updating after a full epoch). Mini-batches are determined randomly at the beginning of each epoch and thus may contain quakes of very different recurrence intervals. To account for the variable recurrence interval, we pad shorter cycles with zeros to match the longest recurrence interval of the batch. The padding is not included in error calculations. We experimented with different mini-batch sizes to find one that was both fast and accurate.

Similarly, we use a predetermined sequence length to control the number of weight updates. Updating the network weights after each AE would be time-consuming and excessive, but waiting until the end of a cycle risks averaging out the data and preventing learning. Instead, we update after a predetermined number of AEs have been seen, and found that 300 was generally a good number (Supporting Information S1). After
each update, this count restarts. As with the other hyper-parameters, we experimented with different values to strike a balance between too few and too many AEs per update.

Finally, we used gradient clipping to reduce the maximum value of the weight gradients (Pascanu et al., 2013). Recurrent networks are prone to excessively large gradients and this can especially be a problem with unbalanced datasets such as the lab data, where short and long recurrence intervals are under-represented. Gradient clipping speeds up convergence by reducing unwanted fluctuations in the network weights. We experimented with several clipping thresholds to find one that was well suited to the data.

To test the universality of these networks, we train on all stress conditions and recurrence intervals simultaneously. The cycles are randomly shuffled at the beginning of each epoch so that each training batch contains a random selection of cycles. We train each network using the Adam optimizer (Kingma & Ba, 2017; Loshchilov & Hutter, 2019) and an adaptive learning rate. Loss during training is calculated using the smooth L1 loss function.

3. Forecasting Benchmark

We begin by testing our proposed forecasting method on a benchmark data set, specifically the Broom model of Daub et al. (2011). Our goals here are twofold: to test the capability of recurrent networks to learn patterns from metastable dynamical systems and to assess the robustness of the forecasting method. We perform these tests using a LSTM network. We do this because the LSTM is simpler and faster to train than the attention network. However, the LSTM network is also less powerful than the attention network (Bahdanau et al., 2015). If the LSTM successfully learns the benchmark data, then we assume that the attention network would work as well.

Next, we will briefly describe the results of the benchmark tests. The full results are in Supporting Information S1. We find that the LSTM network is clearly capable of forecasting failure from the Broom model data. In general, the network forecasts time to failure very accurately and does so consistently across all stress levels and recurrence intervals. With these tests, we show that LSTM networks have the ability to learn from irregularly sampled metastable seismic systems. The Broom model is a simple example of such a system, but the excellent network performance suggests that a similar method could be used on more complex data.

4. Laboratory Experiments and Previous Studies

With the success of the forecasting tests on the Broom model data, the next task is to test the network’s ability to learn a more complex system. For this task, we focus on laboratory generated seismic data. The laboratory data represents fault conditions on a single frictional patch on a tectonic fault with minimal noise, scattering, attenuation, etc. Furthermore, unlike seismogenic fault zones, the characteristics of laboratory earthquakes can be systematically modulated via the boundary conditions of the experiment (e.g., Leeman et al., 2016). Hence, hundreds of laboratory earthquakes can be produced within a single experiment that have similar or related properties (assuming the loading conditions are held constant). In addition, the laboratory data also provide high-resolution AE data from a network of acoustic sensors. The acoustic sensors are placed 22 mm from the edge of the fault zone, which gives a reasonable approximation of the dynamical system. These are data that represent frictional properties of granular materials and thus are more realistic than the Broom model but still are far from the true complexity in tectonic fault systems where a fault may comprise a very large number of frictional patches.

The laboratory datasets were created using a biaxial shearing system in a double-direct shear (DDS) configuration (e.g., Anthony & Marone, 2005; Bolton et al., 2020; Riviere et al., 2018). The DDS configuration consists of two granular fault zones positioned between three steel loading platens. The central block is driven at constant loading rate producing a series of stick-slip events (i.e., laboratory earthquakes). Adjacent to the DDS loading platens are two steel blocks containing an array of piezoelectric transducers (Figure S10). AE data are recorded continuously throughout the experiment at 4 MHz. Experiments were conducted with soda-lime glass beads due to their highly reproducible seismic and frictional properties (Anthony & Marone, 2005; Mair et al., 2002; Scuderi et al., 2014). In this work, we analyze data from an experiment where
the normal stress was systematically increased stepwise from 2 to 8 MPa in steps of 1 MPa, and subsequently
decreased back to 2 MPa (see Figure S11).

Many previous studies have worked to analyze the laboratory data and describe the underlying physics.
Riviére et al. (2018) documented changes in the Gutenberg-Richter b-value over lab seismic cycles. Bolton et al. (2020) examined how AE energy release is impacted by factors such as shear velocity, slip displacement, stress drop, etc. They found that acoustic variance is tied to the slip rate of the fault, which results in higher variance closer to failure, which is consistent with the results of Rouet-Leduc et al. (2017) and Rouet-Leduc et al. (2018) who first demonstrated lab earthquake prediction using the continuous AE signal. It was also observed in these works that AE energy is stronger at high stress than at low stress. In simulations applying DEM and FDEM (Gao et al., 2018; Ren et al., 2019, respectively), it was shown how the movement of individual gouge grains can be used to predict macroscopic friction.

Other studies have examined the role of AEs as precursors to failure. Bolton et al. (2019) clustered AEs in PC space and found a progression of clusters that indicate the slip stage of the fault. Shreedharan et al. (2020) concluded that changes in acoustic transmissivity (a variation on acoustic amplitude) are a precursor to fault failure and two additional works showed that such active source acoustic data can be used to predict labquakes, when available (Shokouhi et al., 2021; Shreedharan et al., 2021). Lubbers et al. (2018) found that there is an evolution of micro events that increase in density as failure is approached. Trugman et al. (2020) located AEs within the fault gouge and observed their spatio-temporal evolution over the seismic cycle. They found that each cycle is different and thus there was not a unifying common pattern.

In this study, we use acoustic data from two laboratory experiments (p4581 and p4583) for a total of nearly 400 stick-slip events. These events take place under a variety of stress conditions with shear stresses ranging from roughly 1–3.5 MPa. We identified events using the recorded shear stress and discarded events that took place during transitions between normal stress levels. We focus only on data from the interseismic period, which is defined as the minimum shear stress of the previous stick-slip event to the maximum shear stress of the current stick-slip event. The recurrence interval of the stick-slip cycles range from 5–25 s, with an average of 10 s (note this is averaged over multiple stress levels) (Figure S11).

