UiO **: Department of Geosciences**
University of Oslo

Predicting the Strain Distance to Macroscopic Failure in Rocks under Triaxial Compression Using Machine Learning

Stig-Nicolai Foyn

Master's Thesis, Spring 2021

UNIVERSITAS OSLOENSIS
MDCCCXI

Abstract

Earthquake prediction is a highly important goal in geoscience. In this study we present usage of machine learning to predict distance to failure in rocks, a problem adjacent to earthquake prediction. We use two machine learning techniques, XGBoost and Neural Networks, to predict the strain distance to failure in 15 rock deformation experiments. In these experiments on six different rock types, we use the local strain components calculated with digital volume correlation (DVC) to predict the normalized macroscopic axial strain, i.e., the distance to failure. We use Shapley Additive Explanation (SHAP) to quantify the impact of each feature on our models, and transfer learning between rock types to constrain the generalizability of each model. We combine data from multiple experiments to generate models with increased generalizability. In this study, the importance of dilation in predicting macroscopic failure is about double the importance of the shear strain or contraction. We found that the differences in failure mechanisms between rock types produces lower transfer scores, and that brittle failure in rocks carry differences even for rock types expected to deform with similar mechanisms. The evolution of the strain components is critical to the model performance: all models with systematic evolution towards failure performed with strong or moderately strong correlation between the predicted and observed values. Lastly, we highlight the increase in model performance when the models use data from multiple experiments, rather than individual experiments.

Acknowledgements

Before getting into the details of my thesis I would like to extend my gratitude to everyone who helped supervise my project. My supervisors Jessica Ann McBeck, Francois Renard and John Mark Aiken. I am also thankful to all my supervisors for the time they have spent reviewing my written work, giving great advice and allowing me to improve at academic writing, an effort I greatly appreciate.

I would like to extend my deepest gratitude to Jess for great advice and discussions in addition to her intuition helping me schedule my project during this thesis. Without these qualities I am certain that I would not have been able to learn as much as I have during this study.

Lastly, I want to thank my girlfriend, friends (especially you overwatch squad!), family and again supervisors for all help, time, effort at big or small scales I have received during this relatively difficult time period. During COVID and the lockdown, motivation and maintaining a structured time schedule has been essential, and I would not have been able to do this without all the support.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Earthquake prediction is an elusive yet deeply important goal in geoscience. In laboratory rock deformation experiments, dilatational strain is an important precursor to macroscopic failure (e.g. Brace, Paulding Jr and Scholz 1966; Brace 1978). From these observations, the dilatancy-diffusion model was created to predict crustal earthquakes (e.g. Nur 1972; Scholz, Sykes and Aggarwal 1973). However, natural earthquakes differ from those reproduced in the laboratory in important ways that are not entirely understood (Main et al. 2012). In recent triaxial rock deformation experiments using X-ray tomography, direct observations of the internal microscopic changes in rocks provide unprecedented new knowledge about rock deformation leading to catastrophic failure, such as earthquakes (e.g. Renard et al. 2018; McBeck et al. 2018).

In recent years, studies have begun using machine learning to predict laboratory earthquakes (Rouet-Leduc et al. 2017; Corbi et al. 2019; Hulbert et al. 2019). Rouet-Leduc et al. (2017) predicted the macroscopic shear stress using acoustic emissions. In that study, the authors found that random forests were able to predict the cyclic nature of stress from small changes in the acoustic emission data. Corbi et al. (2019) predicted the timing and size of earthquake analogues in an experiment with a gelatin wedge and subducting plate. They used gradient boosted regression trees with features derived from spatial correlation to predict analogue earthquake cycles of varying magnitudes with relative success. Hulbert et al. (2019) predict earthquakes in the lab using features extracted from acoustic emissions produced in shear fault zone analogues within a biaxial deformation machine. Using extreme gradient boosted regression trees, XGBoost, these authors predicted timing and magnitude of laboratory earthquakes, and found early signals that they correlated with the failure magnitude. In these studies, laboratory earthquakes were used to train and test the models. One advantage of earthquake analogues in the lab is the high quality, high spatial and temporal resolution data that are more difficult to obtain in the field. An advantage of utilizing machine learning techniques is the enhanced ability to recognize patterns in data otherwise noisy to human eyes. Another motivator for the usage of machine learning is the expectation that these models might also be trained to recognize natural earthquakes in the future. Because of this argument, model interpretability is highly important in such machine learning analyses. It is therefore common to extract the most important features in order to interpret the models.

Previously machine learning has been used to classify the proximity to failure in the same rocks as presented in the present study (McBeck et al. 2020a). In addition, XGBoost regression models have been used to predict the stress distance to failure from the fracture networks in rocks (McBeck et al. 2020b). In these studies, features are derived from the 3D X-ray tomography experiments that tie the internal state of rocks to macroscopic failure (e.g., Renard et al. 2018; McBeck et al. 2018). In the present study, our motivation is to explore the physical processes occurring in a rock during deformation by using machine learning. We conduct this investigation by training XGBoost and neural network regression models on data from digital volume correlation (DVC) calculations on 3D X-ray tomography images of six rock types. Using this data set, we explore if machine learning models can predict the strain distance to failure from the evolutions of the local strain components in rocks. We examine if such a model be able to generalize and predict distance to failure across multiple rock types, and if not, can we find ways to improve the model. In addition, by investigating the impact of each feature and comparing the expected similarities in the mechanism of failure in the rock types to the transfer scores of our models , we examine what kind of information can we extract from our models.

This thesis is presented in the following sections: background, methods, results, discussion and conclusions:

Chapter 2 We begin our background section by providing some information central to the process of brittle failure in rocks. Because we utilize data from experiments on multiple rock types, we discuss some of the characteristics and details regarding the mechanisms of failure for each rock type, and then compare them. For the last part of our background section, we introduce the main ideas behind XGBoost, neural networks and transfer learning.

Chapter 3 For the next part of the study, we summarize the main methods utilized, and begin by outlining the methods used in previous studies to conduct the experiments and extract the features used as data in the present study. After this section, we end our method section by briefly summarizing the machine learning methods and providing an overview of the models used during the study.

Chapter 4 We continue by presenting our results from the different machine learning tests conducted, using individual experiments as training data, and then using the experimental data sorted by rock types as training data.

Chapter 5 We then discuss some results in the context of brittle failure and failure mechanisms, and similarities between rock types that we introduced in the background section. We also suggest possible future research based on our evaluation of the current state of the models generated and possible limitations in the present study.

Chapter 6 In the final section we present the main results and the conclusion of our study.

Appendix A For the additional content of this study, we also structure an appendix in three parts. The parts start off with the additional figures, containing extended versions of other figures in this study

Appendix B supplementary information containing some details that were considered to be less integral to this study.

Appendix B a section containing some of the code written for this study, however the full code can be seen at (https://github.com/Anduron/Strain_masters/blob/master/scripts).

CHAPTER 2

Background

We split this background section into three parts. In these three parts we discuss 1) brittle failure in rocks, 2) the experimental rock types and 3) information on the machine learning methods used in this study. These subjects will carry information to supplement the later sections (methods 3, results 4, discussion 5) and improve our understanding of the important details informing the analysis in the present study.

2.1 Brittle failure in rocks

In this section, we discuss some central concepts of the macroscopic failure of rocks. We begin by discussing some relevant information from previous triaxial experiments and local strain components. We then describe experimental and observational work on earthquake precursors. Lastly, we define and discuss the dilatancy diffusion model, due to its relevance in identifying the strain components most indicative of imminent failure.

2.1.1 Triaxial experiments

In a triaxial experiment, all three orthogonal axes are under stress. It is common to call a deformation experiment triaxial if the stress is variable along one axis, and there is a confining pressure or stress applied to two of the three axes. In a “true triaxial” experiment, the stresses applied to these two axes differ from each other. The data used in this study has been extracted from several triaxial experiments. In these experiments, we deformed sandstone, basalt, monzonite, granite, shale and limestone. In these experiments, the stress was increased along one axis, while a confining pressure was maintained along the two other axes. During deformation of these rocks, 3D images were acquired at the European Synchrotron and Radiation Facility (ESRF) at beamline ID19. Figure 2.1 shows the macroscopic deformation response to loading in three characteristic experiments, as well as examples of the local 3D-strain fields early and late in loading. The experimental deformations sandstone, shale and granite along the left side of figure 2.1 end at the final X-ray scan before failure. In these experiments we observe some typical characteristics for rocks in triaxial compression conditions. These characteristics are seen in the shale (within the first 3 DVC calculations) and granite (for the last two DVC calculations) are the linear or quasilinear behavior preceding yielding and then failure in rocks (Figure 2.1). After this the stress suddenly drops as is characteristic of macroscopic failure.

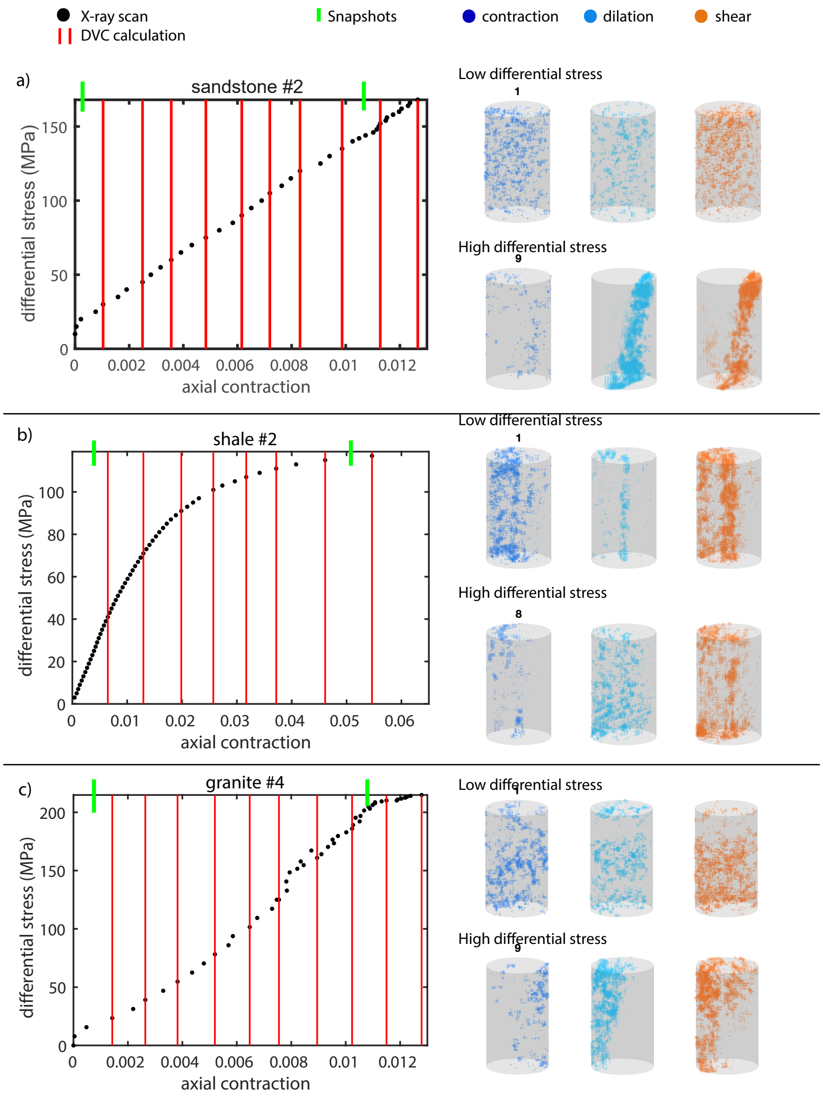


Figure 2.1: a) Macroscopic deformation of sandstone (left), and snapshots of the 3D strain fields at low and high differential stress (right). b) Macroscopic deformation of shale (left), and snapshots of the 3D strain fields at low and high differential stress (right). Figures under c) macroscopic deformation of granite (left), and snapshots of the 3D strain fields at low and high differential stress (right). In the plots of the macroscopic deformation, red lines are the increments of DVC calculation, and black dots show the values when each X-ray scan was acquired. In the 3D strain field, the colored dots show the high magnitudes (90th percentile) of the contractive strains (left, dark blue), dilative strains (middle, light blue) and shear strains (right, orange).

2.1.2 Local strain components

We build our machine learning models using local strain components as the features and the macroscopic axial strain $\varepsilon_{zz}^{M} = \varepsilon_M$, normalized by the axial strain immediately preceding failure, as our target. The macroscopic axial strain, which is parallel to the direction of the maximum principal compression, σ_1, is here parallel to the z-axis. The strain components in the present study were calculated using the positive divergence for dilation, negative divergence for contraction and the magnitude of the curl for shear between each increment in the DVC. In practice, this means that the contraction and dilation is calculated the same way as Renard et al. (2019) and McBeck et al. (2018) while shear strains are calculated in a different way from this previous work.

Volumetric changes in rocks involving the increase or decrease of the volume are called dilation or compaction, respectively. The development of inelastic volumetric deformation in rocks are commonly referred to as dilatancy, or compactancy (negative dilatancy) as is more common for porous rocks. These phenomena have been observed in multiple tests with both porous and crystalline rocks under load, with dilatancy in the crystalline rocks typically attributed to development of microcracks prior to brittle failure (Paterson and Wong 2005) . Dilatancy has been observed in many different experiments with rock deformation prior to macroscopic failure (Paterson and Wong 2005). In crystalline rocks, dilatancy tends to arise due to the growth and opening of microcracks in the rock. Increasing differential stress promotes the growth in length and number of microfractures until they coalesce into a propagating fault zone (e.g., Mjachkin et al. 1975). The onset of dilatancy in crystalline rocks tends to start at stress levels between one third and two thirds of the of the stress at macroscopic fracture (Brace 1978). Porous rocks undergo the onset of dilatancy later in loading than in crystalline rocks. This may be because porous rocks tend to undergo compaction due to pore collapse before the onset of dilatancy. The onset of dilatancy is also considered to be the point at which microfractures begin to propagate in the rock, with an orientation usually parallel to the maximum compressive stress. Dilatancy and the propagation of microfractures will then eventually lead to shear faulting or axial splitting at the macroscopic level (Paterson and Wong 2005).

In porous rocks, we commonly observe compaction-related phenomena. Under hydrostatic stress or in the early stages of deformation, we usually observe pore collapse, resulting in an initial compaction. In addition, other phenomena may be related to compaction; for instance, under deviatoric stress after the aforementioned initial phase of compaction, one may observe shear-enhanced compaction. Shear-enhanced compaction is a process in which collapsing porosity counterbalances dilatancy in porous rocks at stress levels beyond a critical stress where collapse of porosity accelerates more than at hydrostatic stress (Paterson and Wong 2005; Baud, Vajdova and Wong 2006). This does not mean that dilation cannot occur in a rock at the same time as compaction, but measurements of the macroscopic deformation of the sample will indicate the dominance of only one at a given stress level. In porous rocks, it is not uncommon for dilatancy to occur after compaction in later stages of deformation (close to failure). Both dilation and compaction may localize during deformation in either dilation bands or compaction bands. Figure (2.1: a) exemplifies localization of dilation at higher differential stress levels in sandstone. Dilatancy and compaction have also been observed in shear bands that form prior to macroscopic failure (Paterson and Wong 2005). In experimental studies, like in the high temperature (550 degrees Celsius) calcite deformation experiments of Verberne et al. (2017), splitting along narrow shear bands was observed. The findings of their study showed that dilatant velocity-weakening frictional slip, with the capacity of nucleating earthquakes, can be triggered by changes in shear strain rate in areas of the crust normally associated with ductile flow.

2.1.3 Precursors in natural earthquakes

The importance of phenomena that include dilatancy in the brittle curst have long been discussed. Observations by Sibson (1985) of earthquake ruptures along the San Andreas system reveal that these faults may be impeded or terminated due to dilatational jogs. These are segments where the frictional resistance is diminished, facilitating sliding and decreasing the mean compressive stress. Observations of geophysical data preceding the L'Aquila earthquake provide unique insights into earthquake precursors (e.g., Moro et al. 2017). In that study, they utilized InSAR data to investigate the 2009 MW 6.3 L'Aquila earthquake, and observed subsidence in the area after 2006, persisting until the earthquake. The pre-seismic subsidence reached mean values of approximately 1.5 cm, and lowered the groundwater levels in the Gran Sasso carbonate range. These phenomena were hypothesized to result from dilatancy in the rocks. In the pre-seismic phase, shear stresses in the rocks caused volumetric deformation due to the opening of fractures and voids, leading to the migration of water diffusing into the local volume. Dilatancy and diffusion towards the earthquake nucleation area was also inferred from P and S-wave variations that began in 2008. InSAR data was also used in Bignami et al. (2019) to describe precursory events to a

MW 6.0 earthquake in central Italy (Amatrice-Norcia) in 2016. In their study, they found significant subsidence on average about 24 cm, due to hanging wall subsidence. The observed imbalance in volume indicated that a dilated wedge had been generated during the interseismic period. This was followed by the gravitational collapse of the hanging wall due to the loss of strength, and subsequent closing of pre-existing microfractures in the dilated wedge. Lastly, in boreholes located in Hafralækur Iceland, Skelton et al. (2014) found chemical and isotopic changes in the groundwater preceding two MW $>$ 5 earthquakes in 2012. These changes were hypothesized to either be due to the mixing of groundwater components, or exposure of fresh rock surfaces to the groundwater. As reported in their study, both changes can be explained by dilation in the rock, enhancing permeability. These characteristics were observed prior to both earthquakes, meaning that pre-seismic dilation was considered the most likely explanation behind both.

However, there are also reported cases where no precursory dilatational phenomena were observed prior to earthquakes. A study by Bakun et al. (2005) discussing the 2004 MW 6.0 Parkfield earthquake along the San Andreas fault found only very small (nanoscale) changes in strain preceding the earthquake. These small changes are reported as too uncertain to predict the timing of the earthquake, but the possibility to predict earthquakes are discussed as possible, though difficult, in the future. In some cases, later studies using different methods have found more evidence of precursors like dilatancy. We can see this contradiction between the work of Amoruso and Crescentini (2010), which does not find sufficient evidence of precursory changes before the 2009 L'Aquila earthquake, and the previously described study by Moro et al. (2017) that found precursory changes at larger timescales best explained by volumetric changes. Because of inconsistency in the search for earthquake precursors, earthquake prediction may be possible but difficult (e.g., Wyss and Booth 1997; Wyss 2001).

