Deep learning for laboratory earthquake prediction and autoregressive forecasting of fault zone stress

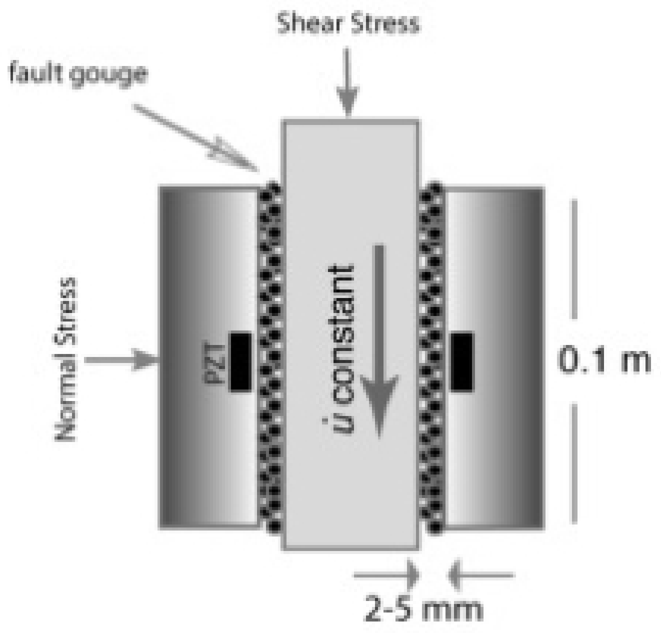
Abstract

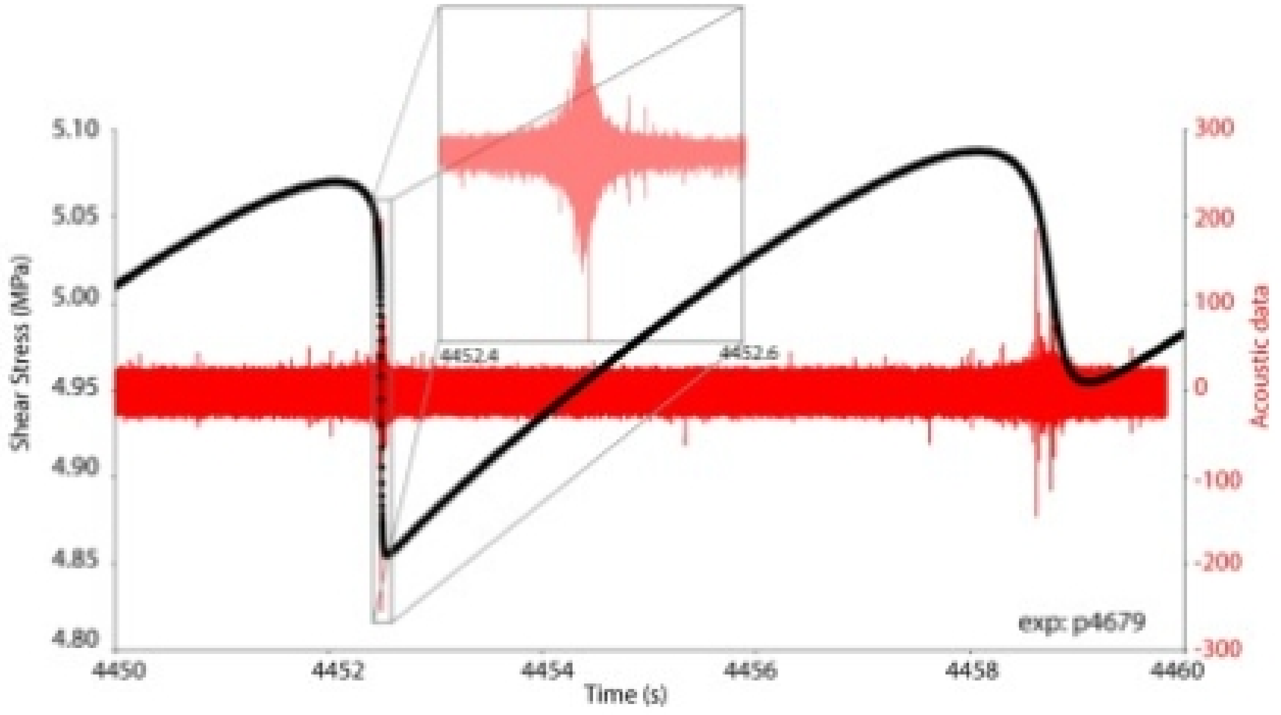
Earthquake forecasting and prediction have long and in some cases sordid histories but recent work has rekindled interest based on advances in early warning, hazard assessment for induced seismicity and successful prediction of laboratory earthquakes. In the lab, frictional stick-slip events provide an analog for earthquakes and the seismic cycle. Labquakes are also ideal targets for machine learning (ML) because they can be produced in long sequences under controlled conditions. Indeed, recent works show that ML can predict several aspects of labquakes using fault zone acoustic emissions (AE).