In the laboratory, it is common for the main stick-slip event to be preceded by smaller events that have measurable stress drops (see Figure S11b). We refer to these small events that precede the main-stick slip event as foreshocks. These events present a particular challenge that was not present in the Broom data set. Generally, it is not clear-cut if foreshocks should be included as separate failure cycles. At high stress, foreshocks tend to have very small stress drops relative to the main slip event and the overall shear stress evolution is usually unchanged. In these cases, we include the foreshocks as part of the longer failure cycle. At low stress, foreshocks tend to be more impactful in that they have proportionally larger stress drops relative to the coseismic events and the shear stress enters a clear recovery phase as a result. In this case, we treat the foreshocks as separate failure cycles. We prepared a separate data set where foreshocks at high stress were also treated as separate cycles and found that it resulted in a clear decline in forecasting accuracy.

AEs were detected by the same process described in Lubbers et al. (2018), though an unsupervised machine learning method like that of Seydoux et al. (2020) could also be used. The detection process uses a thresholding procedure to scan through the continuous data and catalog events according to their peak amplitude (Riviére et al., 2018). The creation of an AE catalog is the first step in the process of reducing noise and identifying useful signals. This process is vital to producing accurate forecasts because it decreases the complexity of the data that the forecasting network must learn. The cataloging process is by no means perfect and some AEs are likely missed due to smaller events that may occur in close proximity to larger events. Furthermore, it is also possible that the magnitude of completeness increases closer to failure because of higher event rates and temporal clustering of larger events. However, the Broom tests showed that the LSTM network is robust in this situation. Figure S11d is a snapshot of the acoustic data with AEs denoted by red circles. The catalog for the two datasets combined contains about 8 million AEs and each stick-slip cycle consists of several thousand AEs. AEs that nucleate during the co-seismic slip phase are not used.
5. Feature Engineering and AE Clustering

The second step of the noise removal process is to cluster the AEs. Later, we will examine cluster evolution over time, which will be used to separate predictive and nonpredictive signals. This procedure is inspired by NDT studies in which clustering algorithms are used to provide information about AE production and importance. In these studies, clusters are thought to be associated with different micro-mechanical processes (i.e., damage mechanisms). In this sense, the clusters provide insights into the physical processes occurring inside the stressed material and helps to discriminate between important and unimportant signals. Here, we employ a clustering scheme to further remove noisy signals, which should map to one or more noise clusters, and identify AEs that may contain predictive information.

5.1. Handcrafted Features

For the damage mechanism clustering, we use a set of five features typically used in NDT studies: maximum amplitude, counts, duration, energy, and rise time (Figure 1a). The NDT literature shows that these five features are sufficient for AE clustering (Grosse et al., 2022). Counts is the number of positive peaks over the duration of the AE and rise time is the length of time until the maximum amplitude is reached. In addition, we use two frequency features, average frequency, and peak frequency, which brings the total number of features to seven. Though we use handcrafted features here, an unsupervised feature extraction method could also be used.

In order to calculate features such as energy, the start and end times of the AEs must be known. Start and end times were determined using an amplitude threshold as described in Godin et al. (2004). During the processing procedure, we found that a simple 10% threshold was inadequate due to the large spread of maximum amplitudes. As a result, we used a decreasing threshold scheme whereby the percentage used is decreased as the maximum amplitude increases. For very low amplitude events, the minimum amplitude is fixed to be slightly above the noise level. Despite these adjustments, amplitude thresholding is an imprecise tool, which may warrant further improvements in the future.

As with the cataloging procedure, this process wrongly eliminates some AEs. For example, amplitude thresholding eliminates AEs that overlap in time or have poorly defined start or end points. In these cases, wrongful eliminations would likely happen even if an expert filtered the AEs by hand. As long as the number of eliminations is not too great, the forecasting network should still be able to learn from the data. In the two lab datasets, less than 10% of AEs were rejected for these reasons.

It is important to note that we do not expect a large number of clusters due to the nature of the AE catalog. The data processing involved in the catalog creation purposely filters out as much noise as possible in an attempt to leave only signals that have the potential to be seismically important. Without such a catalog, we would expect to encounter many more noise clusters.

Unlike many NDT studies, we are not able to determine the true mechanism labels for the AEs. Though this means we will not be able to provide a full account of changes occurring in the fault gouge, labels are not
required for our purposes. Our main goal is to identify elements of the acoustic signal that contain predictive information, which does not necessitate knowledge of the true labels.

5.2. CSOM and Clustering

For the AE clustering we use a conscience self-organizing map (CSOM) (DeSieno, 1988), which is a more efficient variant of the original Kohonen self-organizing map (K SOM) (Kohonen, 1982). In general, SOMs perform a topologically ordered mapping from a multidimensional data space to a 2D lattice space. Data points that map closely together on the lattice are also relatively close together in the data space. SOMs perform a combination of vector quantization and dimensionality reduction similar to PCA and autoencoders. Once the unsupervised SOM training process is complete, clusters are defined on the lattice using either a clustering algorithm, such as k-means, or interactively (by hand) using a variety of visualizations. We chose to use a CSOM over other clustering methods largely due to the data complexity. The nature of the bias is very different from the NDT laboratory setups, so initially we were not sure how many clusters to expect or even if we should expect more than one. If we did end up with only a single cluster, the benefit of a CSOM is that the lattice would provide a way of describing the continuum. Even if we did see clusters, we speculated that they would be complicated shapes with fuzzy boundaries in the data space. With an eye toward the future, we also expected that with the addition of more receivers this picture would only become more complex. Thus, we concluded that this problem is not well suited to algorithms like k-means. An alternative, and likely more precise, method would be to use a coarse clustering network and GMMs (Seydoux et al., 2020) in combination with fine grained clustering using a CSOM.

We trained a CSOM using a 15 × 15 lattice on over 8 million AEs for 20 million learning steps. The training parameters are listed in Table S8. Once training was complete, we used k-means and interactive interpretation to partition the CSOM lattice into clusters. We evaluated many potential cluster configurations using cluster validity indices (CVIs) including CH-VRC (Caliński & Harabasz, 1974), Conn_Index (Tasdemir & Merényi, 2011), DBI (Davies & Bouldin, 1979), the Gap statistic (Tibshirani et al., 2001), GDI (Bezdek & Pal, 1998), PBM (Pakhira et al., 2004), and Silhouette (Kaufman et al., 1990) and visualizations such as the U-matrix (Ultsch & Siemon, 1990) (and modified U-matrix [Merényi et al., 2007]), octagonal erosion (Cottrell & de Bodt, 1996), and CONNvis (Tasdemir & Merényi, 2009).