2.1.4 Dilatancy-diffusion model

There have been many approaches to describe failure in rocks. These include the Mohr-Coulomb failure criterion, which is a macroscopic failure criterion that uses the stress acting on a preexisting fault or hypothetical failure plane, the shear strength, or cohesion and angle of internal friction of the rock (e.g., Labuz and Zang 2012). There is also the Griffith theory of brittle failure, which provides a tensile failure criterion for a brittle material where cracks grow during loading (Griffith 1921). Without changes to the original formulation, these models do not consider changes in volume, as a part of the failure criterion (e.g., McBeck, Ben-Zion and Renard 2020). More relevant to the present work is the dilatancy-diffusion model because of its inherent connection to dilation. In the present work, we will use dilation, as well as the shear and contractive strains, to predict the distance to failure, and thus compare the usefulness of each strain component in predicting the timing of failure.

The dilatancy diffusion model is based on experimental observations where dilatancy occurs in rocks prior to macroscopic failure, and earthquake observations indicating that the opening of cracks due to dilatancy may also be important in earthquakes (Nur 1972). This model hypothesizes that dilatancy in rocks cause the opening of cracks in the rock, and then these cracks fill with water (Stage 1). At some point in this process, the rock may become undersaturated due to the rate of dilatancy being higher than the rate of fluid flow into newly opened cracks. At this point the seismic shear velocity ν_p/ν_s will drop (Stage 2). Because of the flow of water from pre-existing cracks into newly formed cracks, the pore pressure drops and the effective stress increases, leading to increased strength in the rock (dilatancy hardening), perhaps inhibiting the increase in dilatancy. Because of the inhibition of dilatancy, the rate of water flow into the cracks will now be higher than dilatancy (Stage 3). The water flow will now compete against the decrease in pore pressure until a point in time where the rock saturates, increasing the pore pressure again. During the dilatant process, tectonic stresses have increased, and the increase in pore pressure will trigger an earthquake similar to how fluid injection would (e.g., Scholz, Sykes and Aggarwal 1973). With this model, other precursory changes can also be explained, such as radon emissions or uplift in large dilatant zones. In the previous section (2.1.3), we discussed how dilatancy related phenomena likely caused subsidence. In Moro et al. (2017) it is hypothesized that water flowing into localized dilatant areas lowered the groundwater and in turn caused subsidence on the surface due to elastic consolidation. Elastic consolidation can be described as the reversible reduction in volume and increase in effective stress arising from the dissipation of pore water pressure. The dilatancy-diffusion model could then hypothetically predict the time of an earthquake based on earthquake precursors or changes in seismic shear velocity (e.g., Scholz, Sykes and Aggarwal 1973).

To exemplify some of the early data supporting the dilatancy-diffusion model, certain precursors expected to be closely related to dilatancy have been observed in Nur (1974). In that study, observations on the Japan, Matsushiro earthquake swarm is detailed and related to qualitative changes predicted by the dilatancy-diffusion model. Among the features they focused on during the earthquake swarm was large symmetrical vertical deformation associated with volumetric expansion. In addition, the horizontal deformation was asymmetric along the fault trace. After the volumetric expansion there was an outflow of water, and then the ground subsided. The observed gravitational variations near the dilatant regions were among the best quantitative confirmations of the dilatancy-diffusion model at the time. In more recent earthquake observations, additional geophysical features indicate that precursory dilatancy related phenomena have occurred, as discussed in section (2.1.3).

As discussed in section (2.1.3), observing unambiguous earthquake precursors is difficult and has led to inconsistent results. Main et al. (2012) discuss some limitations of the dilatancy-diffusion model when predicting earthquakes. The lack of unambiguous observations of precursors and limitations in laboratory experiments in replicating the conditions of rocks deeper in the brittle crust seems to be an essential part of this problem. Many of the limitations in the dilatancy-diffusion model are caused by problems when scaling up features such as heterogeneities and fractures in rocks, coupled with issues regarding reliable simulation of fluid flow through these fracture networks. In addition, earthquakes tend to occur under lower strain rates than feasible in the lab, which may heavily reduce the crack growth rate in rocks, essentially suppressing dilatant strain in rocks (Brantut et al. 2013). This effect may be relevant as the dilatancy-diffusion model predicts dilatancy hardening, which may be reduced by lower strain rates (Main et al. 2012). In addition to these limitations, Main et al. (2012) discuss that many crustal earthquakes are caused by the reactivation of pre-existing fault structures in the crust. Fault structures tend to weaken these areas and are especially important in cases where faults cut through load bearing regions in the lithosphere (Holdsworth, Butler and Roberts 1997).

2.2 Rock types

In the present study we have 15 experiments on six different rock types. Based on observed deformation mechanisms from previous work, we expect pairs of rock types to deform in similar ways. These pairs are sandstone and basalt, monzonite and granite, and shale and limestone. In this section, we describe each rock type and how they deform, and briefly explain these pairwise similarities.

2.2.1 Fontainebleau sandstone

Fontainebleau sandstone is found in an Oligocene age sandstone outside Paris. It is a relatively homogeneous quartz arenite composed of (>99%) quartz. The grain size of sandstone is well sorted with an average size of $0.25mm$, and remains relatively constant over a wide range of porosities (3-30%) (Bourbie and Zinszner 1985).

The sandstone samples that were used in triaxial experiments prior to the present study were found to have about (<1%) iron oxides (Renard et al. 2019) . In accordance with the average grain size of sandstone found previously, the average grain size in these samples were also 0.25 mm. In those experiments, the porosity was measured as $6 \pm 1\%$ by weighting ten dry samples before and after imbibition with water. The porosity was also measured to be in the range 5.5%-7.5% using the three-dimensional data. The pore space of Fontainebleau sandstone was here found to be almost fully connected in 3D (Renard et al. 2019). This property of the pore space is consistent with the findings of Fredrich, Greaves and Martin (1993) where it was found that the pore spaces of sandstone above 5% were fully connected in 3D, but the pore space of samples with <4% porosity had a permeability of close to one order of magnitude lower, and thus was less connected.

In previous studies (e.g. Goodfellow et al. 2015), a true triaxial deformation experiment with acoustic emission monitoring, Fontainebleau sandstone was observed to undergo failure in three phases. The first phase was characterized by initial compaction due to crack closure. In the second phase, the macroscopic dilation of the sample increased, likely due to preexisting microfractures opening in a direction perpendicular to the direction of the minimum stress. The final phase was characterized by damage accumulating until macroscopic failure.

The X-ray microtomography experiments conducted by Renard et al. (2019) showed an acceleration of damage in the sandstone samples. This acceleration of damage was at a slower rate than the rate of a power law damage accumulation. This slower rate may be reasonable to expect for more porous rocks (rather than crystalline rocks) due to diminishing local stress concentration at the tip of microfractures as they reach pores in the rock. Increases in dilatancy preceding macroscopic failure was observed at micro and macroscales for all sandstone experiments. As the samples approached macroscopic failure, they went through different stages, also characterized in Renard et al. (2019). Initially, voids in the rock closed, leading to a macroscopic compaction. This closure was followed by macroscopic dilation due to microfractures developing and pores opening. As the samples approached macroscopic failure, the macroscopic axial strain was observed to increase and transition from quasilinear at intermediate differential stress values to deviate significantly as dilation in the rock increased. This deviation was due to transgranular and intergranular propagation of fractures. Finally, the rocks underwent macroscopic shear failure as microscopic fractures coalesced.

2.2.2 Mount Etna basalt

The most common basalt from Mount Etna is a porphyritic intermediate alkali basalt. This mineralogy means it is a volcanic rock with high alkali contents, and a distinct difference in grain size of crystals. In Heap, Vinciguerra and Meredith (2009) it is described that the Etna Basalt is composed of groundmass (60%), crystals with feldspar (25%), pyroxene (8.5%) and olivine (4%). The basalt we deformed was extracted from the same quarry as the one described in Heap, Vinciguerra and Meredith (2009). A common characteristic found in Etna Basalt is a pre-existing network of microcracks isotopically distributed, due to rapid cooling (Vinciguerra et al. 2005). Isotopically distributed means that there is no preferred orientation of the cracks and that they are relatively uniformly distributed. The porosity of Etna Basalt may vary depending on the cooling rate and other thermal conditions. The rock tends to have numerous micropores and macropores (Zhu et al. 2016). The porosity of the two Mount Etna Basalt cores serving as the basalt data in the present study was 3% (McBeck et al. 2019).

In a study with Mount Etna basalt Zhu et al. (2016) observed the rock undergoing dilatancy and brittle faulting at low effective pressures. Deformation of Etna basalt occurs from microcracks that originate at macropores and propagate approximately parallel to the σ_1 direction. Eventually multiple cracks coalesce to develop a shear band that cuts through the sample. While the cracks were transgranular, the cracks were observed to avoid cutting through phenocrysts. In some of the samples of Etna basalt that were tested, shear-enhanced compaction (also discussed in section 2.1.2) was observed .

In the X-ray tomography experiments with Mount Etna basalt detailed in McBeck et al. (2019) they observed deformation with localized volumetric and shear strains. They observed that high dilative and shear strains localized in the region that contained the largest fracture network preceding the onset of macroscopic failure. In other parts of the sample, they observed localization of contractive strains. The localization of contraction was observed as a precursor to dilation and shear strain localization in the zone that hosted the largest fracture network. In accordance with the results of Zhu et al. (2016), McBeck et al. (2019) observed thin fractures linking pores, and found that pore emanated fractures were the dominant mechanism of brittle faulting within Mount Etna basalt. However instead of the previously observed subparallel cracks seen in Zhu et al. (2016) the samples in experiments conducted by McBeck et al. (2019) were observed to lengthen 30-60 degrees from σ_1. Shear-enhanced compaction was not observed in these experiments.

2.2.3 Monzonite and Westerly granite

Quartz-Monzonite is a rock with similar mineral composition and structure to other granitic rocks, such as Westerly granite. It consists of 17.9% quartz, 12.8% biotite, 57.6% plagioclase (38% anorthite) and 11.7% clinopyroxene in addition to other minor minerals. The rock has a mean grain size of $450\mu m$ and a low initial porosity (Aben et al. 2016) . The Quartz-Monzonite data we use in the present study were attained from rock samples deformed in experiments by Renard et al. (2018). These samples were found to have an initial porosity of $0.78 \pm 0.03\%$. The deformational mechanisms of these rocks are expected to be representative of other crystalline rocks such as granites.

In X-ray tomography experiments preceding the present study, Renard et al. (2018) observed the deformation of monzonite to fail by shear faulting, with many characteristics typical of the mechanical behavior of crystalline rock. At low differential stress, preexisting microfractures in the rock began closing, causing a non-linear strain-stress curve. As the axial load increased, the curve became increasingly linear, indicating elastic deformation of the rock. At the yield stress, the strain-stress relationship deviates from linearity, as microfractures start nucleating and existing ones start growing in volume. At the onset of macroscopic failure, the microfractures are all oriented parallel to the maximum stress and begin coalescing until failure.

Westerly granite is a low porosity crystalline rock. It is described as a fine-grained uniform and almost isotropic rock comprised mainly of quartz and feldspar. The average grain size of the Westerly granite in the present study is different to that of the monzonite. It is in the range of $100-200\mu m$, compared to the larger $450\mu m$ of the monzonite (McBeck et al. 2020a). There have been many studies documenting the deformation, and the mechanisms of brittle failure in Westerly granite and other granites. The general mechanisms of failure in such crystalline rocks include the nucleation, growth, and interaction, leading to the coalescence of microfractures (Tapponnier and Brace 1976, Reches and Lockner 1994, Katz and Reches 2004).

2.2.4 Green River shale

Green River shale is an organic carbon-rich shale with preexisting mechanical weaknesses. These weaknesses include highly anisometric kerogen (parallel to layering), lacustrine marl/silt sediments that form laminations causing strength anisotropy, planar clay-grain fabrics produced by gravitational compaction. These weaknesses in the rock control where fractures may initiate, grow, and coalesce in the rocks (McBeck, Ben-Zion and Renard 2020, Lash and Engelder 2005).

The Green River shale data used in the present study is from an experiment described in McBeck et al. (2018). McBeck et al. (2018) deformed low porosity shale with a fine grain size. In these experiments, the laminations of the shale were set parallel and perpendicular to the direction of the maximum compressive stress. The macroscopic deformation of the shale was observed to initially have a quasilinear stress-strain relationship. Then, a non-linear phase occurred in which increases in differential stress produced increasingly larger increases in axial strain. After this phase, the sample experienced localization of strain onto larger fractures at the onset of a prefailure phase, and then the rock underwent macroscopic failure. The rocks underwent macroscopic axial compaction, however the local strain components in the rocks showed large magnitudes of both dilatational and contractive strains. This local strain accumulation contributed to lamination-parallel compaction bands forming in the rocks with laminations set perpendicular to the maximum compressive stress. In the present analysis, we only analyze the shale experiments with laminations set parallel to the maximum compressive stress.

2.2.5 Anstrude limestone

Anstrude limestone is a porous oolitic limestone. The size of the ooids in Anstrude has been found to be in the range of $100-1000\mu m$ (Lion, Skoczylas and Ledésert 2005). It is a carbonate rock with a near homogeneous mineralogical composition of almost pure 98% calcite, or calcium carbonate. The limestone cores deformed in the present study, deformed in the experiments of (Renard et al. 2017) had an initial porosity ranging from 4-8% measured using the tomography data. In these porous rocks, brittle failure due to contraction in the rock is often observed (Renard et al. 2017, Huang et al. 2019) .

In an X-ray tomography experiment on porous (22%) Leitha limestone from (Huang et al. 2019) localized compaction was observed in the rock.

They observed a double yielding behavior: at the first onset of yielding, the rock underwent pore collapse, and at the second onset, the samples developed discrete compaction bands. Compaction and pore collapse were also observed in Solnhofen limestone with porosity ranging from 2.7% to 3.3% in (Baud, Schubnel and Wong 2000). However, in these experiments, at lower pressures ($\leq 50MPa$) dilatancy as a precursor to brittle faulting was also observed. At higher pressures ($\geq 350MPa$) the samples failed by cataclastic flow associated with shear-enhanced compaction and strain hardening.

In the limestone experiments prior to the present study, Renard et al. 2017 observed failure characteristic of highly porous rocks. At the onset of yield in the rock, the damage at the microscale was controlled by two processes opposing each other. The first process was the opening of microcracks, producing dilatancy in the rock, however the second process was the closing of pores, leading to compaction. While compaction bands typically arise in compaction processes in limestone, this was not observed here. In most of the limestone experiments, the mean of the dilation exceeded the mean of the contraction at some point during deformation (Renard et al. 2017). However, before failure the volume of the limestone sample was observed to decrease due to the closing of pores and higher magnitudes of contraction was observed at the microscales. These results are consistent with the expected pore-collapse mechanism during brittle failure in limestone.

2.2.6 Rock type similarities

Because we use transfer learning in the present study it is important to categorize the similarities in the deformation mechanisms of the rock types we have discussed so far. With similar deformation mechanisms, we expect higher transfer scores from our models. This may help us quantify both similarities in these rocks and understand which information our models learn from statistics on the local strain components of these rocks.

Fontainebleau sandstone and Mount Etna basalt are both porous rocks. While mineralogically different, with sandstone being more homogeneous and basalt being porphyritic, the same mechanisms of failure has been observed in both rock types. It is often observed that fracture nucleation in these rocks happens due to stress concentrations at the edges of pores and/or grains (Renard et al. 2019; Zhu et al. 2016). McBeck et al. (2020a) found quantifiable evidence of the similarities in the mechanisms governing failure in these rocks. Using machine learning to classify distance to failure, and transfer learning to test the models created, they found high transfer scores between sandstone and basalt. In addition to similarities with sandstone, failure mechanisms in Mount Etna basalt has been compared those seen in porous limestones (Zhu et al. 2016). Considering the results in both studies we mainly expect similarities between basalt and sandstone, but also consider the possibility of similarities between basalt and limestone.

Monzonite and Westerly granite are both crystalline rocks comprised of quartz and feldspar in varying quantities (e.g. Kerrick 1969) . There are similarities in the strain accumulation of granites, with some differences between finer (Westerly granite) and coarser grained (Barre granite) granites (Lockner 1998). The similarities between different granites leads us to expect that there are corresponding similarities between Westerly granite and Monzonite. However, results described in McBeck et al. (2020a) indicate that Monzonite and Westerly granite do not have the expected similarities. That study discusses the possibility that different confining stresses (table 1) may have affected the result. In addition, the different grain sizes may lead to fracture impedance in the grain boundaries of granite, which may cause facture networks to become more distributed and strain localization to differ in the two rock types. Because we focus on regression models instead of classification models in the present work, we will still consider the possibility that we may observe similarities in the strain accumulations of these two rock types.

Green River shale and Anstrude limestone are both porous sedimentary rocks associated with compactive failure mechanisms. Compaction due to pore collapse was observed in the limestone experiments from (Renard et al. 2017). Huang et al. (2019) observed this type of compaction as a primary yielding mechanism. Another such mechanism is compaction band development, which was observed in the shale experiments from McBeck et al. (2018), and the limestone experiments of Renard et al. (2017) and for Huang et al. (2019) as a secondary yielding mechanism. Both shale and limestone may also undergo macroscopic dilation under triaxial compression experiments (Baud, Schubnel and Wong 2000). The results described in McBeck et al. (2020a) show that classification models that were created with one of these rocks hosted higher transfer accuracies on either shale or limestone. We additionally see somewhat higher transfer scores between monzonite, shale and limestone. We may expect this trend to be present in our own transfer scores.

One important consideration is the possibility that the transfer scores may be lower when using transfer learning between the regression models in the present study. In McBeck et al. (2020a) the transfer accuracy scores were lower when the number of classes increased from 2 to 4. The metrics used in regression are different from those used in classification, so it will not be possible to directly compare the trends in the transfer learning scores of the present work and McBeck et al. (2020a).

2.3 Machine Learning Techniques

In this section, we present details regarding the main machine learning methods used in the present study. First, we introduce machine learning, and motivate our choice of models. We then present XGBoost where we quickly mention some of the core optimizations in XGBoost but focus on the main algorithmic details. We then investigate the general algorithm for a dense neural network and the ADAM optimizer. Lastly, we present the definition of transfer learning and its relation to our research.

2.3.1 Principles in supervised learning

In this section we will not present every principle of supervised learning, but we will instead reserve our overview for a short introduction to terminology more immediately related to the present study (we base ideas discussed in this section on Hastie, Tibshirani and Friedman (2009)). In the perspective of supervised learning, machine learning is about finding ways to predict an output based on an input. A model takes input data consisting of features or predictors, that can be used to predict the output called the target (also called response). Because machine learning models need to train on data before making predictions and need unseen data to be tested properly, we can split our inputs (X) and outputs (Y) into a training (D_{train}) and testing dataset (D_{test}):

$$D = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}.$$

We generally split this task into two groups based on the target. When the features are used to predict a qualitative response, we are using our model to solve a classification problem. A classification could for instance be separating images of dogs from images of cats. In this example we use the images as our input values or features. When we instead train a model to predict a quantitative or continuous target, we are attempting to solve a regression problem. An example of a regression problem is predicting the weight of a dog based on the amount of food it eats.