1. Introduction:The majority of existing ML studies of labquakes use decision-tree-based algorithms and a gradient boosting framework (e.g., XGBoost model). In such studies models are built and errors are minimized by gradient descent methods using multiple learning algorithms that iteratively improve predictive performance beyond that of a single classical learning algorithm. These studies show that it is possible to successfully predict labquakes with reasonable accuracy. The success of the original approaches was based on continuous records of AE events broadcast from the lab fault zones. These AE represent a form of lab microearthquakes. AE are also detected during stable frictional slip. In this case, their origin is less clear, as they could represent microfracture events or micro-instabilities that do not impact the macroscopic strength (and therefore they do not appear as a stress drop).

2. Laboratory earthquake experiments

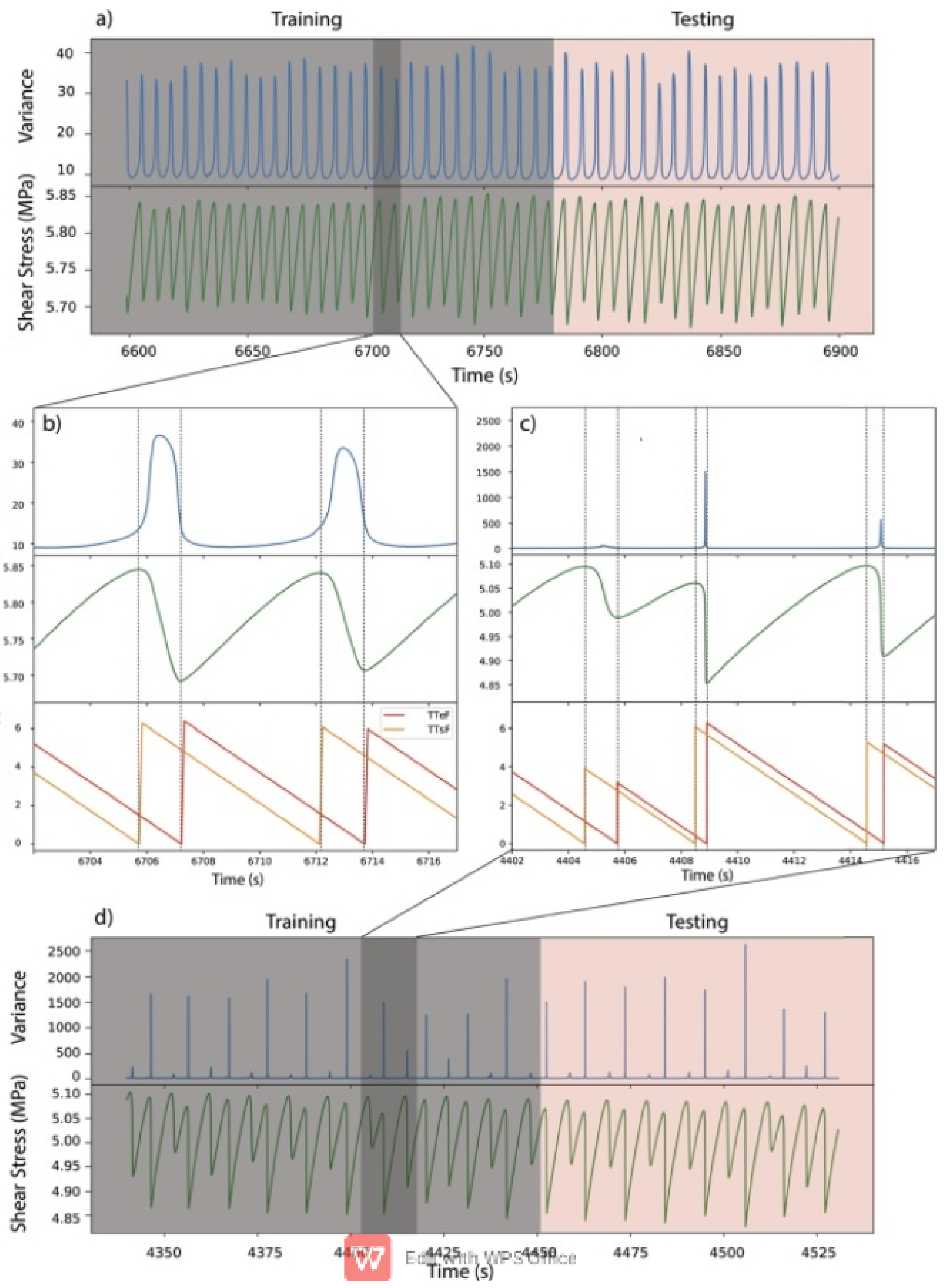
We use data from experiments conducted in the double-direct shear (DDS) configuration with a biaxial deformation machine (see Fig. 1). The DDS experiments consist of two identical fault zones sheared simultaneously between three loading blocks. Two stationary side blocks are supported from below and a central block is driven between them to impose shear.