Ultimately, we found that an interactive clustering with a k value of five produced the best results. The trained and interpreted CSOM lattice is shown in Figure 1 along with the typical waveforms for each cluster. Note that since we do not have the true labels of these clusters, they will henceforth be referred to by color. We show the typical waveforms to illustrate both the complexity of the signals as well as the differences between the clusters. The red cluster contains long, high amplitude events. The blue cluster is similar, but shorter and with lower amplitude. The purple and green clusters look similar in that they are both short with low amplitude, but the purple cluster has a significantly higher average and peak frequency. The orange cluster is somewhat of a catch-all for whatever signals remain, with a middling length, frequency, and amplitude.

The complexity, and decay seen in the waveforms representative of the clusters can be understood as follows. Each gouge layer is a linear elastic system 0.4 × 10 × 10 cm. This elastic system, really a mechanical cavity or wave guide, is embedded in a steel environment (the drive block and side blocks) that has sound speed up to an order of magnitude larger than the gouge. The acoustic contrast between mechanical cavity and environment corresponds to a mechanical index of refraction of order 6. An elastic wave in a gouge layer will be reflected many times in the gouge layer before exiting into the steel environment (see Trugman et al., 2020). Consequently, a pulse received at a sensor has a distinctive pattern.

At a simplistic level the nucleation of AEs in a granular material are likely the result of grain fracturing, particle sliding/rolling, and the breaking of force chains. However, because our experiments were conducted at low normal stresses, we assume that grain fracturing is insignificant (e.g., Anthony & Marone, 2005; Mair et al., 2002; Scuderi et al., 2014). These micro-mechanical processes, sensibly represented by the sudden appearance of a force dipole in the gouge layer, launches an elastic wave that drives the gouge layer in a low lying shear mode. This motion of the gouge layer rings down slowly, decaying at a rate determined by how rapidly the acoustic contrast allows leakage into the steel environment. Note, the detector itself rings down...
as well, complicating the net behavior. This leaked signal goes on to the sensors and is an acoustic pulse contributing to the acoustic emission. The envelope (Figure 2) of these AEs has a sharp onset, followed by exponential decay, and contains energy within the 100+ kHz range. However, the characteristics of the source may complicate this behavior, resulting in AEs with long rise times and long ring down times. Taken together, these factors result in the complex waveforms seen in many of the AEs.

5.3. Cluster Selection

Next, we examined the production of AEs by cluster throughout the failure cycle to look for temporal patterns. This is the first step in the process of identifying which clusters are likely to be useful for forecasting and which can be set aside. In general, predictive features are expected to exhibit nonlinear temporal patterns. Linear trends may provide cycle maturity information, but only when all cycles are roughly the same length. The lab data has a wide range of recurrence intervals and thus the number of AEs per cycle is highly variable. Therefore, we cannot rely on learning absolute parameter values, as would be required for linear trends, and must instead look for universal nonlinear patterns.

As we did with the Broom data, we plotted the cumulative number of AEs over the failure cycles. Three representative cycles are shown in Figure 3. In general, the orange cluster contains the largest number of AEs and the blue and green contain the fewest. Most of the clusters display nonlinear behavior that may be useful for determining the percentage of the recurrence interval that has elapsed. However, not all clusters appear to have predictive capability; the orange cluster is nearly linear and thus will not be useful for failure forecasting.

Our results suggest that AE production is tied to the normal stress of the system. We observe that stick-slip events with comparable normal stress exhibit very similar cumulative AE patterns, as in Figures 3b and 3c.

Figure 2. The envelope of the seismic data showing an instance of the same area of the gouge rupturing three times, each with decreased amplitude.

Figure 3. (a) (Left axis) Histogram of the number of acoustic emissions (AEs) by cluster over one seismic cycle. Colors correspond to those of Figure 1. (Right axis) Shear stress. (b and c) Cumulative AEs by cluster for two events of similar high stress. (d) Cumulative AEs for a low stress event.
For both quakes, the rate of purple and green AE production decreases toward failure. The red AE production rate decreases shortly after stress onset and then increases again in the second half of the cycle. The same is true for the blue cluster to a lesser extent. Even the relative proportions of the clusters are the same between the two quakes; purple and red intersect or nearly intersect just before failure, purple and orange track closely for the first half of the cycle, etc. Though only two quakes are shown here, these similarities appear in all quakes of a similar normal stress.

In addition, we find that quakes at different normal stresses produce different temporal patterns of AEs (Figures 3a and 3b vs. Figure 3c, with additional quakes in Figure S13). The two quakes in Figures 3a and 3b occur at relatively high stress whereas the quake in Figure 3c occurs at low stress. In the low stress quake, every cluster is nearly linear. We do not see the curved patterns that are evident in the high stress quakes. Additionally, the relative proportions of the clusters are completely different at low stress. Now the largest cluster is the blue cluster, followed by the green, purple, orange, and red clusters. Again, though we only show one quake here, this pattern is repeated among other low stress quakes.

The similarity of patterns at similar stress indicates that AE production is connected to the underlying physics of the system. This notion is supported by the laboratory AE analysis in Bolton et al. (2020). The difference in patterns between different stress conditions effectively eliminates the possibility of time-to-failure forecasting based exclusively on temporal patterns of AE production. This is because the linear patterns at low stress are unlikely to contain useful information. In Earth, the stress state of the fault is generally not known and that presents a challenge.

Next, we examined the cumulative AE energy over each failure cycle. Figure 4a is a plot of the cumulative energy from all AEs for a single stick-slip event. As the system progresses toward failure, the amount of energy contained in the AEs increases according to a power law, as was also shown in Bolton et al. (2020). It is likely that acoustic energy derived exclusively from AEs can be used to estimate failure times as was described by Lubbers et al. (2018). If true, then the use of the continuous AE signal is not a necessary ingredient to yield successful predictions with the caveat that the earthquake magnitude of completeness is sufficiently small. Furthermore, this highlights the importance of obtaining high-resolution event catalogs that are rich in small magnitude events (e.g., Mousavi et al., 2020; Ross et al., 2019). This pattern can also be seen in the cumulative energy from events generated by the Broom model (Figure S1). The power law cumulative energy was vital to the forecasting success using the Broom data, which suggests that the AEs contain the necessary information for TTF forecasting. If this is true, then we have no need for the rest of the acoustic signal and can simply discard it.

Continuing the task of cluster selection, we divide the cumulative energy by AE cluster, as in Figure 4b. At first glance, it appears that the red cluster is all that is needed for failure forecasting because it contains the vast majority of the total AE energy and retains the power law shape. However, as with cumulative AEs, the stress state of the fault plays a role in the energy contained within in each cluster. Figure 4c shows the cumulative energy by cluster for a low stress event. Compared to the high stress event in Figure 4b, the proportion of the total energy contained in the red cluster is reduced. As a result, the blue cluster contains...
a much larger proportion of the total energy. The power law shape is not as prominent at this stress level, but it is still present in the blue and red clusters. This indicates that the blue cluster is a vital source of predictive information at low normal stress. The three remaining clusters contain very little energy and have linear trends at both stress levels, which indicates that they will not be useful for forecasting. For additional examples, see Figure S14.