When training a model, we want to improve its performance by minimizing the error it makes. We quantify the error made by the model, using the loss function. There are many different loss functions, but often it is convenient to use convex functions as they have global minima. For regression a typical loss function is the mean squared error $\mathrm{MSE} = \frac{1}{n}\Sigma_{i=1}^{n}\left(Y_i - \widehat{Y_i}\right)^2$, where $\widehat{Y_i}$ is the model prediction.

When quantifying the performance of a model we use the training data as it provides unseen data for the model to be tested on, instead of the previously seen training data. When testing the model performance, we may see many different issues with our models. Usually these include that the model does not predict new data as well as seen data due to high variance, or that the model is too general to really account for the patterns seen in the data due to high bias. These phenomena are called overfitting, and underfitting but belong to a bigger problem in machine learning, which is the bias-variance tradeoff. The hypothesis is that there is an underlying function f(x) that describes the target but using real world data leads to this function being perturbed by some noise $y = f(x) + \epsilon$. We desire a model that can generalize well on new data and represent the underlying mechanism since the noise will vary between the training and testing data, but this will come at the cost of some of the model performance. This is with moderation an acceptable tradeoff since we prefer that the model does not represent the noise in our data, so long as the model still represents the underlying function. Overfitting and attaining a model flexible enough to represent underlying mechanisms in our data is what motivates the use of regularization techniques when training our model. There are many techniques and we will not be discussing them in detail apart from the ones presented in the XGBoost section (2.3.2), but the general intention is to inhibit the amount a model is allowed to learn just enough to avoid picking up noise.

2.3.2 XGBoost

XGBoost or extreme gradient boosting is a tree boosting system that has gained wide use for many machine learning problems. Much of this section is based on the XGBoost paper by Chen and Guestrin (2016), and supplemented by (XGBoost Documentation). Gradient boosting is a technique where one additively combines weak learners into a single strong learner. Gradient tree boosting generally uses an ensemble of decision trees, referred to as classification and regression trees (CART), as the model (e.g. Friedman, Hastie, Tibshirani et al. 2000). XGBoost is a highly optimized variant of gradient boosting that is scalable and has been used as a tool in many machine learning applications (Chen and Guestrin 2016). XGBoost has a regularized learning objective which consists of a convex loss function $l\left(\widehat{y}_i, y_i\right)$, for example the mean squared error, and a regularization term Ω given by:

$$\mathcal{L}\left(\phi\right) = \Sigma_i l\left(\widehat{y}_i, y_i\right) + \Sigma_k^K \Omega\left(f_k\right), \tag{2.1}$$

where $f_k \in \mathcal{F}$ is a function in the function space of the CART models. To aid our understanding of the regularization in XGBoost we define the function space as:

$$\mathcal{F} = f\left(x\right) = w_{q(x)}\left(q : R^m \rightarrow T, w \in R^T\right),$$

where w is the vector of scores on leaves, q is a function assigning each datapoint to the corresponding leaf and T is the number of leaves.

To understand how XGBoost works we need to define this function and how the additive training works. The additive training starts with a naïve model prediction at step $k = 0$ for example $\hat{y}_i^0 = 0$, then the prediction of the model at step k, and at the i-th instance, is given by:

$$\hat{y}_i^{(k)} = \Sigma_j^{(k)} f_j\left(x_i\right) = \hat{y}_i^{(k-1)} + f_k\left(x_i\right).$$

This means that at each step we add a tree $f_k\left(x_i\right)$ that minimizes our objective given by:

$$\mathcal{L}^{(k)}\left(\phi\right) = \Sigma_i l\left(y_i, \hat{y}_i^{(k-1)} + f_k\left(x_i\right)\right) + \Omega\left(f_k\right). \tag{2.2}$$

To support loss functions other than the MSE, for example the log loss, XGBoost now uses the Taylor expansion of the loss function up to the second order instead of the first. This means that XGBoost can be viewed as a newton tree boosting method, and the support for multiple loss functions increases the adaptability of the method. The commonly used gradient boosting uses the first order Taylor approximation to minimize the loss function. Newton boosting similarly uses the Taylor approximation of the loss function, but instead of using the first order expansion, it uses the second order expansion. In some ways gradient boosting can be viewed as a special case of Newton boosting , except that gradient boosting readjusts leaf weights after approximating tree structure . For the mean square error loss function the approximations used for gradient boosting and Newton boosting are equivalent to each other (Sigrist 2021). The Taylor expanded loss function at the t-th iteration is given by:

$$\mathcal{L}^{(t)} \approx \Sigma_i^n \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right) \right] + \Omega\left(f_t\right) + constant. \tag{2.3}$$

Where the first and second gradient of the loss function here is defined by $g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$. We can now look at the regularization term which will help penalize the complexity of the tree. The term XGBoost uses as a measure for complexity is defined as:

$$\Omega\left(f\right) = \gamma T + \frac{1}{2} \lambda \Sigma_{j=1}^T w_j^2. \tag{2.4}$$

With this we can remove the constant and look only at the objective of the new tree we are generating at the t-th iteration:

$$\tilde{\mathcal{L}}^{(t)} = \Sigma_{i=1}^n \left[g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right) \right] + \gamma T + \frac{1}{2} \lambda \Sigma_{j=1}^T w_j^2.$$

We can now use the definition of the function $f(x)$ and define $I_j = \{i|q(x_i) = j\}$ as the index set of datapoints assigned to the j-th leaf. In addition we write $\Sigma_{i \in I_j} g_i = G_j$ and $\Sigma_{i \in I_j} h_i = H_j$ to rewrite this objective into:

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \Sigma_{j=1}^T \left[\Sigma_{i \in I_j} g_i w_j + \frac{1}{2} \left(\Sigma_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T, \\ \tilde{\mathcal{L}}^{(t)} &= \Sigma_{j=1}^T \left[G_j w_j + \frac{1}{2} \left(H_j + \lambda \right) w_j^2 \right] + \gamma T. \end{aligned}$$

With this we can find the optimal leaf weights by minimizing w_j because they are independent of each other for a set $q(x)$ structure by setting the derivative to 0. This will give us a ratio between the first and second derivative, we interpret it as a step to minimize the loss function in the direction of G_j scaled by H_j. Thus, the optimal value of the leaf weights are given by:

$$w_j^* = -\frac{G_j}{Hj + \lambda}, \tag{2.5}$$

$$\mathcal{L}^{*(t)} = -\frac{1}{2}\Sigma_{j=1}^{T}\left[\frac{G_j^2}{H_j+\lambda}\right] + \gamma T.$$

This score is used to evaluate the quality of a tree structure q, and supports different objective functions. We can now review how XGBoost finds splits that optimize the loss reduction given that we can assign index sets I_L, I_R and $I = I_L \cup \ I_R$ to the left and right split:

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\Sigma_{i\in I_L} g_i)^2}{\Sigma_{i\in I_L} h_i + \lambda} + \frac{(\Sigma_{i\in I_R} g_i)^2}{\Sigma_{i\in I_R} h_i + \lambda} \ - \ \frac{(\Sigma_{i\in I} g_i)^2}{\Sigma_{i\in I} h_i + \lambda}\right] - \gamma,$$

or

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} \ - \ \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] \ - \ \gamma. \tag{2.6}$$

The last form is intuitive in the sense that we can directly see how XGBoost will be able to prune splits that does not minimize the loss function. Essentially we evaluate the score on the new leaf to the left, then do the same to the one on the right and then we evaluate the score of the original leaf, finally we regularize trees with low gain. The regularization γ parameter is a cutoff value for pruning the trees. Given that the score is negative we will prune the new leaves and leave the tree structure as it was before the split. The best split is found by using the exact greedy algorithm which enumerates over all the possible splits on all the features in sorted order. A benefit to XGBoost over some gradient boosting methods are the tree structure allowing for deeper trees which for some problems can increase the flexibility of the method. However, since XGBoost also allow for shallow trees or stumps the exact greedy algorithm combined with the regularization γ is useful because we can find an optimal set of trees for our model and prune the trees that are do not fill this criterion. The exact greedy algorithm is computationally heavy, and sometimes not possible, so XGBoost also has an a local and a global approximate algorithm. These algorithms generally propose splits based on percentiles of feature distribution. The local and global algorithm differ in when the proposal is given. The global algorithm proposes splits initially and keeps using the same splits, whereas the local algorithm makes new proposals after each split (Chen and Guestrin 2016).

To complete this overview of the XGBoost algorithm we present a short overview of some of the last essential techniques. XGBoost has a shrinkage factor (learning rate) η that scales the contribution of each tree and leaves more room for future trees to improve the model. In addition, XGBoost has column subsampling which is a technique to allow each new tree to only see a subset of the features. These two features help further regularize XGBoost and reduce overfitting. XGBoost also has some key optimizations, that improves scalability and time spent per tree. The algorithm is sparsity aware, meaning that it can deal with missing data, frequent zero entries and structures like one-hot encoded data. It can also handle data that is too large for memory by splitting it into blocks and storing it on the main disc. For the exact greedy algorithm everything is stored as one block but for the approximate algorithm XGBoost can use multiple blocks. In addition, finding splits for each column can be parallelized. With this we know some of the most important details to the XGBoost algorithm. In the present study we will among other reasons employ this method due to the success in previous works by McBeck et al. (2020a).

2.3.3 Neural Networks

Neural networks are a class of machine learning methods with many variations. Some popular variants of this include convolutional neural networks (CNN), generally used for image analysis and recurrent neural networks for tasks such as speech recognition. However, for the problem presented in this work we use a densely connected feed forward neural network for regression, a type of deep neural networks. Much of the work in this section is supplemented by Nielsen (2015).

Neural networks are structured in layers, the layers are structured into the input layer, one or more hidden layers, the amount is generally selected by the user, and an output layer. The hidden layers consist of "neurons", in a dense feed forward neural network a neuron in one layer is connected to all neurons in the next layer. Each neuron has weights and biases that dictate their influence on the neurons in the next layer. The biases restrict the minimum weight needed for a neuron to influence the next layer.

Weights in one layer affect the next layer by going through that layer's activation function. Activation functions are often non-linear functions like the commonly used sigmoid $1/\left(1+e^{x}\right)$ activation function. When we let an untrained network complete the first forward pass, we get an output which usually will just be a random guess. For the network to start "learning" we use gradient decent on the loss function of the network. With this information we can adjust the weights and biases in the network through backward propagation. Which lets us adjust the weights and biases in the network based on information from the gradient decent. The process of feed forward,

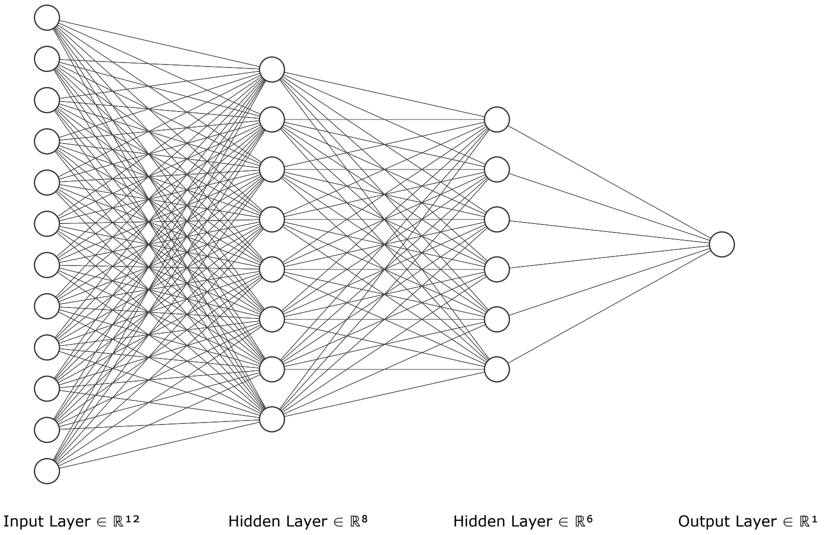


Figure 2.2: Schematic showing a densely connected neural network architecture. Each row represents a layer in the network, each node represents a neuron, and as visible, each neuron has a set of connections to the neurons in the next layer. The first layer is the input layer, the middle two are the hidden layers, and the last is the output layer (schematic from LeNail 2019).

finding the error and adjusting weights through backward propagation is repeated under the assumption that we can iteratively minimize the loss through each repetition.

The activation function in each layer takes a weighted sum (or linear combination) $z_i^{(l)}$ of the neurons in the previous layer. The weighted sum is given by $z_i^{(0)} = \Sigma_j^M w_{ij}^{(0)} x_j + b_j^{(0)}$ in the input layer and $z_i^{(l)} = \Sigma_j^M w_{ij}^{(l)} a_j^{(l-1)} + b_j^{(l)}$ in layers after the input layer. Here (l) represents the layer, w represents the weights and b represents the biases. The activation is here noted as $a_j^{(l)} = f\left(z_j^{(l)}\right)$ where f is the activation function contributing to the weighted sum in the layers after. Usually all neurons in one layer use the same activation function but from layer to layer there may be different activation functions. Using different activation functions can serve different purposes, but generally we need specific activation functions for the output. In a classification the activation function often used in the last hidden layer is the cross-entropy activation function, but for regression we often just see the identity function $f(x) = x$ used. In the other hidden layers, we typically see non-linear functions used. This is because neural networks with non-linear activation functions have been proven to be universal approximators (Hornik, Stinchcombe and White 1989).

Computing the activation function of the weighted sum for each subsequent layer is called a feed forward pass. A feed forward pass ends after we reach the output layer and we can start quantifying the error made through the loss function. After a feed forward pass, we use the result to get a correction from our optimizer (gradient decent) algorithm, this correction represents a step along the gradient of the loss function that we can use to improve the model. For the network performance to improve we utilize the information from our optimizer to adjust all weights and biases throughout the network. This is called backwards propagation because we start with the correction to the output layer and then compute the corrections layer by layer backwards through the network. The correction in the output layer is computed by:

$$\delta_j^{(L)} = f'\left(z_j^{(L)}\right) \frac{\partial \mathcal{L}}{\partial a_j^{(L)}},$$

where $\mathcal{L}$ is the loss function. For regression we use the identity function as activation function, with the identity derivative being given by $f\left(z_j^{(L)}\right) = z_j^{(L)}$ and $f'\left(z_j^{(L)}\right) = 1$. With the mean square error (divided by 2) $\mathcal{L}\left(a_i^{(L)}\right) = \frac{1}{2n}\Sigma_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2$ as the loss function this derivative should be $\frac{\partial \mathcal{L}\left(a_i^{(L)}\right)}{\partial a_i^{(L)}} = a_i^{(L)} - y_i$. With this we see that the error for the output layer for the MSE loss function is given by:

$$\delta_i^{(L)} = a_i^{(L)} - y_i.$$

With this output error we can go through each layer in the neural network backwards and compute the error and correction to the gradients. For the layers l=L-1, L-2, ..., 1 we compute the corrections as:

$$\delta_j^{(l)} = \Sigma_k \delta_k^{(l+1)} w_{kj}^{(l+1)} f'\left(z_j^{(l)}\right).$$

And with these error corrections we update the weights according to the Adam optimizer introduced in Kingma and Ba (2014). Adam estimates updates using a running average of the first and second order momentum. This optimizer is one of the more popular optimizers currently used, being available in large scale machine learning applications such as tensorflow and pytorch. In addition, it is also the default optimizer in scikit learn (Pedregosa et al. 2011). In addition to being simple to implement, Adam comes with some advantages. These advantages improve the convergence speed and increases robustness. One such advantage is momentum which may increase the robustness against getting trapped in local minima. This may improve the speed at which the optimizer converges. Another advantage is that each step is scaled by the steepness of the gradient. This improves convergence for loss functions with “flat” features, such as saddle points (Kingma and Ba 2014). Adam’s parameter update is described by this set of equations:

Algorithm 1 The Adam parameter update (Kingma and Ba 2014):

```
$m_0 \leftarrow 0$;
$v_0 \leftarrow 0$;
$t \leftarrow 0$;
while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta \mathcal{L}(\theta_{t-1})$ (Gets gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1)\, g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2$ (Update biased second raw moment estimate)
    $\hat{m}_t = \frac{m_t}{1-\beta_1^{(t)}}$ (Compute bias-corrected first moment estimate)
    $\hat{v}_t = \frac{v_t}{1-\beta_2^{(t)}}$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$
end while
```

Where β_1 and β_2 are decay factors, typically set to 0.9 and 0.999 respectively. For the other parameters t is the training iteration, α is the learning rate and ϵ is some small number to prevent zero division. Note that $\nabla_\theta \mathcal{L} = a_k^{(l-1)} \delta_j^{(l)}$ for the weights and $\nabla_\theta \mathcal{L} = \delta_j^{(l)}$ for the biases (Kingma and Ba 2014).

With this we know the most important algorithmic details to understand a dense feed forward neural network. To summarize we have a structure consisting of neurons split into layers, each neuron can send signals to the neurons in the next layer. Neural networks learn by quantifying the error they make during training, with an optimizer like Adam. The weights and biases of the neurons are then adjusted with backprop, based on the correction found by the optimizer. Because of success in many machine learning problems and the popularity of deep learning they are widely used.

2.3.4 Transfer Learning

In machine learning it is common to split data into training and testing data drawn from the same feature space with the same probability distribution. However, problems like limited data has been a motivator to make models that can adapt previously learned knowledge on new tasks. This is the idea behind transfer learning, a technique where a model is taught information in a domain different from the target domain with the aim to improve the performance.

In machine learning it is common to split data into training and testing data drawn from the same feature space with the same probability distribution. However, problems like limited data has been a motivator to make models that can adapt previously learned knowledge on new tasks. This is the idea behind transfer learning, a technique where a model is taught information in a domain different from the target domain with the aim to improve the performance. More formally defined, as seen in Weiss, Khoshgoftaar and Wang (2016) and (Pan and Yang 2009), we start by defining a domain $\mathcal{D}$ in two parts, a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in \mathcal{X}$. With this the domain is defined as $\mathcal{D} = \{\mathcal{X},\ \ P(X)\}$. To define the task $\mathcal{T}$, we look at the components of the task, the label space $\mathcal{Y}$, which is the target, and a predictive function $f(\cdot)$, which is learned from the feature vector and label pairs $\{y_i, x_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Then the task is defined as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$. To exemplify in a way relevant for our study, $f(x)$ is our model predicting the distance to failure y given the features x. The last pieces to define before transfer learning is the source domain with a corresponding source task, and a target domain with a corresponding target task. The source domain data is defined as $D_S = \{(x_{S1}, y_{S1}), \dots, (x_{Sn}, y_{Sn})\}$, where $x_{Si} \in \mathcal{X}_S$ and $y_{Si} \in \mathcal{Y}_S$ is the i-th instance of the data and corresponding label of D_S respectively. The target domain data is defined as $D_T = \{(x_{T1}, y_{T1}), \dots, (x_{Tn}, y_{Tn})\}$, where $x_{Ti} \in \mathcal{X}_T$ and $y_{Ti} \in \mathcal{Y}_T$ is the i-th instance of the data and corresponding label of D_T respectively. The source task is notated as $\mathcal{T}_\mathcal{S}$, the target task is notated as $\mathcal{T}_\mathcal{T}$, the source predictive function and target predictive function are noted as $f_S(\cdot)$ and $f_T(\cdot)$ respectively.