Shear Stress
fault gouge
Normal Stress
PZT
$\dot{u}$ constant
0.1 m
2-5 mm


Shear Stress (MPa)
5.10
5.05
5.00
4.95
4.90
4.85
4.80
4450
4452
4454
4456
4458
4460
Time (s)
Acoustic data
300
200
100
0
-100
-200
-300
4452.4
4452.6
exp: p4679

3. Prediction and forecasting models

We use DNNs by adopting three of the most well-known sequence modelling approaches: LSTM (Long short-term memory) (Hochreiter and Schmidhuber, 1997), TCN (Temporal Convolutional Network) (Bai et al., 2018) and TF (Transformer Network) (Vaswani et al ., 2017). Fig. 3, Fig. 4 and Appendix 9.1 provide a summary of our modelFig. 3. Schematic model of our combined LSTM and CNN architectures. LSTM scans the input to produce an embedding (lowest level above input). The LSTM layer is followed by three successive CNN layers (see far left) that make the predictions. The input for each layer is the output of the previous layer.


a)
Training
Testing
Variance
Shear Stress (MPa)
Time (s)
b)
c)
Time to Failure (s)
TTeF
TTsF
d)

Prediction

Problem definition

Our goal is to predict the present value of shear stress or TTF, as target variables where t is the present time given current information about AE, as an input variable, and its recent history . We note that the temporal evolution of the AE signal during the lab seismic cycle differs for our range of experiments, in particular during the creep phase preceding labquakes . Previous works have observed that AE statistics (in particular signal amplitude variance and higher-order moments) are highly effective at predicting laboratory earthquakes (Rouet-Leduc et al., 2017). Thus we begin by following this approach. However, whereas previous works focused on data from only one acoustic sensor, we use data from two sensors, one on either side of the DDS assembly . This was done by simply using each data stream as a separate input for the continuous seismic signal. Using multiple sensors will clearly be valuable when these techniques are applied to tectonic faults and/or to event location. We did limited testing of the lab data and did not see dramatic improvement over what one would expect by having additional data to constrain the model loss during training.

Forecasting with LSTM

A key component of an LSTM is the memory cell that regulates information flow using three gates (i.e., Forget gate, Input gate, Output gate). The Forget gate deletes useless information by not releasing it to the next stage. The Input gate regulates new information into the network. The Output gate decides which part of the memory to output from the cell. During the training process, inputs and model weights pass to all gates, which in turn are connected to the self-recurrent memory cell . Our model has 3 stacked layers with size 300, for a total of 1808701 parameters

Prediction dataset split

We train and validate with 70% of the data and test with 30% (Fig. 2). Validation data were chosen randomly from the first 70% of the data and this value (10%) was removed from the training set. Thus the final division was: 63% for training, 7% for validation and 30% for testing.

CODING :

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt

import seaborn as sns  # visualization tool

from subprocess import check_output

#print(check_output(["ls", "../input"]).decode("utf8"))

data = pd.read_csv('AI_phase3dataset.csv')

#data.info()

data.head()

data.tail()

data.columns

data.shape

data.info()
```