We also examined the cumulative trends for the other features used in the clustering scheme (frequency, duration, counts, etc.). These features show mostly linear patterns with some nonlinearity introduced as a byproduct of the energy (e.g., in maximum amplitude). We also observe consistent patterns among quakes of the same stress level and different patterns between different stress levels. Like with the cumulative number of events in the Broom model, we continue to include these features because the raw values may provide the forecasting network clues as to the overall stress level.

Taken together, these findings suggest that only the red and blue clusters are seismically useful for forecasting time-to-failure at all relevant stress levels. Though the orange cluster is the largest at high stress, it consistently displays low, linear cumulative energy. As a result, this cluster is unlikely to contain much, if any, predictive information, so we will set it aside. The purple cluster is nonlinear at high stress and has slightly more energy at higher frequencies, which in turn, produces a strong boundary between it and the other clusters in the CSOM lattice. This cluster is also unlikely to be predictive and we will not use it. The green cluster is perhaps the least remarkable; it is relatively small and has low, linear energy. Green AEs are proportionally larger at low stress than at high stress. This cluster may consist of spurious signals that were mistakenly included during the cataloging process. As such, we will not include it in the forecasting scheme. The blue and red clusters, on the other hand, display the nonlinear energy we saw in the Broom data at all stress levels. The differing proportions of each at various stress levels mean that both clusters are required to capture the full stress spectrum. Therefore, the inputs to the forecasting network will include only the blue and red clusters and the other three will be discarded.

6. Forecasting

6.1. Training and Testing Data

For time-to-failure and shear stress forecasting, we only use AEs from the red and blue clusters. Network inputs consist of the seven cumulative features used to train the CSOM as well as the cumulative number of AEs. To improve network performance, we experimented with three additional input features: timestamps, shear stress, and shear stress slope (Table S10 and Figure S22). In the real earth, where shear stress measurements are not available, the latter two features could be obtained for TTF forecasting via an additional network. For the timestamps, we subtracted the respective cycle start time from each AE time so that each cycle began at time zero. Shear stress was included as recorded at the time of each AE. We calculated shear stress slope using a best-fit line on the shear stress data. For each AE, we calculated the best-fit line to the shear stress versus time data on a window of 50 AEs centered on the AE in question. We found that the best results came from including the stress slope during training. Including timestamps slightly improved the MAE for long cycles at the expense of short cycles. Including shear stress had little impact on network performance.

All input features were normalized to [0,1] according to their maximum and minimum values across the entire data set. In order to maintain the temporal ordering of AEs, each input vector contains the eight features for each of the two clusters plus stress slope, bringing the input size to 17. In addition, each input vector contains the features for the current AE as well as the features from the previous AE from the other cluster. Over time, the inputs resemble a step function where the features for each cluster are held constant until an AE from that cluster occurs.

We divided the labquakes into train and test sets, with 80% of the quakes for training and the remaining 20% for testing. We did not use a validation set because of the limited data availability. Unlike other studies, we divided the data based on entire cycles rather than on time windows. We used this approach because as the biax operates, the fault system changes via human interventions such as load increases and natural consequences such as gouge thinning. These factors can affect the recurrence interval and by using contiguous time chunks for the sets, the network may come to rely on these system variables as a means of forecasting.
Randomizing set selection in time can help mitigate this and force the network to learn the physics of failure.

The test set was randomly chosen to have roughly the same recurrence interval distribution as the full data set (Figure 5). As with the Broom model data, all recurrence intervals and stress levels were included in a single data set and trained upon simultaneously. The vast majority of recurrence intervals are around 10 s, which left very long/short recurrence intervals underrepresented in the training set. To address this, we experimented with techniques meant to alleviate issues with unbalanced data sets. We began by upsampling long and short cycles so that these quakes were seen by the network more than once per training epoch. This had little impact on the total MAE and substantially increased the network training time (Table S10). Next, we tried pretraining by decreasing the number of average cycles by only including a small random subset of them in each training epoch. These network weights were then used to initialize another network that trained on the full data set. Both the pretraining networks and the fine-tuned networks generally performed worse than the default training method (Table S10). Finally, we experimented with loss weighting. We weighted average cycles so that they contributed less to the total loss than short and long cycles. Again, this did not improve the total MAE (Table S10).

6.2. Network Variations

We tested a variety of network architectures, including variations of LSTM, attention networks, and meta learning. To improve the results, we experimented with network modifications such as different optimizers, gradient clipping (Pascanu et al., 2013), weight decay (Loshchilov & Hutter, 2019), etc. In addition, to avoid overfitting, we employed dropout (Srivastava et al., 2014), early stopping (Prechelt, 1998), and ensembling (Dietterich, 2000). Table S9 shows the variety of hyperparameters and network modifications that we tested.

We began by forecasting with a simple multilayer LSTM network (Figures S16 and S17). We used this network to test the input vector modifications (i.e., the stress slope) as well as the methods of handling the unbalanced recurrence intervals in the data set. We also experimented with time-gated LSTM (TGLSTM), which is an LSTM variation designed to better handle irregularly sampled data by modifying the network according to the time interval between data points (Sahin & Kozat, 2018) (see Figure S18).

We also forecasted TTF and shear stress with the attention network described previously. In addition, we tested a variation of this network that attempts increase efficiency by adding momentum (Nguyen et al., 2020).

Finally, we tested various meta learning techniques that focus on learning from multiple datasets. As with the Broom data, our tactic is to start with a simple network and then add complexity as needed. The attention network is more complex than the LSTM network and meta learning is yet another step up in complexity. We include this network because it has the potential to outperform both LSTM and attention.

In our experiments, we used a domain adaptive meta learning (DAML) network (Qian & Yu, 2019). Before training, data must be divided into “domains” representing some joint characteristic. We assigned cycle domains based on the normal stress (see Figure S15). The step-up-step-down laboratory data consists of seven load levels for a total of seven domains. For comparison, we also experimented with using three domains based on recurrence interval (short, average, long). The training process begins by initializing a “meta network” that can be any network type of the user’s choosing (e.g., LSTM, attention, etc.). Weights from the meta network are used to initialize a new network for each domain (Figure S21). At this stage, each domain network is an exact copy of the meta network. The domain networks are trained separately for some small predetermined number of epochs. After this training period, weight changes are averaged across the domains and used to update the meta network (Rajeswaran et al., 2019). The domain networks are discarded and the process begins again until some number of meta-epochs is reached. We tested this network using our multilayer LSTM and attention networks as the base.
All networks were trained using the smooth L1 loss function. Results from these networks frequently contained illogical undulations in TTF that we wished to avoid. To address this, we added a penalty term to the loss function that encourages the network to produce TTF values with a slope of $-1$. The $-1$ slope comes from the simple fact that TTF should decrease by 1 s for every second that has elapsed. We calculated slopes using least squares on the denormalized TTF values for each sequence. Next, we found the difference between the actual slope and desired slope of $-1$ and renormalized this loss. We weighted the slope loss term to be less important than the L1 loss for the network update. In some cases, the addition of this term produced smoother results and was most beneficial early in the training.