Definition 1 *Given a source domain $\mathcal{D}_S$ with a corresponding source task $\mathcal{T}_\mathcal{S}$, and a target domain $\mathcal{D}_T$ with a corresponding target task $\mathcal{T}_\mathcal{T}$, where either $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_\mathcal{S} \neq \mathcal{T}_\mathcal{T}$. Transfer learning is then defined as the process of using the information in domain $\mathcal{D}_S$ and $\mathcal{T}_\mathcal{S}$ with the aim to improve the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$.*

This definition comes with some implications, based on the criterion for transfer learning. These implications are detailed in Weiss, Khoshgoftaar and Wang (2016), Pan and Yang (2009) and Zhuang et al. (2020). This criterion $\mathcal{D}_S \neq \mathcal{D}_T$ may be true for either $\mathcal{X}_\mathcal{S} \neq \mathcal{X}_\mathcal{T}$ or $P(X_S) \neq P(X_T)$, or both at the same time. For the first part of the criterion, we have two categories of transfer learning: 1) The case where the criterion is $\mathcal{X}_\mathcal{S} \neq \mathcal{X}_\mathcal{T}$, is called heterogeneous transfer learning and 2) for the case where the criterion is $\mathcal{X}_\mathcal{S} = \mathcal{X}_\mathcal{T}$, the feature spaces are the same and we have homogeneous transfer learning. The second part of the criterion addresses situations where the marginal probability distributions of the features are different. The second criterion $\mathcal{T}_\mathcal{S} \neq \mathcal{T}_\mathcal{T}$ also consist of two parts, and can be rewritten as $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$. The first part of this criterion it is possible that $\mathcal{Y}_\mathcal{S} \neq \mathcal{Y}_\mathcal{T}$, means that there can be a mismatch between the class space of the task. The second part of the task criterion means that it is possible for $P(Y_S|X_S) \neq P(Y_T|X_T)$, meaning that a model trained in the source domain performs differently on the source task and target task. This is usually in the case where the data in the source and the target domains are unbalanced.

The features in our own study are generated by digital volume correlation and we have the same set of features for all the rock types. The numerical values of those features are different from experiment to experiment and even more so from rock type to rock type. This means that the transfer learning conducted in our study is categorized as homogeneous, however the domains of each experiment are still different because of different marginal probability distributions. For the second criterion we have the same kind of target space, since the target for all our models is to find the value of the macroscopic axial strain, given a set of features. With this we have the definition of transfer learning and some detail on how it relates to our research. Transfer learning is an expansive topic, with many techniques, some of which may help us improve the work presented in the present study, in the future.

CHAPTER 3

Methods

In this section we discuss the methods used in our research. In the initial sections we provide a summary to the most important details on the methods used in previous studies, directly related to the data, and methods used in this study. We then provide an overview of the machine learning methods that was central to this study, including XGBoost, neural networks and the usage of SHAP and transfer learning. We finalize this section by providing and overview of the models generated during the present study.

3.1 Experimental conditions

Data used in this analysis come from experiments conducted on rocks in the HADES triaxial deformation apparatus. This X-ray transparent deformation apparatus installed on the microtomography beamline ID19 at the European Synchrotron Radiation Facility was used to acquire 3D tomograms, at intervals of 2 minutes, of rocks at in situ stress conditions of the upper crust (Renard et al. 2016). We initially applied isotropic confining pressure ranging from $5-35MPa$ (McBeck et al. 2020a) to the rock samples inside the deformation apparatus. During the experiments, we increased the axial stress incrementally until the rock underwent macroscopic failure. The size of each stress step was dependent on both rock type and proximity to failure, with the axial stress step size ranging from $0.5-5MPa$ (McBeck et al. 2020a). As the rock approached macroscopic failure, we decreased the stress steps to more closely monitor the deformation of the rock. Poor scan quality could indicate brittle deformation as the loading is held constant (McBeck, Ben-Zion and Renard 2020). However, the high scan quality of the tomograms used in this analysis indicates that the influence of brittle creep and deformation during scan acquisition is negligible.

Rock Type	Experiment Code	Confining Stress (MPa)	Sample Diameter (mm)	Number of X-ray Tomograms
Fontainebleau Sandstone	FBL01	20	5	184
	FBL02	10	5	47
Mount Etna Basalt	ETNA01	10	4	32
	ETNA02	10	4	36
Monzonite	MONZ04	35	4	65
	MONZ05	25	4	80
Westerly Granite	WG01	5	4	43
	WG02	5	4	30
	WG04	10	4	66
Green River Shale	GRS02	20	5	60
	GRS03	20	5	61
Anstrude Limestone	ANS02	20	5	41
	ANS03	5	5	35
	ANS04	20	5	26
	ANS05	5	5	26

Table 3.1: The table show experiment codes, corresponding rock types, sample diameter and number of X-ray tomograms of all 15 experiments, on which we conduct our analysis, in this study. The experiments have been documented by previous papers (McBeck et al. 2018, Renard et al. 2019, McBeck et al. 2019), and the X-ray tomograms are publicly available (Renard 2017; Renard 2018c; Renard 2018a; Renard and McBeck 2018; Renard 2018b).

3.2 Feature Extraction

We used the 3D tomograms from the experiments to extract features for the machine learning analysis. This step was done using digital volume correlation (DVC) with the code TomoWarp2 (Tudisco et al. 2017). This code finds the local displacement vector in a set of subvolumes inside a tomogram by maximizing the correlation between corresponding subvolumes in sequential pairs of tomograms (Tudisco et al. 2017, Renard et al. 2018). Thus, the DVC provides the incremental displacement done between each sequential pair. In order to perform DVC, each experiment was separated into about ten intervals of approximately equal increments of the cumulative macroscopic axial strain (e.g., McBeck et al. 2018; Renard et al. 2018).

From these displacement fields, we calculate measurements of three local strain components: the contractive, dilatational and shear strains (McBeck et al. 2020a). These three strain components are calculated using the negative divergence, positive divergence and the magnitude of the curl of the displacement field. To extract features of the strain field, we split these strain fields into subvolumes with two spatial resolutions 0.5mm (i.e., low-resolution) and 0.2mm (high-resolution). The node spacing, and thus spatial resolution, of the DVC fields is 20 voxels or 0.13mm. Thus, there are about 4^3 measurements for each subvolume in the low resolution data and $(3/2)^3$ per subvolume for the high resolution data. To derive the features, we also calculate nine statistics of the three strain populations (dilation, contraction, shear) in each subvolume: the 90^{th}, 75^{th}, 50^{th}, 25^{th} and 10^{th} percentile, the mean, standard deviation, the sum of the strain population and the number of measurements within a subvolume. The number of values is the total number of measurements within a subvolume, and does not change for the shear strain because each subvolume contains the same number of measurements, and we do not consider the positive and negative curl populations separately. For the contractive and dilatational components, it tends to decrease and increase respectively close to failure in these experiments (McBeck, Ben-Zion and Renard 2020). In addition, it tends to accelerate as a sample approaches macroscopic failure (McBeck et al. 2020a).

From the tomograms, we calculate the macroscopic axial, radial and volumetric strain. In the tomograms, the distance between the two pistons, at two points normal to the piston faces, was used to calculate macroscopic axial strain, which is the target for our machine learning methods in the present study.

3.3 Machine Learning Methods

In this analysis, we develop machine learning models to predict the macroscopic axial strain, ε_M. Rather than predicting the proximity to failure using classification (McBeck et al. 2020a), we predict this proximity using regression: a more difficult problem than our previous analysis. We use two different machine learning methods that are both used in a large variety of problems: XGBoost (extreme gradient boosting) and neural networks. XGBoost is a method that utilizes gradient tree boosting, an algorithm that combines an ensemble of decision trees or "weak learners" into a single strong learner. The decision trees are made sequentially and represents an iterative step along the gradient of the loss function. We use the Mean Squared Error, so the gradient of the loss function with respect to our ensemble would

Feature Statistic	Local Strain Component		
	Contraction	Dilation	Shear
90^{th} percentile	#1: dn_p90	#10: dp_p90	#19: cur_p90
75^th percentile	#2: dn_p75	#11: dp_p75	#20: cur_p75
50^th percentile	#3: dn_p50	#12: dp_p50	#21: cur_p50
mean	#4: dn_mean	#13: dp_mean	#22: cur_mean
25^th percentile	#5: dn_p25	#14: dp_p25	#23: cur_p25
10^th percentile	#6: dn_p10	#15: dp_p10	#24: cur_p10
standard deviation	#7: dn_std	#16: dp_std	#25: cur_std
number of values within subvolume	#8: dn_num	#17: dp_num	#26: cur_num
sum	#9: dn_sum	#18: dp_sum	#27: cur_sum

Table 3.2: This table shows the feature number and the nametag of a feature in the code input files. The nametag convention informs on statistic (e.g. p50 being 50th percentile), and whether the strain component is found using the negative divergence (dn), positive divergence (dp) or curl (cur). Note that in the files columns are preceded by target data (the second column "ep" is the strain data used for this analysis) and positional data, meaning that the column number of a feature is the feature number in this table plus seven for the preceding columns.

commonly be the residuals. XGBoost uses the second order expansion of the loss function, and is also combined with regularization (and multiple optimizations) to reduce overfitting and improve generalizability (Chen and Guestrin 2016). To find the best set of hyperparameters for the XGBoost models, we perform a grid search over our hyperparameter space.

Deep neural networks (DNNs) are a type of neural networks with multiple layers between input and output. Neural networks learn by minimizing loss using gradient decent and adjust weights and biases in each layer through backwards propagation (backprop). For our analysis, we use the Scikit learn implementation of neural networks. This implementation comes with built in (L2) regularization to reduce overfitting. After testing activation functions, we use the hyperbolic tangent function for the neural networks because of improved results over ReLU (rectified linear unit) and sigmoid function, which has been used in early neural networks. We use a fully connected network with two hidden layers of 128 neurons for the first hidden layer, and 64 neurons for the second.

Extracting the features from the DVC data provides datasets with 34 columns of data, including 27 features and seven measures of time, including the macroscopic axial strain and differential stress. These features arise from the statistics of our three different local strain components: contraction, dilation and shear strain. In addition to these 27 features, the dataset contains the macroscopic axial strain evolution, which is the measurement of time that we develop the machine learning models to predict, i.e., target. Because the DVC was split into around 10 intervals, we have this number of unique axial strain values. We scale the features of our data using the Scikit Learn built-in RobustScaler function. The target, ε_M, is normalized so that each strain evolution starts at the normalized macroscopic axial strain, $\widehat{\varepsilon_M} = 0$ and ends at $\widehat{\varepsilon_M} = 1$. Then, the data is split into 80% training data and 20% testing data. The split of the data is random, but on average we get sets of features and targets representing each DVC macroscopic axial strain increment in both the training and testing data. To minimize autocorrelation between datapoints when splitting into training and testing data the train_test_split function in scikit learn splits the data row by row to preserve the independence between rows. Each row is then shuffled so that potential autocorrelation between sequential rows will on average affect our models less. Finally, the testing data, is still shuffled to preserve this effect while testing. When plotting the testing data, we use a sorting algorithm allowed to look at the metadata of each row to reconstruct the observed and predicted data into a sequential dataset. Even accounting for these steps taken against autocorrelation, there may still be autocorrelation between rows of data that affect our results. Therefore, our transfer learning tests are very important in this study, because they provide tests without this type of autocorrelation. Note that even though autocorrelation may affect the results, lower transfer scores than train test scores within the same individual experiment are not necessarily only because of autocorrelation.

3.3.1 Shapley Additive Explanation (SHAP)

Shapley Additive Explanation (SHAP) values measure the impact of individual features on the model predictions. SHAP values are calculated using a model with a feature S and comparing the result to multiple reference models without the feature S. The resulting SHAP feature importance value is calculated by averaging over all the possible orderings of models (Lundberg and Lee 2017). For this analysis we report the average SHAP value across all samples to get the average importance for each feature.

To examine trends in feature importance, we build a metric that considers the model performance. We use the normalized, average SHAP value, and scale them by the model performance. This step provides 27 scaled SHAP values, one value for each feature, for each of the models. Finally, we sum these values for each model, thus $I = \sum_{i=1}^{N} R_i^2 |\widehat{SHAP}|_i$ for the importance (where N is the number of models).

3.3.2 Generalization

Ideally, we desire to create a machine learning model that generalizes well on multiple rocks of the same type, and different rock types (Table 3.3, Table 3.4). We consider a model to generalize well if it performs well on data that it has not encountered during training. To test our generalizability, we use transfer learning , a technique that involves training a model on one dataset and testing it on another. We train models on a single dataset, and then test each model on all the other datasets (Table 3.4). When using transfer learning our models must make predictions on previously unseen data. We expect that this step will tend to lower the performance of the model. To improve the model performance and generalizability, we can train the model on multiple experiments (Table 3.3). We have multiple experiments for each rock type (Table 3.1). Thus, we couple experiments together by rock type when we develop models with multiple datasets. The data from each experiment is scaled individually before being coupled together. We train our models on 80% of the data within one or two rock types and test the model on all unseen data in each rock type.

3.3.3 Overview Over Models

The table below contains each model we developed. It contains the model type, XGBoost or Neural Networks, resolution of the data set, training data and a label, or code, for each model. The code indicates the model identifier (a number) and whether the model we developed using single (S) or multiple (M) datasets (experiments).

Model Type	Resolution	Training Data	Model code
XGBoost and Neural Networks	Low and High	FBL01	S1
		FBL02	S2
		ETNA01	S3
		ETNA02	S4
		MONZ04	S5
		MONZ05	S6
		WG01	S7
		WG02	S8
		WG04	S9
		GRS02	S10
		GRS03	S11
		ANS02	S12
		ANS03	S13
		ANS04	S14
		ANS05	S15
XGBoost	Low	Sandstone	M16
		Sandstone and Basalt	M17
		Sandstone and Monzonite	M18
		Sandstone and Granite	M19
		Sandstone and Shale	M20
		Sandstone and Limestone	M21
		Basalt and Monzonite	M22
		Basalt and Granite	M23
		Basalt and Shale	M24
		Basalt and Limestone	M25
		Monzonite	M26
		Monzonite and Granite	M27
		Monzonite and Shale	M28
		Monzonite and Limestone	M29
		Granite	M30
		Granite and Shale	M31
		Granite and Limestone	M32
		Shale	M33
		Shale and Limestone	M34
		Limestone	M35

Table 3.3: This table shows each model we created with model type (XGBoost or Neural Network) and the experiment and resolution we used. In cases where we used multiple experiments, we write which rock type or types were used. Lastly we give each model created a model code with an identifier (a number) and whether the model was developed using a single (S) or multiple (M) experiments. The experiment codes are explained (3.1).

Model Code	Test Data
S1	FBL01
S2	FBL02
S3	ETNA01
S4	ETNA02
S5	MONZ04
S6	MONZ05
S7	WG01
S8	WG02
S9	WG04
S10	GRS02
S11	GRS03
S12	ANS02
S13	ANS03
S14	ANS04
S15	ANS05
M16	Sandstone (FBL01, FBL02) Basalt (ETNA01, ETNA02) Monzonite (MONZ04, MONZ05) Granite (WG01, WG02, WG04) Shale (GRS02, GRS03) Limestone (ANS02, ANS03, ANS04, ANS05)
M17	
M18	
M19	
M20	
M21	
M22	
M23	
M24	
M25	
M26	
M27	
M28	
M29	
M30	
M31	
M32	
M33	
M34	
M35	

Table 3.4: This table shows which data was used to test each model. Models are tested on all unseen data meaning models tested on 80% of the data in one experiment will be tested on 20% of the data in the same experiment. If a model has not been trained on data in an experiment, the model will be tested on 100% of the data in that experiment. Model codes, signifying which data was used to train the model, are explained in table 3.3.

CHAPTER 4

Results

Our results are split into two main sections based on whether we used individual experiments or combinations of multiple experiments to train and test our models.

First, we compare model performance of XGBoost and Neural Networks at low and high resolution, and the training time for these models. We then go on to comparing input data and model prediction at low and high resolution to investigate how resolution changes our results. After that we investigate the model predictions in more detail, and continue by comparing them to the local strain components and SHAP-importance values of the local strain components. We finish the first section by looking at the transfer scores of models trained on individual experiments.

In the second section we detail transfer scores on models trained and tested on combinations of the experiments sorted by rock type. We continue by detailing the SHAP values of the models trained on multiple experiment. We finish this section by training models where we combine all experiments in two rock types and test the models on all the rock types.

4.1 Individual Experiments

In this section we detail the results of training and testing models on individual experiments. We go over non-transfer model performance, compare local strain values to model prediction, review SHAP values, and transfer performance.

4.1.1 Model Performance

First, we examine the non-transfer model scores for the low- and high-resolution datasets of individual experiments (Figure 4.1). These non-transfer model scores are derived from training a model on 80% of the data in one experiment, and then testing it on the remaining 20% unseen data of the same experiment. The split is random in time and space. Examining the performance of the models at both spatial resolutions will indicate which resolution and model type produce the highest model performance. The sample points in the data are not independent in time and space, and so autocorrelation may contribute to a larger R^2 for the non-transfer tests. Since XGBoost can be described as a self-evaluating algorithm (Chen and Guestrin 2016) and the neural networks we used are not, we might expect that the neural networks are affected by overfitting to a slightly larger degree than our XGBoost models. In addition, while we apply regularization to the neural networks in the present study, there are multiple techniques like dropout that we do not employ.