6.3. Shear Stress Results

We trained the networks to simultaneously forecast shear stress and time-to-failure, but for simplicity, we will discuss them one at a time. We report our results using the mean absolute error (MAE) calculated over each failure cycle. MAE is presented separately for shear stress and TTF, though they were combined during network training. It is important to note that we do not expect the trained network to have perfect accuracy. Due to the nature of the challenge, forecasting results rarely achieve very high accuracy. Instead, our goal is to present a technique that makes reasonably accurate forecasts and can be improved upon in the future.

We began by training the LSTM network. The learning history for our best run is shown in Figure 6a and the hyperparameters are listed in Table S11. Though we ran the network for an additional 150 epochs, we used the weights from the point marked “early stopping” at which the test loss stagnates. The large spikes in error, particularly near the beginning of training, are typical of recurrent networks and are exacerbated by the unbalanced data set. For this network, we found no benefit from incorporating TTF slope into the loss function.

Next, we trained the attention network using the same training and testing sets as with the LSTM. The learning history for our best run is shown in Figure 6b and the hyperparameters are listed in Table S11. Once again, we use the network weights from the “early stopping” mark to make the final forecasts. This network did benefit from incorporating TTF slope into the loss function. Compared to the LSTM network, the attention network has fewer error spikes, but takes longer to converge.

Example cycles with the forecasted and measured shear stress are shown in Figure 7, with additional results in Figure S25. As with the Broom data and the cumulative feature plots, these figures each portray a single, representative quake from the onset of stress to failure. The MAE for each run is listed in Table 1.
The LSTM network produces a mix of acceptable and poor forecasts. For a number of cycles, the error is quite low, but for others the LSTM forecast is far from the true shear stress. Often, the forecasted stress is quite jagged, even after the initial network adjustment period. In some cases, the network suddenly becomes much more accurate in the last few seconds of the recurrence interval, whereas in others it drifts away from the true values. The network often does not even capture the general shape of the stress and instead meanders until failure is reached.

We can further examine the network forecasts by dividing the quake cycles by stress level. The nature of the data is such that low stress has the widest variety of recurrence intervals whereas high stress consists of mostly average recurrence intervals (see Figure S15b). We divide the stress according to the steps in the step-up-step-down lab experiment. This produces seven domains where domain 1 has shear stress <1 MPa and domain 7 has shear stress around 3–3.5 MPa.

The LSTM network has the highest MAE in the middle domains (3–5) and a lower MAE at very low and very high stress. Visually, the high stress forecasts are the most accurate and least erratic. The network generally gets the shear stress shape correct and the forecasted values are more or less around the true stress. At medium stress, when the network is wrong, it is very wrong (see the y-scale in Figure 7). A similar phenomenon happens at low stress, but this is balanced against cycles that have very accurate forecasts. Clearly the network is not able to fully overcome the differences in recurrence interval, leading to large differences in quality in the domains with wide recurrence interval ranges.

It is important to acknowledge that we only obtain this interesting result because we trained the LSTM network on all stress conditions simultaneously. Had we only trained on a single stress level, high stress quakes, for example, the network would appear to perform very well and thus inspire false confidence. In the lab, stress conditions are highly controlled and straightforward, but in the real earth they are likely unknown and complex. For our network to perform well on the real earth, it must be able to generalize across multiple background stress conditions.

### Table 1

| Stress MAE by domain | Name     | Total   | 1     | 2     | 3     | 4     | 5     | 6     | 7     |
|----------------------|----------|---------|-------|-------|-------|-------|-------|-------|-------|
| LSTM                 | 0.1087   | 0.0532  | 0.0817| 0.1617| 0.1325| 0.1354| 0.1035| 0.1034|
| Attention            | 0.0658   | 0.0395  | 0.0579| 0.0945| 0.0709| 0.0632| 0.0564| 0.0788|
| Ensemble             | 0.0699   | 0.0378  | 0.0618| 0.0801| 0.0794| 0.0876| 0.0644| 0.0780|

*Note. Normal and shear stress increase from domain 1 to domain 7. The best result in each domain (as well as total MAE) is in bold. MAE is broken down by the seven stress domains in the step-up-step-down laboratory experiment.*
We also note that the network performs particularly poorly on cycles from the second lab experiment, p4583. This behavior is not unexpected because this experiment was included to diversify the stress conditions and recurrence intervals. This experiment has fewer quakes overall, which makes learning them more challenging. However, it is quite apparent that LSTM struggles to adapt to new conditions. Perhaps given more data the network would perform better, but we have to be cognizant of the fact that in the real Earth data is limited and stress conditions are more complex.

After this disappointing performance, we turn to the forecasting results from the attention network. The attention network performs significantly better than the LSTM network, with a total MAE that is nearly 50% smaller. In general, the attention forecasts are very smooth, without the jagged edges that defined the LSTM results. The attention forecasts for individual cycles are very accurate after the initial adjustment period, which tends to be fairly short. Even the less-accurate forecasts generally capture the shape of the shear stress curve. Unlike the LSTM network, once the attention network has settled on a pattern for a specific quake, it consistently follows that curve and doesn't drift away at the end of the cycle.

Dividing the cycles by stress, we see that the attention network performs well at all stress levels. Like the LSTM, it performs the worst on domain 3, but even there the attention network is a clear improvement. Visually, when the LSTM network performs poorly, the attention network is more likely to be slightly off in its forecasts. However, this is not always the case and the attention inaccuracies are much smaller than for the LSTM. This indicates that the attention network has a greater capacity to learn from a variety of stress environments than the LSTM network.

On the topic of the two lab experiments, the attention network performs much better on p4583 than the LSTM network does. In general, the p4583 quakes have higher attention error than the p4581 quakes, but the network performs better on p4583 than LSTM does on p4581.

There’s an argument to be made that even when the attention network is wrong, its forecasts are more useful than those of the LSTM. A watch that is always off by 5 min is more useful than a watch that is erratic. In the same way that the first watch only needs a small modification to be accurate, the attention network likely also only needs small tweaks to correct inaccurate forecasts. This could come from the addition of another input feature or a minor architecture modification. This is comparatively simple to the work required to improve the LSTM network, which calls for significantly more data and even then may not produce accurate results.

Finally, we prepared an ensemble by averaging the results of the two networks. Even though the LSTM performs much worse than the attention network, this is a good practice because different networks often learn different aspects of the data. The combination gives a wider view of the data and can lead to better forecasts. Indeed, the ensemble performs better than its constituents in stress domains 1, 3, and 7. However, the attention network still reigns supreme overall.

Overall, we find that the attention network is easily able to forecast shear stress. The LSTM network performs well on some quakes, but the different stress conditions are simply too complex for it to learn with this small amount of data. The attention network, however, performs well under all conditions and thus is a good candidate for use in the real earth.

6.4. Time-To-Failure Results

As before, we report our results using the MAE calculated over each failure cycle. In our calculations, we omit the first 0.5s of each cycle since the network has not yet received enough data to make reasonable forecasts. Including this adjustment period in the error disproportionately impacts the short/long cycles and portrays their forecasts as worse than they are. With TTF forecasting, we determine that a cycle has been successfully forecasted if the MAE is less than 10% of the recurrence interval. For comparison, the results for a control network that always outputs the average recurrence interval of the data set are shown in Table 2.

The learning histories are shown in Figure 8 and the hyperparameters are listed in Table S11. We again employ early stopping to select the appropriate network weights. We note that in the DAML runs the test data nearly always outperforms the train data. This was the case in every network configuration that we
tested. The loss calculations for the test and train sets are identical, so this appears to be caused by the network itself.

MAE for all runs is listed in Table 2 and success percentages are listed in Table S12. Forecasted versus actual TTF for all networks on three example cycles are shown in Figure 9, with additional examples in Figure S26. The recurrence interval for each quake is noted in the bottom left corner.

Compared to the shear stress forecasts, the LSTM network performs better than expected. In general, the forecasts are relatively close to the true values and the overall shape is roughly a line with a slope of −1. We continue to see the jaggedness we saw in the shear stress forecasts, though to a lesser extent. The forecasts often drift away from the actual TTF (and the −1 slope line) in the last second or so of the cycle.

As discussed previously, the data set consists of a wide range of recurrence intervals. We broke these down into short, average, and long cycles for further analysis. The LSTM network performs very well for average quakes, which have a recurrence interval of around 8.5–11 s. However, quakes with very short or very long cycles have much higher error. The network generally produces an output with the correct shape for these cycles, but fails to adjust the positioning on the y-axis. This may occur because the network prefers to decrease loss for the average case rather than risk larger loss by producing a generalized model. However, when compared to the control results in Table 2, it is clear that the network is an improvement over simple averaging.

Next, we combined the best performing LSTM runs into an ensemble. We did this because we observed that the LSTM struggled to learn both the short and the long quakes at once. In general, any given run would perform well on either the short or long quakes, but not both. For the LSTM ensemble, we specifically selected the best runs for each recurrence interval bin. The resulting ensemble performs better than the best single LSTM run at every recurrence interval.

The attention network clearly performs better than the LSTM network: MAE is lower and visually the forecasts are closer to the true values. Some of this improvement is likely due to the smoothing effects of the attention layer. Like with the shear stress forecasts, the attention network results are very smooth without any sudden directional changes. Even when the attention network misses the mark, it is generally closer to the true TTF than the LSTM network.

The TTF forecasted by the attention network for the short and average cycles is more accurate than from the LSTM network and its ensemble. Surprisingly, the networks perform nearly the same on the long quakes, perhaps signaling that there are simply not enough cycles for complete learning. Though the attention network still struggles with some of the very long/short cycles, it produces successful forecasts for the majority of the test set.

Unfortunately, none of our DAML experiments were particularly successful (see Figures S23 and S24). The total MAE for both network configurations is worse than the LSTM and attention networks. This is largely

### Table 2

| Network                  | TTF MAE | Total | <8.5s | 8.5–11s | >11s |
|--------------------------|---------|-------|-------|---------|------|
| Control                  | 2.0726  | 3.2165| 1.6655| 4.1711  |
| LSTM                     | 1.1582  | 1.6203| 0.8331| 1.4430  |
| LSTM ensemble            | 1.1000  | 1.5296| 0.8039| 1.3531  |
| Attention                | 1.0504  | 1.5125| 0.6861| 1.4098  |
| DAML (LSTM)              | 1.3969  | 1.6650| 0.9056| 2.1401  |
| DAML (attention)         | 1.5066  | 1.9529| 0.9223| 2.2974  |
| Ensemble                 | 1.0400  | 1.4181| 0.7641| 1.2919  |

Note. Bold entries indicate the best MAE for each recurrence interval bin.

*Test events were divided into recurrence interval bins for further comparison.*

---

**Figure 8.** Time-to-failure (TTF) training histories for the (a) LSTM (b) attention (c) domain adaptive meta learning (DAML) with LSTM and (d) DAML with attention networks showing loss over time for the train and test sets. The final network weights were taken from the point marked early stopping.
because DAML performs astoundingly poorly on the long cycles. On the short and average cycles, both DAML variations do about the same as LSTM. DAML has been used successfully on other datasets, but we appear to lack the data volume and computing power required to train it well. We believe this network's performance could improve given additional training data and more powerful computing resources.

Finally, we calculated a mixed ensemble consisting of the best runs on short and long cycles across all the networks tested. Indeed, the ensemble improves the forecasting accuracy on the short and long cycles. The ensemble does unfortunately have some jagged edges due to its LSTM components, but on the whole it performs nicely.

These results pose a critical question: why do some quakes always have poor forecasts? For some quakes, this may be due to foreshocks before failure that muddy the cumulative features. Some of these foreshocks were large enough to be considered coseismic stress drops and separated out, but this was not always the case. The attention network clearly combats this muddying by examining the feature trend over a window of previous time steps as well as the data at the current time step. For simple networks such as LSTM, a trend-based view may be beneficial.

For the long quake cycles in particular, poor performance is likely due to lack of data. The spread of long cycles is much greater than the spread of short cycles. As we saw with the Broom tests where only some normal stress values were seen during training (Supporting Information S1), the further the test data are from the training data, the less accurate the forecasts become. The extremely long quakes are seen very rarely, so we should expect higher error.