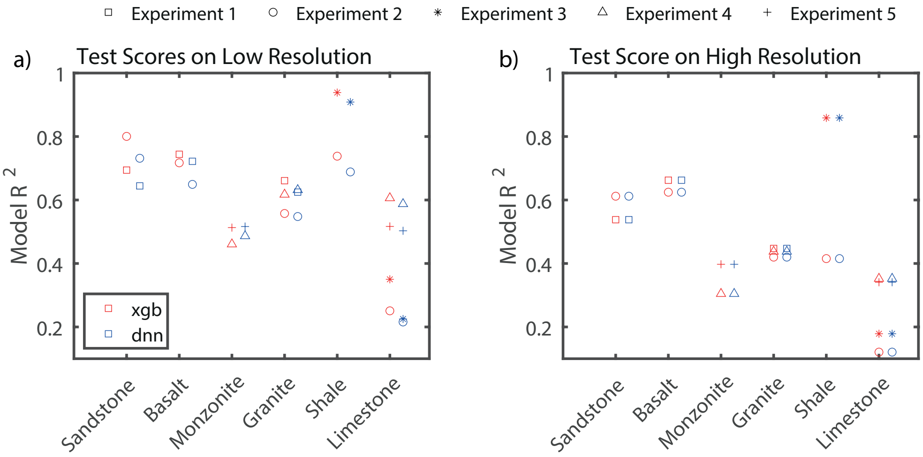


Figure 4.1: R^2 scores of each experiment on low (a) and high (b) resolution data. XGBoost scores (red) and Neural Network scores (blue) are sorted into rock types and experiments are displayed with a symbol showing the experiment number (upper legend).

For the low resolution data , XGBoost tends to perform better than the neural networks (Figure 4.1). For high resolution data, the performance is similar for both methods. However, the scores tend to be lower for higher resolution data than the lower resolution data. This result may occur because the higher resolution data appears to contain more noise than the

Model Type	Parallel Training	Resolution	Average time (s)
XGBoost	No (Serial)	Low	10.75s
		High	219.57s
Neural Network	No (Serial)	Low	4.29s
		High	19.05s
XGBoost	Yes (Parallel 4 Cores)	Low	10.24s
		High	232.55s

Table 4.1: The average training time of our model types at different resolutions and parallelism. Average time is calculated by averaging the time spent training for one model type on each experiment. The number of sample points for each experiment is similar, which also means time spent training each model is similar (except for ANS04). We see that XGBoost is considerably slower than Neural Networks, and parallelizing the XGBoost algorithm does not improve times.

low resolution data (Figure 4.2). Intuitively, we could assume that machine learning models would be able to learn more and perform better at higher resolutions as they have access to more data points. However, our models do not benefit from the increase in the number of data points. The standard deviation of the local strain components shows that spatial resolution affects both the data and the prediction in a similar way (Figure 4.2). The variance in the model prediction increases as the standard deviation of the local strain components increase.

While the model performance is very important, the required computation time to train a machine learning model is also relevant to examine. For our dataset, the computation time tends to vary by an order of magnitude from the low to high resolution data (Table 4.1). Here, neural networks have the lowest average time. The average time is calculated using a set parameter space so that we do not include the time spent by other functions such as GridSearch. We take the time spent training each model for individual experiments at one resolution and average across the times for all the experiments. Table 4.1 shows that XGBoost is slower than the neural networks. Moreover, running the algorithm in parallel does not improve the timing. Another method of reducing the computation time is to parallelize the training loop so that we train multiple models at the same time.

Because the highest performance achieved was on the XGBoost models developed with the low resolution data, we focus on these models for the remainder of the analysis. First, we categorize some of the main

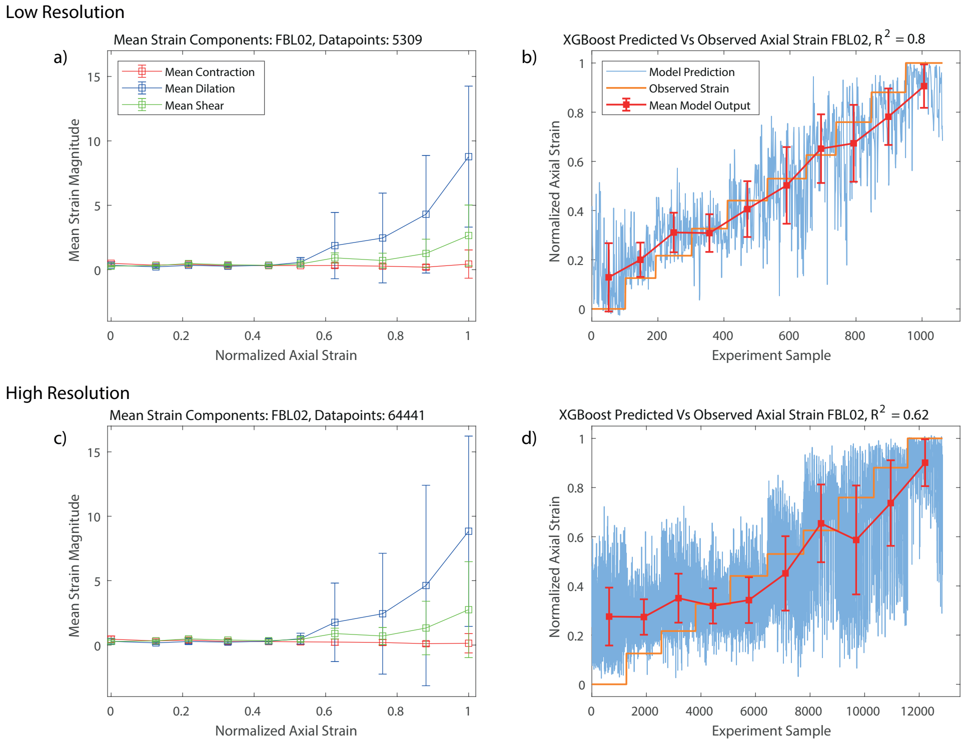


Figure 4.2: Mean and standard deviation of the local strain components in low (a, b) and high resolution data (c, d), and the corresponding model prediction and observed macroscopic axial strain for FBL02. The mean of the strain components is similar in both resolutions, but the standard deviation increases at high resolution. The increase in the standard deviation from low to high resolution correlates with a similar increase in the variance of the model prediction.

characteristics that separate weakly ($R^2 < 0.5$), moderately ($0.5 < R^2 < 0.7$) and strongly ($R^2 > 0.7$) correlated models. Out of the 15 models produced during training on individual experiments (Table 3.3, models S1-S15), we examine the predictions of the model with the highest performance for each rock type (Figure 4.3). Because the experiment includes discrete time steps, when we compare the observed and predicted macroscopic axial strain, the observed strain forms a function with discrete steps (i.e., stair steps) plotted against the sample number (i.e., index of the row in the dataset). Note that the sample number is the enumerated value, or index in the list, of each unique value of the strain components. These indexes are ordered by time so that multiple samples correspond to a single unique value of

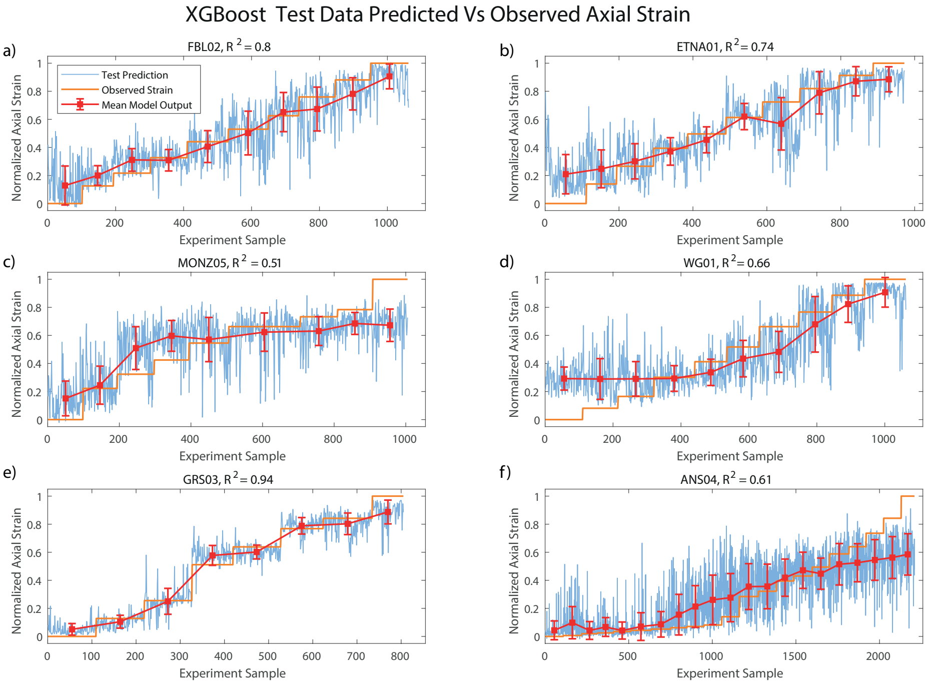


Figure 4.3: Predicted (blue) and observed (orange) axial strain in the test dataset for the model with the highest R^2 score (displayed in each subtitle) of each rock type. Red lines show the mean of the model prediction for each of the observed strain values.

macroscopic axial strain. Thus, because we have multiple values of the strain components throughout the 3D rock core at each discrete time step, the observed macroscopic strain forms a discrete, piecewise function with stair steps.

With these plots of the predicted and observed normalized axial strain, $\hat{\varepsilon}$, we can highlight the strain values that the models predict with higher and lower accuracy (Figure 4.3). As expected, the variance between the predicted and observed values increases as R^2 decreases. The models developed with MONZ05, WG01 and ANS04 are more inaccurate early and late in the strain evolution. However, the models developed with FBL02, ETNA01, WG01 and GRS03 produce predictions that closely match the "stair-step" pattern of the observed data.

4.1.2 Local strain values that control the model performance

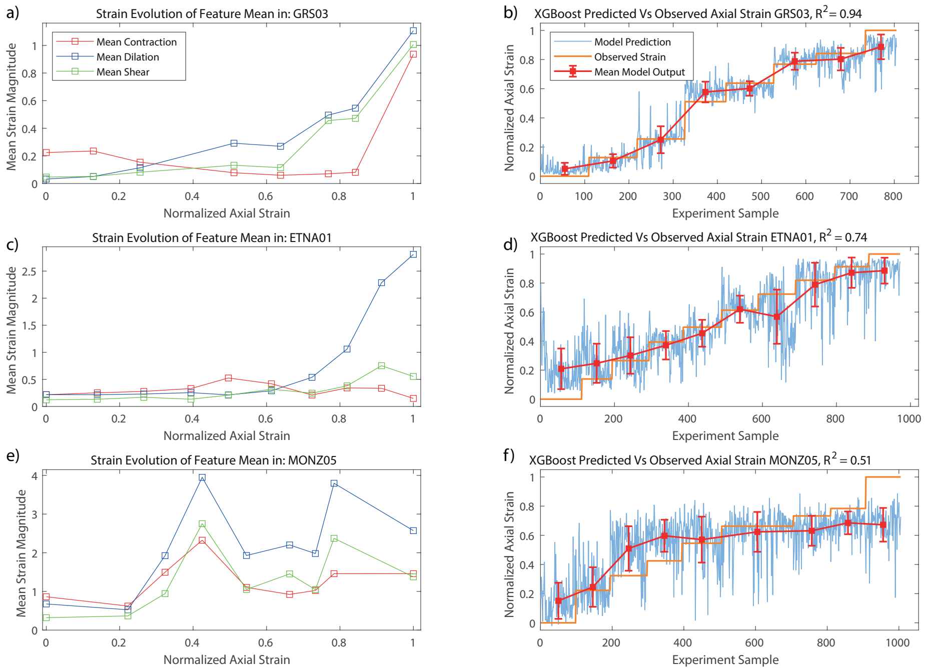


Figure 4.4: Evolution of the local strain components (a, c, e) and predicted and observed macroscopic strain (b, d, f). The mean strain magnitude (y-axis, a, c, e) is the mean value of for each corresponding macroscopic axial strain value (orange, b, d, f), scaled by the RobustScaler function. The mean values reported here (and in similar plots) are not the exact same values as the ones reported in the raw DVC dataset, but instead the scaled data with which we train our model.

Next, we compare the evolution of the local strain components and the model prediction of the macroscopic axial strain to investigate how trends in the data influences the model performance. The correlations between trends in local strain components and model prediction may reveal why certain models are more accurate than others. We examine the evolution of the mean of the local strain at each time step of macroscopic axial strain (Figure 4.4).

The GRS03 experiment hosts a systematic evolution of distinct local strain values that increase toward failure (Figure 4.4a). Correspondingly, the model has lower variance and higher R^2 (b) than the other models. The evolution of the shear and contraction in the ETNA01 experiment is less systematic than those in the GRS03 experiment. Consequently, the model has a higher variance, and a lower R^2. The evolution of the local strain values in the MONZ05 experiment is not systematic, and many of the values are relatively similar to each other. In this case, the model performance is significantly lower than the other models: the model tends to guess values around $\hat{\varepsilon} \approx 0.6$, producing a high variance. The model performs somewhat better for the earlier parts of the strain evolution, in which the local strain components systematically increase. After this early stage, the evolution becomes less systematic . These trends suggest that higher performance and quality in a model is connected to more systematic evolution of the local strain components, and correspondingly less systematic evolution for lower performing models.

The model SHAP values further help explain the predictive power of the features in our models (Figure 4.5). Revealing trends in the feature importance of our models may help indicate what each of our models has learned about the physics of macroscopic failure. We also may identify differences between the features with the most predictive power in models with higher and lower performance. We examine feature importance using a cumulative importance, which is calculated using the average |SHAP| value of each feature. The values are normalized and weighted so that the score of each model influences the degree to which that model affects the overall importance. The equation for this importance metric is $I = \sum |\widehat{SHAP}| R^2$.

Features that use statistics that quantify the extreme values (10^{th} and 90^{th} percentiles) rank lower than the feature statistics that quantify the intermediate values (mean, 50^{th} percentile) (Figure 4.5). Features that use the dilatational strain provide more predictive power than the other strain components. The models in which dilation is the most important strain component also tend to have higher model scores. The most important feature using this cumulative metric is the dilation mean, which is also the most important feature for several individual models.

4.1.3 Transfer Learning Performance

Because we wish to assess how much our models have learned about the underlying physics of macroscopic failure under triaxial compression, we test (Table 3.4) them using transfer learning (Figure 4.6). Transfer learning involves training models with data from one or several experiments, and then testing the models on data from other experiments. Our transfer

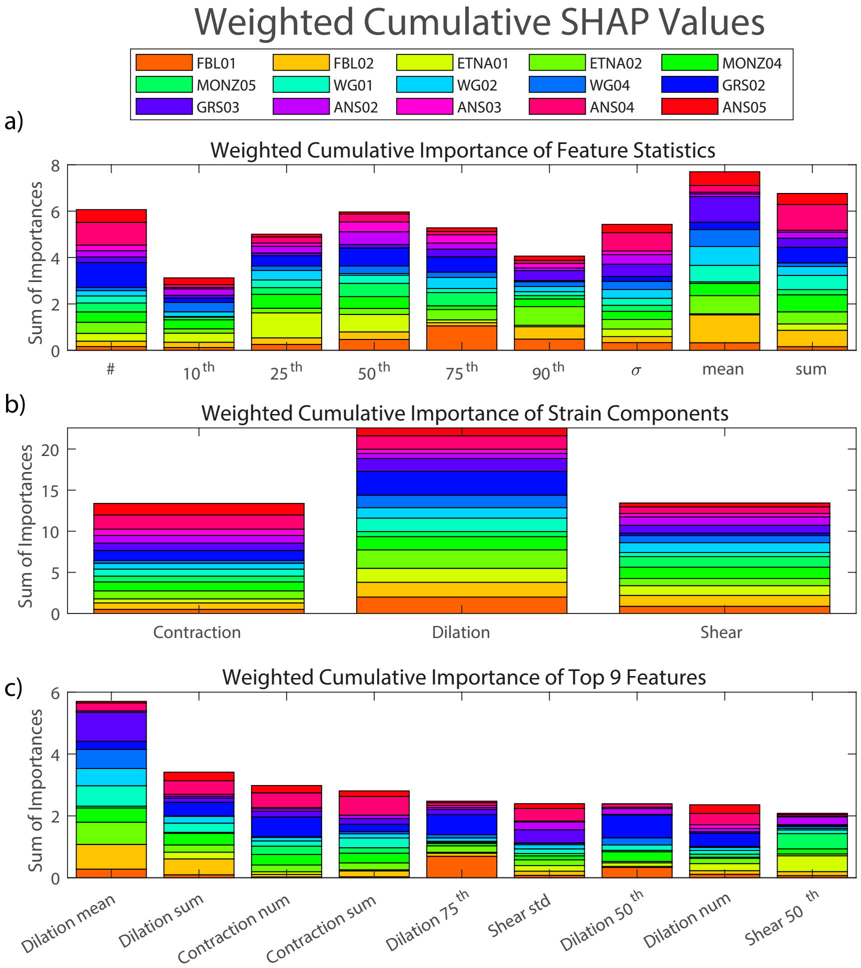


Figure 4.5: Cumulative SHAP value sorted by statistic (a), strain component (b) and top 9 features (c) for models trained on each experiment. This cumulative importance is calculated using the average $|SHAP|$ values of each feature. The values are normalized and weighted so that the score of each model influences the degree to which that model affects the overall importance. The equation for this importance metric is $I = \sum |\widehat{SHAP}| R^2$. We also display every importance value color coded and corresponding to each experiment used to train the XGBoost model.

scores have a large range of values (-1.01 to 0.82). The mean and standard deviation of the transfer scores indicate weak or no correlations between the predicted and observed values (0.04 ± 0.34). In contrast, the mean score of our non-transfer models 0.61 ± 0.17. As described in the background sections, we expect transfer learning performance to be higher between 1) sandstone and basalt, 2) monzonite and granite, and 3) shale and limestone.

As discussed previously, most of the non-transfer scores of this set of models are reasonable, except MONZ04, ANS02 and ANS03, where the scores are below $R^2 = 0.5$ (Figure 4.6). The models developed with the limestone (ANS) data have lower transfer scores than all the other rock types. Most of the transfer learning scores for the other rock types are low (<0.5). However, there are some exceptions to this trend, such as the FBL02 and GRS03 transfer learning scores. Thus, when we develop the models using data from only one experiment, most of the models generalize poorly.

4.2 Combinations of experiments

In the previous section (4.1) we have trained models on one experiment and tested them on a different experiment. In this section we review the results of using combinations multiple experiments to train our models. With this we expect to improve the generalizability seen in the previous transfer tests.