7. Prior Work

Recently, Rouet-Leduc et al. (2017) used the random forest algorithm to forecast TTF for lab data using the continues AE signal for feature extraction (in contrast to event catalogs applied here). The input data consisted of 100 statistical features calculated from moving windows of raw data from 10 sequential quake cycles. The subsequent 10 cycles were used as the test set. The quake cycles had variable recurrence interval, but similar shear stress. The authors reported an $R^2$ of 0.883 for the test data, a metric which we do not calculate here. The authors performed an additional test where the random forest was trained on one load level (5 MPa) and tested on another (8 MPa), which resulted in a test set $R^2$ of 0.741.

Los Alamos National Lab and the Department of Energy Office of Science (Geoscience program) sponsored a Kaggle competition in 2018–2019 with the goal of improving lab forecasting results (Johnson et al., 2021). The winning solution used a combination of gradient boosted trees and a convolutional neural network to forecast TTF from raw seismic data. The team found that the two networks learned different components of the signal so averaging the two results gave a more complete view of the data. This setup achieved a MAE of 2.26 s. Some teams published papers detailing their methods, including Zaidi et al. (2020), who used a CNN and LSTM for an MAE of 1.51, and Brykov et al. (2020), who used XGBoost for an MAE of 1.91. It is unclear
if the reported MAE from these follow-up papers comes from the small “public” test set available to all during the competition, or the larger “private” test set, which was used to determine the competition winners.

Decision tree forecasting methods have been used in several subsequent papers, again employing the continuous AE signal. Hulbert et al. (2019) forecasted TTF, among other features, for fast and slow slip in the lab with an $R^2$ of 0.88. van Klaveren et al. (2020) used random forest to forecast TTF from rotary shear experiments on salt samples. After correcting for machine resonance, the team achieved an $R^2$ of 0.85. Corbi et al. (2019) used gradient boosted regression trees on geodetic data from a laboratory subduction experiment. They report an overall $R$ value of 0.3 for their TTF forecasting model, though most individual cycles are in the 0.7–0.8 range. They also report RRMSE values for each cycle, which range from mid-80s to low-20s. The average RRMSE for each of our LSTM and attention networks are in the range 28–33. Tanyuk et al. (2019) also forecasted TTF from biaxial lab data using statistical features. The authors experimented with several decision tree algorithms, and achieved the best MAE of 1.65s from random forest. Our MAE values are reported in Table 2, where the lowest MAE is 1.04s.

Other studies have used machine learning to forecast earthquakes without directly outputting time-to-failure. Corbi et al. (2020) used the same laboratory subduction setup as Corbi et al. (2019) with RUSBoost to forecast the alarm state of the fault (i.e., a binary classification of if the slip rate of the fault exceeds some threshold). The AETA Competition (The AETA Competition, 2021) challenged participants to make binary forecasts each week given a combination of acoustic and electromagnetic data from the Sichuan and Yunnan provinces from the previous week. The top team for the 2020 competition achieved 75% correctness.

8. Discussion

Overall, the forecasting scheme was successful. The attention network in particular was able to forecast TTF and shear stress within a reasonable amount of error for the majority of the test set. These results indicate that AEs from the red and blue clusters contain enough information to forecast time-to-failure without using the rest of the seismic trace.

In general, the LSTM network performs better on the average and long cycles. The attention network (and the ensemble) performs roughly equally well on the short and long cycles, though the average cycles had substantially lower MAE. This difference is largely due to the fact that average cycles are overrepresented in the lab data. With the short and long cycles, the network simply does not have enough examples to adequately learn. With the Broom model, we found that the network was able to learn all recurrence intervals given a equally distributed training set. Given more short and long lab cycles, it is likely that forecasting performance would improve.

In their work with forecasting via random forest, Rouet-Leduc et al. (2017) found that variance of the signal was the most important input feature they tested. Determining the most useful feature from an LSTM model is much more difficult. From the temporal AE results in Section 5.3, it appears that energy, which is related to variance, likely has great importance. We tested this by forecasting using cumulative energy as the only input feature. The forecasts of this network are clearly inferior to our previous results (Table S10). This suggests that accurate forecasting is more complicated than simply monitoring AE energy. As we move to Earth and attempt to apply ML approaches to discern fault physics from seismic signals, in depth exercises analyzing laboratory and simulation data are key to informing how to proceed. This study represents a key element in this exercise.

9. Conclusions

In this study, we forecasted shear stress and TTF for laboratory earthquakes using machine learning. Through clustering, we found that a subset of acoustic emissions contain sufficient information for these tasks. Thus, network training does not require continuous seismic data. We obtained the most accurate forecasts using an attention network, though further improvements could be made. Our procedure produces more accurate forecasts than existing work listed in Section 7 and does so over a wider variety of normal stress conditions.
Acknowledgments

H. Jasperson and M. de Hoop were supported by U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Chemical Sciences, Geosciences, and Biosciences Division under grant number DE-SC0020345 and P. A. Johnson and R. A. Guyer by grant 89233218CNA000001, respectively. C. Marone was supported by the European Research Council via Advance Grant 835012 (TECTONIC) and by the US Department of Energy via grants DE-SC0020512 and DE-EE0008763. Special thanks to Erzebet Merényi for the use of the NeuroScope software tools for CSOM training and to Michael Puthawala for his contributions to Supporting Information S1.

References

Anthony, J. L., & Marone, C. (2005). Influence of particle characteristics on granular friction. *Journal of Geophysical Research, 110*(B8). https://doi.org/10.1029/2004JB003399

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio, & Y. LeCun (Eds.). *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings. Retrieved from http://arxiv.org/abs/1409.0473

Bezdek, J. C., & Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28*(3), 301–315. https://doi.org/10.1109/3477.678624

Bolton, D. C., Shroedharian, S., Rivière, L., & Marone, C. (2020). Acoustic energy release during the laboratory seismic cycle: Insights on laboratory earthquake precursors and prediction. *Journal of Geophysical Research: Solid Earth, 125*(8), e2019JB021895. https://doi.org/10.1029/2019JB021895

Brykov, M. N., Petryshynets, I., Prunuc, C. I., Efremenko, V. G., Pimenov, D. Y., Glisan, K., et al. (2020). Machine learning modelling and feature engineering in seismology experiment. Sensors, 20(15). https://doi.org/10.3390/s20152428

Calabrese, L., Campanella, G., & Proverbio, E. (2013). Identification of corrosion mechanisms by univariate and multivariate statistical analysis during long term acoustic emission monitoring on a pre-stressed concrete beam. *Corrosion Science, 73*, 161–171. https://doi.org/10.1016/j.corsci.2013.03.032

Caliski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Corbi, F., Huguet, S., Rouet-Leduc, B., Pyrak-Nolte, L. J., Beroza, G. C., Marone, C. J., Hulbert, C., & Reade, W. (2021). Laboratory earthquake forecast

Data Availability Statement

The laboratory data used in this study are publicly available from the Penn State Lab at https://personal.ems.psu.edu/~cmj38 and https://scholarsphere.psu.edu/resources/a8e93370-2151-40e7-932e-4116df643bd.
Sutskover, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), Advances in neural information processing systems. Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2014/file/a14ac55af2f7472c5d894ec13c743d2-Paper.pdf

Tanyuk, O., Daviau, D., South, C., & Engels, D. W. (2019). Machine learning predicts aperiodic laboratory earthquakes. SMU Data Science Review, 2(2).