4.2.1 Transfer and non-transfer rock type model performance

We now examine the model test scores when we combine the experimental data by rock type (Table 3.3, models M16-M35). Examining these transfer learning scores indicates that some of the higher transfer scores have decreased slightly, while the lower transfer scores have improved moderately (Figure 4.7). The mean and standard deviations of the non-transfer model performance indicate moderate-strong correlations between the predicted and observed axial strain (0.63 ± 0.13). The transfer scores indicate weak correlation for most of the tests (0.20 ± 0.26). Although these transfer scores are not moderately correlated, this range is a significant increase from the model scores of the individual experiment transfer learning scores. This general trend does not apply for Anstrude Limestone, which still produces lower transfer scores. The highest scores are achieved by models trained on Green River Shale, Fontainebleau Sandstone and Mount Etna Basalt, consistent with the individual experiment transfer learning scores (Figure 4.6). We still do not see the expected trends in transfer scores between rock types.

4.2.2 Local strain values that control the model performance

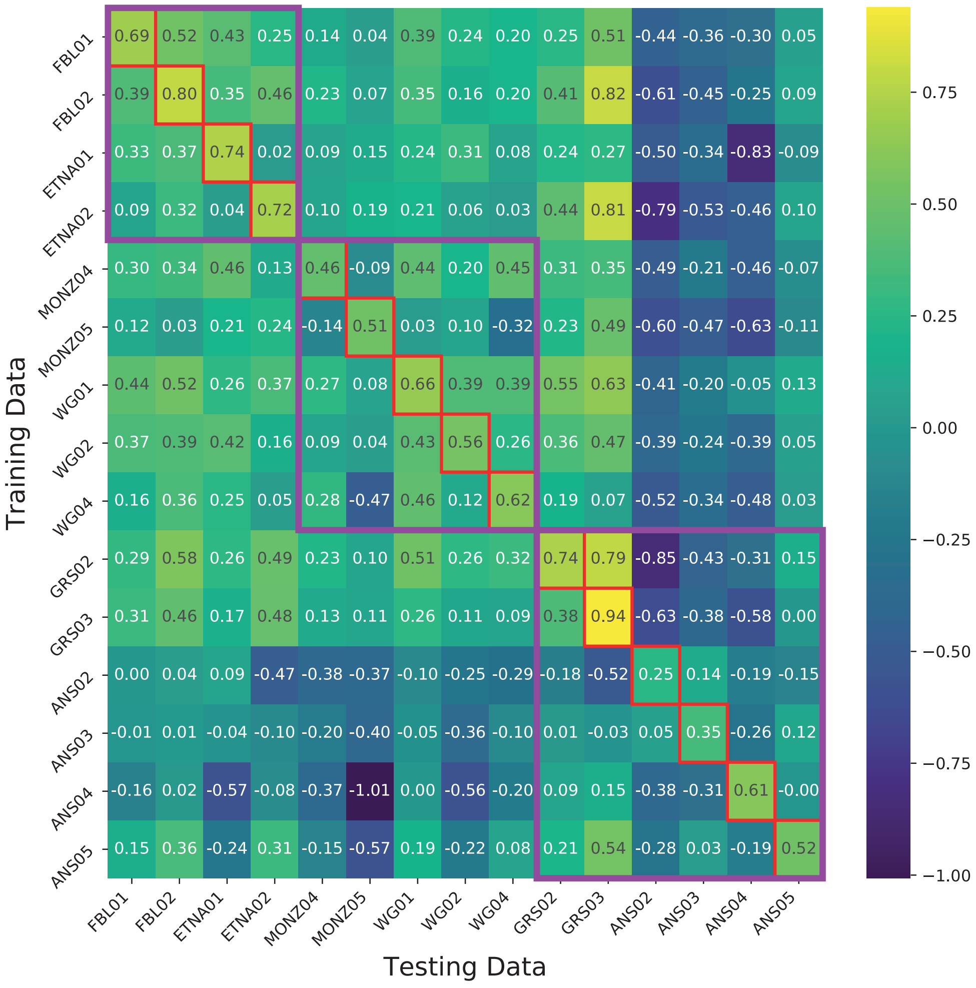
Comparing the SHAP values of the models developed with individual experiments to those developed with multiple experiments may help us understand the slight improvement in generalization (Figure 4.8). When we develop models from individual experiments, the most important feature statistics include the mean, sum, 50^{th} percentile, and number of strain values, in that order (Figure 4.8). Similarly, when we develop models from groups of experiments, the most important feature statistics include the number of values, the 50^{th} percentile, mean and sum, in that order (Figure 4.8). Note, the number of strain measurements in a subvolume tends to increase for dilation and decrease for contraction as we approach macroscopic failure. For both types of models, developed with groups of experiments and individual experiments, dilation is considerably more important than the other local strain components . This tells us that while similar some differences occur when using multiple experiments. These differences may help us understand more about macroscopic failure in rocks, and why our performance increases. For example, exchange in the number of values seems to be a general trait in macroscopic failure.

4.2.3 Extending generalizability to multiple rock types

Comparing the mean of the transfer learning R^2 scores of models developed with individual experiments (0.04 ± 0.34) and multiple experiments (0.20 ± 0.26) indicates that combining experimental data improves the generalizability of the models. Thus, next, we combine two rock types in order to increase the generalizability. In general, models developed using sandstone, basalt and shale perform better than the other models. Additionally, limestone models generally have a lower performance than other models.

The model R^2 scores improve overall from the transfer learning scores of the models developed with one experiment, but most of the scores still indicate weak or moderate correlation. The mean ± one standard deviation of the R^2 transfer scores of these models is 0.23 ± 0.25 , while this range of the transfer scores of the single rock type models is 0.20 ± 0.26, and the mean of the transfer scores of the individual experiment models is 0.04 ± 0.34. Thus, the generalizability of our models has increased from scores indicating (almost) no correlation, to scores indicating low correlation. For the non-transfer scores, there is a small tradeoff in the scores achieved by the models trained on a single rock type (0.63 ± 0.13) and the models trained on multiple rock types (0.58 ± 0.15), but the scores are still very close in performance . With this we see the quantifiable success of increasing the generalization of our models using multiple data.

It is difficult to precisely quantify what more our models learn by using data from multiple experiments. We could intuitively think that since dilation is highly important, our models will learn more about dilation when combining data. Observing the SHAP value, we do not see any such effect, instead we see that the overall importance of each SHAP value stays the same. A difference between individual and multiple experiments is that the number of values increases in importance. A possible reason for this is that the number of values will generally vary between a lower number for dilation and a higher number for contraction at lower differential stress, and higher number for dilation, lower number for contraction at higher differential stress. The variation of the number of values will increase when adding more experiments, thereby also increasing its importance. We can also attempt to explain the increase in generalizability considering the hypothesis that inputting similar values in the local strain components should return similar values for the macroscopic axial strain. This would help explain the great increase in average performance at the cost of a slight increase in model variance when combining data. This effect may occur because it is more likely to predict the correct values when a model has seen more possible relations between local strain components and macroscopic axial strain. When exposed to unseen data our models may use information from multiple other experiments to predict the distance to failure .



	FBL01	FBL02	ETNA01	ETNA02	MONZ04	MONZ05	WG01	WG02	WG04	GRS02	GRS03	ANS02	ANS03	ANS04	ANS05
FBL01	0.69	0.52	0.43	0.25	0.14	0.04	0.39	0.24	0.20	0.25	0.51	-0.44	-0.36	-0.30	0.05
FBL02	0.39	0.80	0.35	0.46	0.23	0.07	0.35	0.16	0.20	0.41	0.82	-0.61	-0.45	-0.25	0.09
ETNA01	0.33	0.37	0.74	0.02	0.09	0.15	0.24	0.31	0.08	0.24	0.27	-0.50	-0.34	-0.83	-0.09
ETNA02	0.09	0.32	0.04	0.72	0.10	0.19	0.21	0.06	0.03	0.44	0.81	-0.79	-0.53	-0.46	0.10
MONZ04	0.30	0.34	0.46	0.13	0.46	-0.09	0.44	0.20	0.45	0.31	0.35	-0.49	-0.21	-0.46	-0.07
MONZ05	0.12	0.03	0.21	0.24	-0.14	0.51	0.03	0.10	-0.32	0.23	0.49	-0.60	-0.47	-0.63	-0.11
WG01	0.44	0.52	0.26	0.37	0.27	0.08	0.66	0.39	0.39	0.55	0.63	-0.41	-0.20	-0.05	0.13
WG02	0.37	0.39	0.42	0.16	0.09	0.04	0.43	0.56	0.26	0.36	0.47	-0.39	-0.24	-0.39	0.05
WG04	0.16	0.36	0.25	0.05	0.28	-0.47	0.46	0.12	0.62	0.19	0.07	-0.52	-0.34	-0.48	0.03
GRS02	0.29	0.58	0.26	0.49	0.23	0.10	0.51	0.26	0.32	0.74	0.79	-0.85	-0.43	-0.31	0.15
GRS03	0.31	0.46	0.17	0.48	0.13	0.11	0.26	0.11	0.09	0.38	0.94	-0.63	-0.38	-0.58	0.00
ANS02	0.00	0.04	0.09	-0.47	-0.38	-0.37	-0.10	-0.25	-0.29	-0.18	-0.52	0.25	0.14	-0.19	-0.15
ANS03	-0.01	0.01	-0.04	-0.10	-0.20	-0.40	-0.05	-0.36	-0.10	0.01	-0.03	0.05	0.35	-0.26	0.12
ANS04	-0.16	0.02	-0.57	-0.08	-0.37	-1.01	0.00	-0.56	-0.20	0.09	0.15	-0.38	-0.31	0.61	-0.00
ANS05	0.15	0.36	-0.24	0.31	-0.15	-0.57	0.19	-0.22	0.08	0.21	0.54	-0.28	0.03	-0.19	0.52

Figure 4.6: Transfer learning scores for the XGBoost models trained on the low resolution data for each experiment. Along the y-axis we see the experiment code of the experiments used for training, and along the x-axis we see the experiment code of the experiments used for testing. Along the diagonal (marked red) are the non-transfer scores (training and testing on the same experiment). The Purple squares show areas where we expect transfer scores to be higher due rock types within each square.

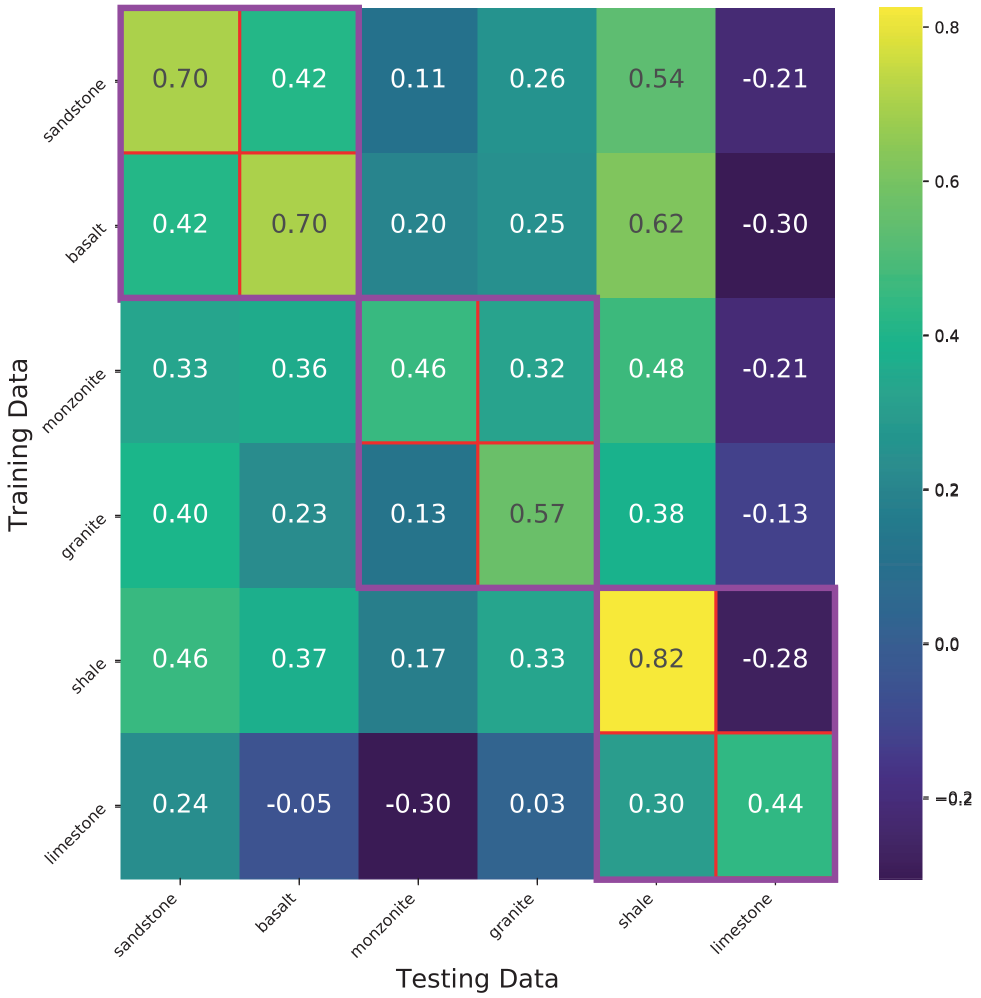


Figure 4.7: R^2 scores for models developed with multiple experiments on the same rock type. The score matrix shows the non-transfer (training and testing on the same rock type) and transfer (testing on different rock types). The squares colored red are non-transfer test scores, and the larger squares colored purple are the squares where we expect higher transfer scores due to expected similarities in the rock types.

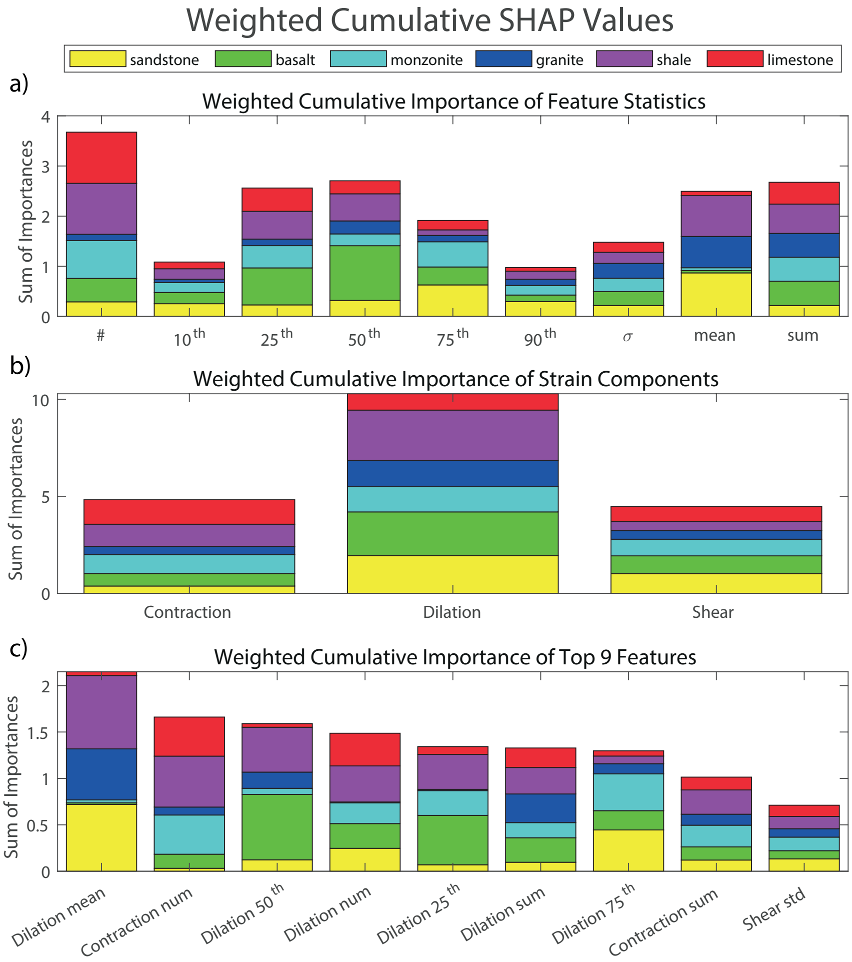


Figure 4.8: Cumulative SHAP value for models trained on rock types sorted by statistic (a), strain component (b) and top 9 features (c). This importance is calculated using the average |SHAP| values of each feature. The values are normalized and weighted so that the score of each model influences the degree to which that model affects the overall importance. Dilation is the most important strain component (b), similar to the models trained on one dataset. The main difference between these models and models developed for individual experiments (Figure 4.5) is that the most important statistic is the number of strain values, rather than the number, mean, and sum.

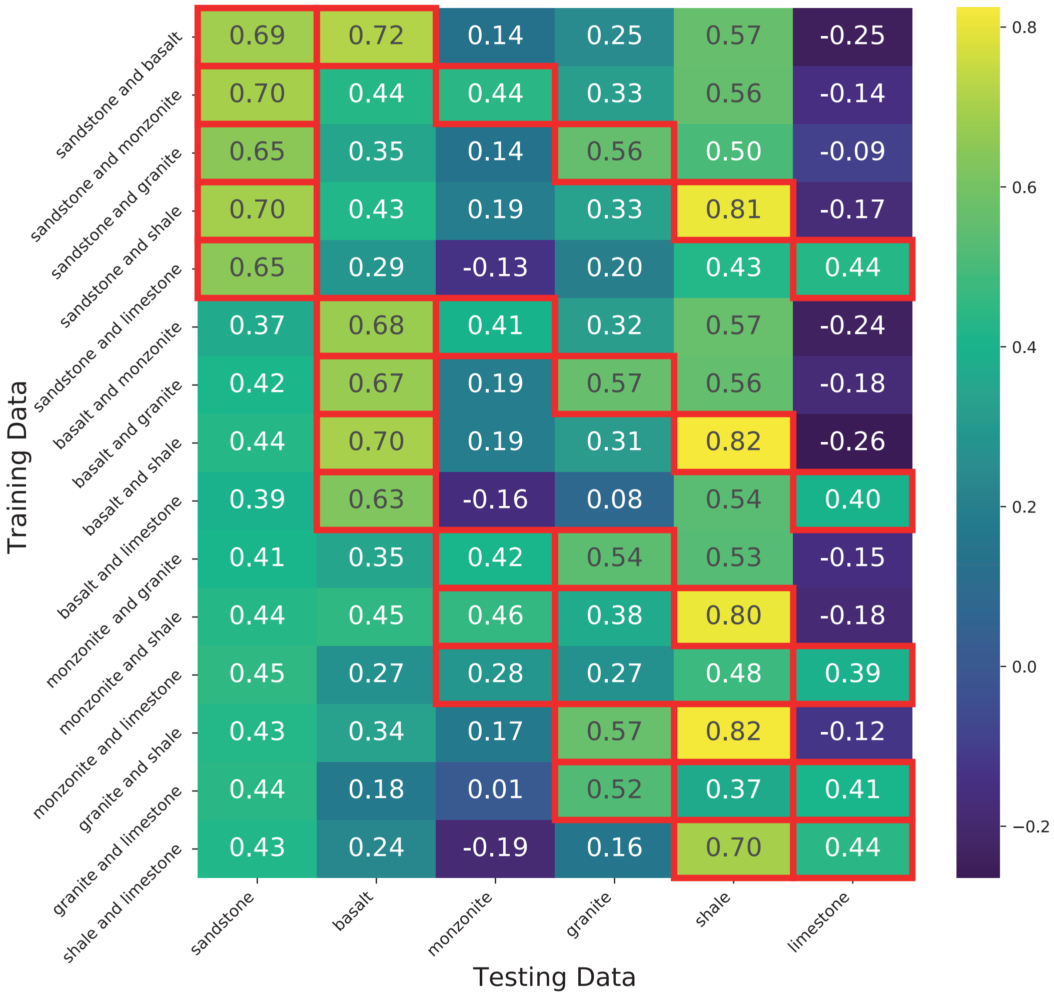


Figure 4.9: XGBoost R^2 scores of models found from combining two rock types, then testing these models on each rock type. Models are created using 80% of the data in each rock type combination, and tested using all unseen data for each rock type. The rows show the training data and the columns show the rock type used as testing data. The non-transfer scores have been marked red.

CHAPTER 5

Discussion

5.1 The individuality of rock deformation in porous and crystalline rocks

When conducting experiments with transfer learning, on both the individual experiments, and the experiments combined by rock type, we expected transfer scores to increase between pairs of rock types that were assumed to have similar strain accumulation. These pairs were 1) sandstone and basalt, 2) monzonite and granite, and 3) shale and limestone. As seen in figures 4.6, 4.7 and 4.9, the transfer scores between these three pairs of rock types are not systematically higher than other transfer scores. This result contrasts with the result of McBeck et al. (2020a) on the same experiments. In that study they predicted the distance to failure as a classification problem, and found that the transfer scores between sandstone and basalt, and shale and limestone were higher than other rock type pairs. There may be multiple reasons for the discrepancy between these results, including the differences between classification and regression. First, we consider why the first pair of rocks have lower transfer scores in this analysis. For the samples with Fontainebleau sandstone and Mount Etna basalt, we expect failure to occur due to microcracks developing at the edges of pores due to higher stress concentrations. In previous studies, this failure mechanism was observed in both rock types (Renard et al. 2019; Zhu et al. 2016). The porosity of the Etna basalt in this study is lower than for the sandstone, by about half, which may cause the buildup of stress concentration to be somewhat different for the two rock types. In addition, Zhu et al. (2016) observed that microcracks tend to avoid propagating across phenocrysts, such a mechanism is not seen in sandstone due to it being more homogeneous in mineralogical composition . Examining the mean local strain components, we see that

ETNA02 hosts a less systematic strain accumulation than the other rock types (Figure A.1). In addition, the evolution of FBL01 is relatively flat. The magnitudes of the dilatational strains are different for all experiments. The overall scores between these rock types are somewhat more stable and on average higher than scores seen for the other pairs. However, if we take other rock types into consideration, we see that the transfer scores are still not significantly higher than other transfer scores for these rocks . To illustrate, the range of transfer scores between basalt and sandstone is 0.04 to 0.52, while for monzonite and granite they are -0.47 to 0.46 and lastly for limestone and shale they are -0.85 to 0.79 (with most scores here being very low). The other transfer scores of sandstone and basalt range from -0.83 to 0.82, but here all the negative scores are from limestone.

Next, we discuss the differences found in monzonite and Westerly granite, the second pair of rock types. These rock types are both crystalline rocks, with very similar mineralogical composition and low porosity. All the transfer scores between these rocks indicate a low correlation, with some close to moderate (MONZ04). As discussed in McBeck et al. 2020a, where similar results between these rock types were seen, some possible reasons for this include differences in confining stresses, and differences grain size. Section 2.2.6 describes these differences in more detail. Note that the scores of models trained on MONZ04 and tested on WG01 and WG04 were among the higher scores achieved by that model. The mean local strain component curves (Figure A.1) of these rocks have more similar trends than between for example MONZ05 and any of the other granite samples . This indicates that similarities in the local strain components affect our transfer scores, which we will discuss further by the end of this section.

The last pair of rocks are the Green River shale and Anstrude limestone, where we expect deformation to occur with dominantly compactive failure mechanisms. The main mechanisms are compaction bands and pore collapse, however both rock types may also deform due to dilatancy. In the experiments, shale deformed with compaction bands but higher magnitudes of dilatational strains were also observed. In most of our limestone experiments, compaction leading to pore collapse was the dominant failure mechanism (Renard et al. 2017) . The SHAP values of the two rocks shows that for the experiments with shale, the impact of dilation on our models is highly important. However, we do not observe dilation to be as important in the limestone experiments, instead contractive strains are more important in these experiments. This result may be a quantifiable reason behind the negative transfer scores we observe between shale and limestone. One possible explanation of the differences in these experiments include the low porosity of shale compared to limestone, as limestone failed due to pore collapse in the rock. In addition, we did not observe compaction bands in the limestone experiments, as opposed to the shale experiments.

When considering the individuality of rock deformation in the context of machine learning, we may use transfer learning to aid our understanding. We know that between two domains with the same types of features, there may still be differences in their marginal probability distribution. Marginal probability distributions are important to transfer performance: if these are too dissimilar between two domains the performance will decrease (Shimodaira 2000) . In the context of our results, these differences represent the differences in input data, such as whether the experimental data exhibits acceleration towards failure in both experiments. The concept of similarities in marginal probability distributions should even extend to whether acceleration towards failure is seen in the same features in both experiments. We know from our results that systematic evolution of the local strain components improves the performance of the model. We can consider the possibility that to achieve good transfer performance we must not only have a systematic evolution of the local strain components, but the evolution must also have similarities in both experiments. This may partially explain why transfer scores for the GRS03 experiment were particularly high. This experiment may have had a domain that carried similarities to multiple others. The domain or features in GRS03 had a systematic evolution in multiple features. From the perspective of transfer learning this may help explain some of the lower transfer scores and increase our understanding regarding why rock types expected to have similar strain accumulation do not have elevated transfer scores. In some cases, these scores may be improved using domain adaptation techniques to closer match domains of different rock types. In particular, techniques to reduce differences in either marginal or conditional probability distributions, may improve transfer scores. Future investigation into such techniques this may also prove useful in explaining why the expected patterns to did not emerge.

The machine learning methods that we use in the present study were sensitive to differences in the feature spaces of different experiments. This means that models made using individual experiments would not generalize as well on unseen data from different experiments. Models made using multiple rock types were able to learn patterns in most of the information of both experiments, and perform well on tests with unseen data from experiments they had trained on. However, models trained on multiple rock types were also able to predict the distance to failure of rock types outside the training data (in transfer learning), better than models trained on individual experiments could predict the distance to failure on other experiments. This suggests that models may learn more about rock deformation in general by training on multiple rocks.

5.2 The importance of dilation in rock deformation within the upper brittle crust

In this study , we used SHAP values to compare the importance of the features in each set of input data (experiments, or rock types) (Figure 4.5, 4.8). In the previous section, we discussed the individuality of rock deformation for each rock type. It is clear from our results that different rock types accumulate strain with varying signatures, with some commonalities. Figures 4.5 and 4.8 show that the dilatational strains are significantly more important than the contractive and shear strains. For every model, the dilatational features are considered the most impactful on the model performance for every rock type apart from limestone, where contractive strains are slightly more important. Contractive strains were expected to be important in the deformation of shale, and while we observed high magnitudes of contraction close to failure we also saw high magnitudes of dilation and shear strains with the deformation being dominated by dilation. The discrepancy in the importance of features between limestone and the other rock types may help explain why transfer scores between limestone and other rock types are generally the lowest.

While the transfer scores in this study differed from the earlier analysis done in McBeck et al. (2020a), the results of the SHAP value analysis is consistent with their findings. In both studies, the intermediate values of the dilatational strains had a higher impact on the models than the contractive and shear strains when predicting the distance to failure. In addition, predicting distance to failure from the characteristics of fracture networks also indicates that features associated with dilation (fracture aperture, anisotropy, clustering) holds higher predictive power than the other features (McBeck et al. 2020b).

Laboratory experiments show that volumetric strain plays an important role, with accelerating dilatancy being a well-known precursor of macroscopic failure in rocks (Paterson and Wong 2005). Our agreement with previous results further confirms the importance of dilation in rock deformation experiments and quantified its importance compared to contraction and shear. Figures 4.5 and 4.8 show that dilation is close to being as important as contraction and shear added together. This information, combined with cases of observed precursory dilatancy related phenomena, explained in section (2.1.3), suggests that dilation should be highly important in the breakage of rocks in conditions analogous to the upper brittle crust. This observation supports the basis of the dilatancy-diffusion model: dilatancy is expected to precede, and play a central role in, crustal earthquakes. In a laboratory environments, this theory is highly successful, however based on the lack of evidence supporting dilatancy as a precursor in natural earthquakes, the model's predictive power may be limited (e.g., Main et al. 2012).

To reconcile the differences between rock deformation experiments in laboratory conditions and natural observations of earthquakes, we need to account for effects that can dampen the predicted precursory dilatational changes preceding earthquakes. Among the differences between natural earthquakes and laboratory macroscopic failure, the scale is an obvious factor to consider. Fracture networks, fluid flow and heterogeneities may be difficult to properly scale up from laboratory to earthquake conditions. With some spatial scaling effects accounted for, we should also consider the confining pressures may be lower, and the strain rate higher in the laboratory than in natural earthquakes. We know that decreased strain rates may dampen the growth of fracture networks and therefore suppress dilatant strain (Brantut et al. 2013). Lastly, many earthquakes are known to occur due to the reactivation of pre-existing faults, which may be highly important, especially in load bearing regions in the lithosphere (Holdsworth, Butler and Roberts 1997). These effects may contribute to the difficulty in finding precursory changes in strain before earthquakes.

5.3 Outlooks

With these models, we have gained much valuable information and observed some interesting results. With our 15 models made using individual experiments and 19 models made using multiple experiments with either one or two rock types, we learned that multiple data provide an advantage for attaining improved generalizability. It is however important to note that we cannot assume that these models by themselves can reliably predict failure in rocks during compression. One reason for this limitation is that these models do not operate on raw data, but instead use derived features. In addition, these features are all scaled, and predicting the distance to failure on unscaled data may lower the performance. These two reasons, coupled with the weakly correlated average transfer scores even seen in the best models ($R^2 = 0.23 \pm 0.25$), leads us to keep this limitation in mind.

However, even when considering the limitations of our models, there may be ways to further improve performance. We may accomplish this task using transfer learning techniques, or we could make models that also take 3D-images into consideration. Eventually a future model based on similar or adjacent techniques may provide the needed generalizability for predicting failure in rocks, both after and during macroscopic failure. A more obvious way to improve the generalizability of our models is to utilize additional data, should it become available.

While additional data may improve model performance, the performance of the models could also improve if we used different experimental conditions in the lab to test for other earthquake mechanisms. One of the mechanisms considered important in earthquakes is the previously discussed repeated reactivation of existing faults (Holdsworth, Butler and Roberts 1997). The importance of this mechanism suggests that conducting X-ray tomography on rock samples with pre-existing fault structures could further improve the search for a predictive model of earthquakes. Statistical analysis and machine learning on such experiments may prove highly useful for furthering the understanding of rock and earthquake physics.

When reviewing the application of neural networks in the present study we can consider ways to better increase their strengths. We used a relatively small network architecture as our models. We chose this architecture because we observed a reduction in performance when we increased the complexity. However, many of the more successful applications of neural networks are for image analysis, and with more complex architectures (Ronneberger, Fischer and Brox 2015). If we investigated utilizing convolutional neural network type architectures in the future, we may find that the performance of neural nets on the 3D-images exceed other models. In such a case they may supplement predictions made by XGBoost from the local strain components. Applying image analysis may even prove more beneficial when analyzing the previously suggested experiments with preexisting faults.

CHAPTER 6

Conclusions

To briefly summarize this study, we used machine learning techniques to predict the distance to failure in rocks under triaxial compression. We generated 79 different models on both individual experiments of rock deformation, and a combination of experiments in either one or two different rock types. Out of these models, we developed 30 neural network models and 49 XGBoost models. The 30 models (15 DNN, 15 XGB) trained on the high resolution data performed markedly worse than the ones trained on the low resolution data, suggesting that they picked up more noise during training. Here, we also reported that XGBoost performed with convincingly better R^2 scores than the neural networks at the cost of a bit more training time. In addition to evaluating the model performance with the R^2 metric, we also evaluated the impact of each feature on our models using SHAP. The most important results detailed in this study are listed below:

1. The evolution of the local strain components controls the performance of our models; Less similar values of local strain components between each DVC calculation increases performance. This trait is evident from the experiments with a systematic evolution of local strain approaching failure, which produce higher model performance.

2. Although models trained on individual experiments may perform with moderate or high correlation on test sets, they perform with lower correlation during transfer learning tests. While this decrease in performance was expected, we here show that the lowest transfer scores are between rocks with different SHAP importance values, and that higher transfer performance is somewhat correlated to trends in the feature space of a deformation experiment.

3. Even when the SHAP values are similar for two models, their transfer scores may be low, especially when trends in their feature spaces are different.

We do not see the expected patterns in scores between the pairs of rock types. This result suggests that the local process of rock deformation is individual to each rock type, and may also vary between individual experiments for some rock types.

4. The dilatational strain components are highly important to predicting rock deformation in laboratory conditions analogous to the upper brittle crust. Both in models created with individual experiments and multiple experiments, features that include dilation have close to double the importance of either the contraction or shear strain.

5. We can increase the generalizability of our models by training them on multiple experiments and multiple rock types. In the present study, we were able to go from a mean and standard deviation of ($R^2 = 0.04 \pm 0.34$) for models trained on individual experiments to ($R^2 = 0.20 \pm 0.26$) and ($R^2 = 0.23 \pm 0.25$) for models trained on multiple experiments from one or two rock types respectively.

The techniques used in this study did not yet result in models that are ready for predicting the distance to failure in rocks during triaxial compression. However, we conclude that the systems being used in this study may result in something that could achieve this significant goal in the future, possibly with other adjacent techniques incorporated. Finally, we suggest that there may be merit in conducting more experiments with rocks under different conditions such as using samples with pre-existing faults to find more possible effects interacting during more complex systems such as natural earthquakes.

Appendices

APPENDIX A

Additional Figures

A.1 Figures

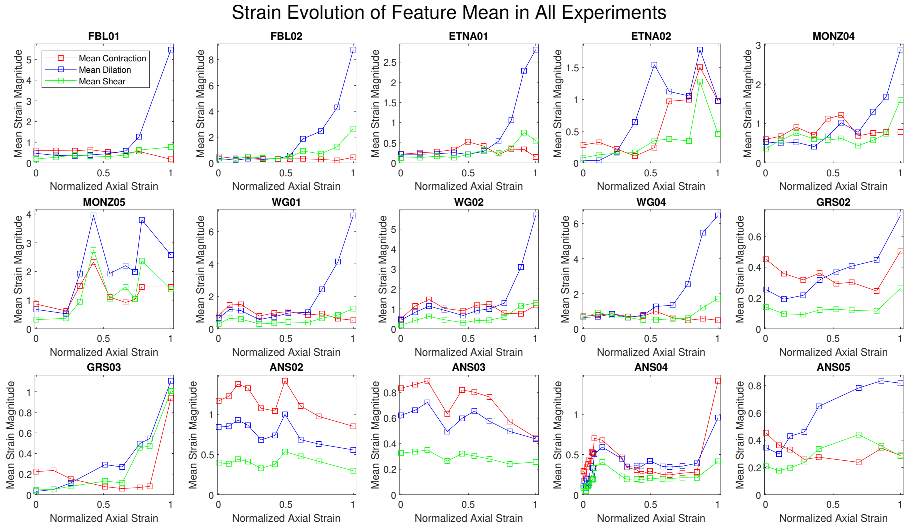


Figure A.1: This figure shows the mean statistic of the local strain components for each experiment reported in the present study at low resolution. The squares show the mean value per DVC calculation for each of these strain components.

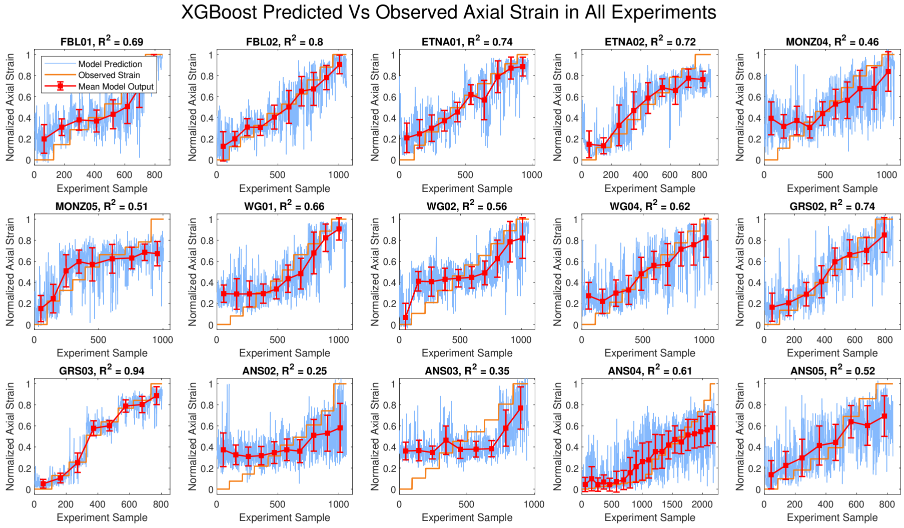


Figure A.2: This figure shows the XGBoost model prediction at low resolution versus the observed axial strain for all experiments reported in the present study. The blue lines are the raw model prediction, the orange lines are the observed normalized axial strain and the red bars are the mean and standard deviation of the model prediction (blue lines).