Tasdemir, K., & Merényi, E. (2009). Exploiting data topology in visualization and clustering of self-organizing maps. IEEE Transactions on Neural Networks, 20(4), 549–562. https://doi.org/10.1109/TNN.2008.2055409

Tasdemir, K., & Merényi, E. (2011). A validity index for prototype-based clustering of data sets with complex cluster structures. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(4), 1039–1053. https://doi.org/10.1109/TSIBM.2010.2104319

The AETA Competition. (2021). Retrieved from https://aeta.io/

Tibshirani, R., Walthier, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Methodological), 63(2), 411–423. https://doi.org/10.1111/1467-9868.00293

Trugman, D. T., McBrearty, I. W., Bolton, D. C., Guyer, R. A., Marone, C., & Johnson, P. A. (2020). The spatiotemporal evolution of granular microslip precursors to laboratory earthquakes. Geophysical Research Letters, 47(16), e2020GL088404. https://doi.org/10.1029/2020GL088404

Ultsch, A., & Siemon, H. (1990). Kohonen’s self-organizing feature maps for exploratory data analysis. In Proc. of the international neural network conference (pp. 305–308).

van Klaveren, S., Vasconcelos, I., & Niemeijer, A. (2020). Predicting laboratory earthquakes with machine learning.

Yin, C.-L., Liu, M.-C., & Chen, J.-L. (2013). Applying the self-organization feature map (SOM) algorithm to AE-based tool monitoring in micro-cutting. Mechanical Systems and Signal Processing, 34(1–2), 353–366. https://doi.org/10.1016/j.ymssp.2012.05.001

Zaidi, T., Samy, A., Kocatürk, M., & Ateş, H. F. (2020). Learned vs. hand-crafted features for deep learning based aperiodic laboratory earthquake time-prediction. In 2020 28th signal processing and communications applications conference (pp. 1–4). https://doi.org/10.1109/SICAP.2020.9302474

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. 2017 IEEE international conference on prognostics and health management (ICPHM) (pp. 88–95). https://doi.org/10.1109/ICPHM.2017.7998311

**References From the Supporting Information**

Beniöff, H. (1951). Earthquakes and rock creep: Part I: Creep characteristics of rocks and the origin of aftershocks. Bulletin of the Seismological Society of America, 41(1), 31–62. https://doi.org/10.1785/01184518182

Ben-Zion, Y., & Lyakhovsky, V. (2002). Accelerated seismic release and related aspects of seismicity patterns on earthquake faults. In M. Matsuda, P. Mora, A. Donnellan, & X.-c. Yin (Eds.), Earthquake processes: Physical modelling, numerical simulation and data analysis part II (pp. 2385–2412). Birkhäuser Basel. https://doi.org/10.1007/978-3-0348-8197-5_12

Brodsky, E. E. (2019). The importance of studying small earthquakes.

Benioff, H. (1951). Earthquakes and rock creep: Part II, 89–121. American Geophysical Union. https://doi.org/10.1029/93JB00357

Bufe, C. G., Nishenko, S. P., & Varnes, D. J. (1994). Seismicity trends and potential for large earthquakes in the Alaska-Aleutian region. Pure and Applied Geophysics, 142, 83–99. https://doi.org/10.1007/BF00875969

Bufe, C. G., & Varnes, D. J. (1993). Predictive modeling of the seismic cycle of the greater San Francisco Bay region. Journal of Geophysical Research, 98(B6), 9871–9883. https://doi.org/10.1029/93JB00357

Gigante, S., van Dijk, D., Moon, K. R., Strzalkowski, A., Wolf, G., & Krishnaswamy, S. (2019). Modeling global dynamics from local snapshot data with deep generative neural networks. In 2019 13th international conference on sampling theory and applications (SAMPTA) (pp. 1–4). https://doi.org/10.1109/Sampta45681.2019.9030978

Gutenberg, B., & Richter, C. F. (1944). Frequency of earthquakes in California. Bulletin of the Seismological Society of America, 34(4), 185–188. https://doi.org/10.1785/01184518182

Hardebeck, J. L., Felzer, K. R., & Michael, A. J. (2008). Improved tests reveal that the accelerating moment release hypothesis is statistically insignificant. Journal of Geophysical Research, 113(B8). https://doi.org/10.1029/2007JB005410

Keskar, N. S., & Socher, R. (2017). Improving generalization performance by switching from Adam to SGD.

Raiisi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of Adam and beyond. In International conference on learning representations. Retrieved from https://openreview.net/forum?id=r1yQ07F-RZ

Sangiorgio, M., & Dercole, F. (2020). Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. Chaos, Solitons & Fractals, 139, 110045. https://doi.org/10.1016/j.chaos.2020.110045

Santeramo, R., Withey, S., & Montana, G. (2018). Longitudinal detection of radiological abnormalities with time-modulated LSTM, In T. Stoyanov, Z. Taylor, G. Carneiro, et al. (Eds.), In Deep learning in medical image analysis and multimodal learning for clinical decision support (pp. 326–333). Springer International Publishing. https://doi.org/10.1007/978-3-030-00889-5_37

Tampo, K. F., & Shcherbakov, R. (2012). Seismicity-based earthquake forecasting techniques: Ten years of progress. Tectonophysics, 522–523, 89–121. https://doi.org/10.1016/j.tecto.2011.08.019

Trugman, D. T., Daub, E. G., Guyer, R. A., & Johnson, P. A. (2013). Modeling dynamic triggering of tectonic tremor using a brittle-ductile friction model. Geophysical Research Letters, 40(19), 5075–5079. https://doi.org/10.1002/2012GL05981

Varne, D. J. (1989). Predicting earthquakes by analyzing accelerating precursory seismic activity. Pure and Applied Geophysics, 130(4), 661–686. https://doi.org/10.1007/BF00881603

Wu, H., Mardt, A., Pasquali, L., & Nae, F. (2019). Deep generative Markov state models.

Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., & Cai, D. (2017). What to do next: Modeling user behaviors by time-LSTM. In Proceedings of the 26th international joint conference on artificial intelligence (pp. 3602–3608). AAAI Press. https://doi.org/10.24963/ijcai.2017/504