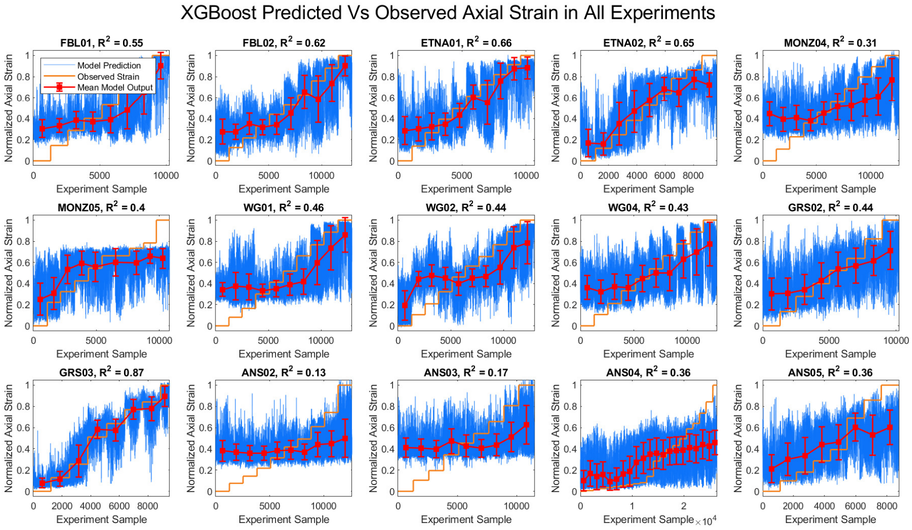


Figure A.3: This figure shows the XGBoost model prediction at high resolution versus the observed axial strain for all experiments reported in the present study. The blue lines are the raw model prediction, the orange lines are the observed normalized axial strain and the red bars are the mean and standard deviation of the model prediction (blue lines).

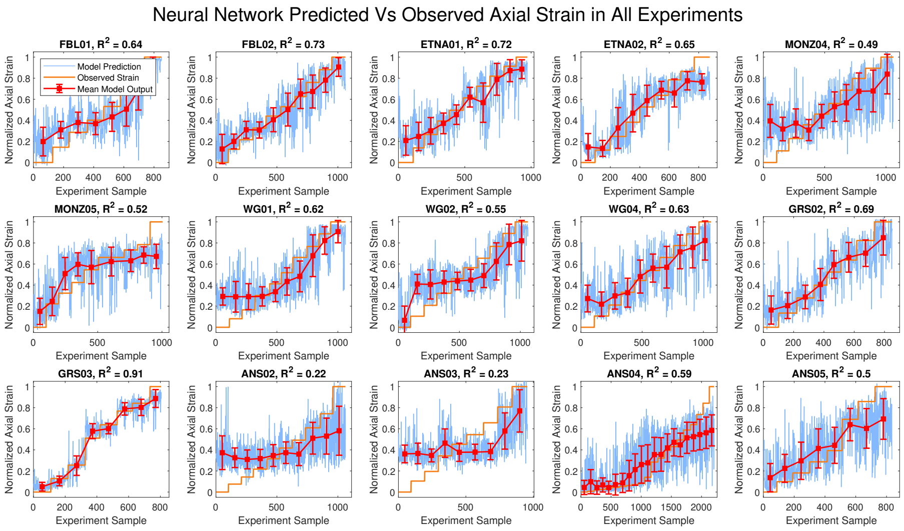


Figure A.4: This figure shows the Neural Network model prediction at low resolution versus the observed axial strain for all experiments reported in the present study. The blue lines are the raw model prediction, the orange lines are the observed normalized axial strain and the red bars are the mean and standard deviation of the model prediction (blue lines).

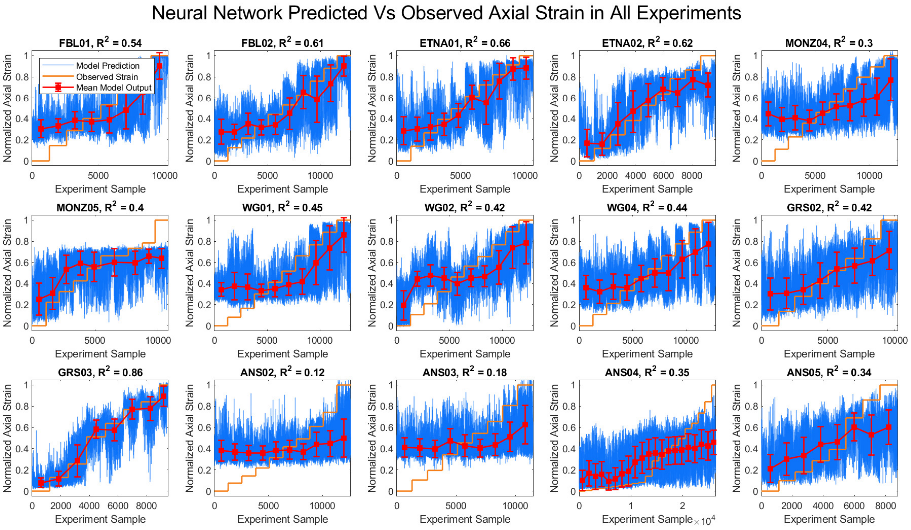


Figure A.5: This figure shows the Neural Network model prediction at high resolution versus the observed axial strain for all experiments reported in the present study. The blue lines are the raw model prediction, the orange lines are the observed normalized axial strain and the red bars are the mean and standard deviation of the model prediction (blue lines).

APPENDIX B

Supplementary Information

Here we define the stress and strain tensors because of the frequent usage of these terms in the present study.

B.1 Stress

Stress is a product of the internal forces in a medium resulting from an external force. The stress on a surface can be defined as the magnitude of the external force applied to the surface, divided by the area of that surface $\sigma = F/A$. The more general definition of stress is based on finding the state of stress at a point in a material like a rock. This state is described by a tensor composed of three orthogonal stress vectors.

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

The elements of this matrix are also often written with numeric labels. The diagonal elements are called the normal stress components, and the off-diagonal elements are called the shear stress components (sometimes written with the letter τ). In the special case of a stable stress-state, and the coordinate system is chosen so that the contribution of all off-diagonal elements is zero, we get:

$$\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

The diagonal elements of this matrix are called the maximum, intermediate and minimum principal stresses respectively. When discussing the differential stress in experiments with rock deformation, we then refer to the difference between the maximum and minimum principal stresses $\sigma_D = \sigma_1 - \sigma_3$. We can also define pressure as a state where there are no shear stress components, and all normal stress components are equal $\sigma_1 = \sigma_2 = \sigma_3$.

B.2 Strain

When a medium like a rock is under external forces, it may start deforming. Deformation is the transition from one shape to a different shape. Strain is a type of deformation called distortion, which is a non-rigid deformation. In a rock ,the change in shape, regardless of change in volume, results in particles changing their relative positions. A special case of strain is the change in length (from L_0 to L) of a rock sample along one axis (called L in the equation below), which can be described as the change in length due to deformation divided by the length prior to deformation:

$$\varepsilon_L = \frac{L - L_0}{L_0}$$

This is often referred to as the axial strain. In addition to this strain component, we should also define the shear strain and the volumetric strain, which refers to deformation that changes the volume of a rock. The shear strain is defined by:

$$\gamma = tan\,(\psi)$$

Where ψ is the angular shear, which is the change in angle in a deformed medium between two lines originally perpendicular. The volumetric strain is defined as the change in volume divided by the original volume:

$$\varepsilon_V = \frac{V - V_0}{V_0}$$

In the present study, we consider that positive changes in volume are related to dilation and negative changes are related to contraction. Similar to stress, strain can also be written in tensor form where the components are split into tensile and shear strains along the diagonal and off-diagonal respectively.

B.3 Strain Invariants

Change in local strain components can be calculated by using the invariants of the incremental strain tensor . The incremental strain tensor is calculated from the displacements between two scans, found using DVC. The first invariant $I_1(\Delta\varepsilon) = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}$, characterizes the local volumetric strain, which is negative for contractive strains and positive for dilatative strains. As described in Renard et al. (2018) and McBeck et al. (2018), we may find the local shear strains by relating the first invariant and second invariant of the incremental strain tensor $I_2(\Delta\varepsilon) = (\varepsilon_{xy})^2 + (\varepsilon_{xz})^2 + (\varepsilon_{yz})^2 - (\varepsilon_{xx}\varepsilon_{yy} + \varepsilon_{xx}\varepsilon_{zz} + \varepsilon_{yy}\varepsilon_{zz})$ to the second invariant of the incremental deviatoric strain: $J_2(\Delta\varepsilon) = \frac{1}{3}(I_1(\Delta\varepsilon))^2 - I_2(\Delta\varepsilon)$. Using this formulation, the relation to the local shear strains is found using the von Mises yield criterion equivalent strain $(3J_2(\varepsilon))^{\frac{1}{2}}$. Instead of using these invariants to calculate the contraction, dilation and shear strains, we use alternative criteria (McBeck, Ben-Zion and Renard 2020) .

B.4 Quality of Data

During the research presented in present study we hypothesized that outliers in the data may have affected the model performance. This was due to the standard deviation of some of the datasets seeming high and the results from 4.2 where we see that the higher standard deviation of the high-resolution data correlates with lower model scores. To test this hypothesis, we implemented a function that could remove the top percentage of points for each DVC in a dataset. To outline how the function works:

1. Find all unique values of strain, store the start and end point of each value of strain and store them in a list called counter.

2. For each feature we then go through the counter list and remove all values that exceed the percentile where we define points to be outliers.

3. We make a new dataframe that only contains the kept points.

4. If verbose is true we check how many points were removed.

This works because we know that every unique value of strain is sequentially increasing and because all non-unique values are sorted. Important to note is that this function will not remove 3% of the datapoints in the dataset when removing the top 3% percentile. We tested removing the top 5% (a), the top 3% (b) and no outliers (c):

a. When removing 5% of outliers our models had a score with a mean and standard deviation of 0.54 ± 0.19 non-transfer, and 0.05 ± 0.30 transfer for the individual experiments.

b. When removing 3% of outliers our models had a score with a mean and standard deviation of 0.55 ± 0.19 non-transfer, and 0.05 ± 0.30 transfer for the individual experiments.

c. With no removal of outliers our models had a score with a mean and standard deviation of 0.63 ± 0.13 non-transfer, and 0.04 ± 0.34 transfer for the individual experiments.

As we can see the effect of removing the outliers was to the detriment of our model performance. And while the mean of the transfer score does increase by 0.01 when removing outliers, the standard deviation decreases and many of the higher scores are significantly lower. This suggests that outliers in our data carry important information and noise in our data is probably in the form of some small irreducible noise. We conclude that the early hypothesis was inaccurate and that the quality of the data is not affected by large perturbations of noise.

APPENDIX C

Functions and Code

In this appendix we will show parts of the code explicitly, for the full code visit https://github.com/Anduron/Strain_masters/blob/master/scripts/Strain_work.ipynb. The code created during this thesis uses a set of python functions to create simple training and testing loops for many experiments at once. The code is not set up as a package, but is instead a jupyter notebook containing all machine learning experiments conducted during the present study. Possible improvements to the code include making things easier to generalize or abstract, by for instance object orienting. Using functions have made things simple enough to extend, test and work with for our purposes and might, depending on the observer, be easier to interpret. Note that many of the longer lines of codes has here been cut in half due to the format of the document.

To import our data we create a python function that can import a dataset and create a scaled dataframe, we also want to be able to concatinate multiple datasets. For this purpose, we use the following python function:

```python
def get_experiment_data(folder, filename, features, target, scale=True,
                                                          indices=False):
    """
    IMPORTS STRAIN DATASET FOR THE ROCK EXPERIMENTS FROM FOLDER AND
    CREATES DATAFRAME WITH FORMAT BASED ON TYPE OF TESTING.

    folder: The folder of the data
    filename: string with full name of file or list with string filenames
    features: Dataset features contained within the file
    target: String indicating which feature to predict
    scale: Determines if time array should be rescaled
    indices: if true return indices that splits test sets for multiple rock types
    """

    if indices == True:
        flag = False #A suboptimal solution
        for i in range(0,len(filename)):
            if filename[i-1][13] != filename[i][13]:
                flag = True

        if flag == False:
            indices = False


    if isinstance(filename,str):
        DF = pd.read_csv(folder+filename, delim_whitespace=True)
        DF = DF.dropna()

    elif isinstance(filename,list):
        DF = pd.read_csv(folder+filename[0], delim_whitespace=True)
        DF = DF.dropna()

        scaler = RobustScaler()
        DF[features] = scaler.fit_transform(DF[features].values)
        times = DF[target].values
        times = (times-min(times))/(max(times)-min(times))
        DF[target] = times

        for i in range(1,len(filename)):
            df = pd.read_csv(folder+filename[i], delim_whitespace=True)
            df = df.dropna()


            df[features] = scaler.fit_transform(df[features].values)
            times = df[target].values
            times = (times-min(times))/(max(times)-min(times))
            df[target] = times

            if indices == True:
                #print(filename[i-1][13])
                if filename[i-1][13] != filename[i][13]:
                    index_1 = len(DF) #Quick fix, not optimal!

            DF = DF.append(df, ignore_index=True)#DF = DF.append(df)



    else:
        print("filename variable must be valid type, string or list of strings")

    scaler = RobustScaler()# works better with outliers
    DF[features] = scaler.fit_transform(DF[features].values)


    if scale == True:
        times = DF[target].values
        times = (times-min(times))/(max(times)-min(times))
        DF[target] = times

        if indices == True:
            return DF, times, index_1

        else:
            return DF, times

    else:
        times = DF[target].values

        if indices == True:
            return DF, times, index_1

        else:
            return DF, times
```

We then create a function for calling and training the XGBoost library, and training a set of models to find the optimal parameters in a GridSearch:

```
def train_xgb_model(dataframe, target, features, config):
    """
    TRAINS AN XGBOOST MODEL ON A (PORTION OF) DATAFRAME
    dataframe: The data to train on
    target: String indicating which feature to predict
    features: Dataset features for the model to train on
    config: Dictionary that contains parameters, test_size and objective
    """
    xgb_model = xgb.XGBRegressor(objective=config['model_details']['objective'])
                #regression
    grid_search = GridSearchCV(estimator=xgb_model,
                  param_grid=config['parameters'], cv=10, n_jobs=-1)

    grid_search.fit(dataframe[features], dataframe[target])

    final_model = grid_search.best_estimator_

    return final_model
```

We create a function for training a neural network, this one does not use GridSearch:

```
def train_nn_model(dataframe, target, features):
    """
    TRAINS A NEURAL NETWORK ON A DATAFRAME
    dataframe: The data to train on
    target: String indicating which feature to predict
    features: Dataset features for the model to train on
    #no config for this function because no grid search
    """

    NN_model = MLPRegressor(hidden_layer_sizes=(128,64,),
               max_iter=1500, activation='tanh', alpha=0.0005)
#regression

    final_model = NN_model.fit(dataframe[features], dataframe[target])

    return final_model
```

To test any of our models, we utilize this python function:

```
def test_regression_model(model, features, target, dataset):
    """
    FUNCTION TESTS MODEL ON TEST SET
    model: Trained model to test on test set
    features: Features/Predictors in the dataset
    target: String indicating which feature to predict
    dataset: The data to test on (can send in either train, test or different set)
    """

    predicts = model.predict(dataset[features])
    rmse = np.sqrt(mean_squared_error(dataset[target], predicts))
    r2 = r2_score(dataset[target], predicts)

    scores = [len(dataset[target]), rmse, r2]
    return scores
```

And lastly we define a python function to plot the model prediction vs the observed values of strain to evaluate our model:

```
def plot_feature_vs_prediction(DataFrame, feature, prediction, plot_strings):
    """
    PLOTS THE GIVEN FEATURE AGAINST THE NUMBER OF DATAPOINTS OF THE FEATURE
    DataFrame: The dataframe
    feature: list of features
    prediction: The model prediction
    plot_strings: name of title and/or axes
    """
    x = np.linspace(0,len(DataFrame)-1,len(DataFrame))
    y = DataFrame[feature]

    plt.plot(x, prediction, 'r')
    plt.plot(x, y,'b') #plt.plot(x , y, 'ro')

    plt.legend(['prediction',feature])
    plt.ylabel(plot_strings[2])
    plt.xlabel(plot_strings[1])
    plt.title(plot_strings[0])
    plt.show()
    return
```

Note again that not all plotting functions are shown here, for instance the score matrices created using seaborn (Waskom 2021). After defining our experiments, features and targets we can now use all our functions together in this training loop where we train multiple models, print their scores and our progress, plot different metrics (not all functions shown here), save our model using pickle and save important information such as test scores:

```
print(f"\nTraining xgb models on {num_exps} experiments, testing on same dataset.
        \nStoring models for later transfer learning:\n")

for i in range(num_exps):

    print("Current experiment: %s, Completion: %d%%" %(filenames[i],(100*(i+1)/num_exps)))

    dataframe, timespan = get_experiment_data(folder, filenames[i], features, target)
    #dataframe = remove_outliers(dataframe, dataframe['ep'],
                ['dn_p50', 'dp_p50', 'cur_p50'], threshold = 0.91, verbose=True)
    dataframe.to_csv(scaled_data_folder+'scaled_'+
                        filenames[i],sep=' ', index=False)

    df_train, df_test = train_test_split(dataframe,
                        test_size=conf['model_details']['test_size'])

    plot_data_time_evolutions(dataframe, [features[2],features[11],features[20]],
                                [’Evolution of ’+features[2]+’, ’+features[11]+’,
                                ’+features[20],’Sample’,’feature’])
    plot_data_errorbars(dataframe, dataframe[’ep’],
                        [features[2],features[11],features[20]],
                        [’Evolution of ’+features[2]+’, ’+features[11]+’,
                        ’+features[20],’ep’,’feature’])

    train_time = time.time()
    model = train_xgb_model(df_train, target, features, conf)
    #other_model(df_train,target,features)
    models.append(model)

    train_time = time.time() - train_time

    with open(saved_models[i], ’wb’) as file:
        pickle.dump(model,file)

    train_scores = test_regression_model(model,features,target,df_train)
    test_scores = test_regression_model(model,features,target,df_test)

    save_test_data(model, features, target, df_test, result_folder,
                "TEST_prediction_xgb_"+experiments[i]+"_g"+rad+"0.txt")

    r2_train_vector[i] = train_scores[-1]
    r2_test_matrix[i,i] = test_scores[-1]

    print(f"Train: {train_scores}", f"\nTest: {test_scores}")

    plot_observed_vs_predicted(df_test, target, model, features,
                        [’Testing’,’Observed’,’Predicted’])

    shap_vals = represent_model_results(model,df_train,features,target,
                ’Feature impact on model ’+experiments[i]+ ’ g’+rad+’0’,
                result_folder+’Shap_vals_xgb_’+experiments[i]+’_g’+rad+’0’)

    plot_feature_vs_prediction(dataframe, ’ep’, model.predict(dataframe[features]),
        [’Evolution of ’+’ep’+’ vs prediction’,’Sample’,’prediction vs observed ep’])

    score_str+=(experiments[i]+" "+str(test_scores[1])+
                " "+str(test_scores[2]))+" "+str(train_time)+"\n"

    pred_vs_observed = np.column_stack((
                       model.predict(dataframe[features]),dataframe[’ep’]))

    np.savetxt(result_folder+model_preds[i],pred_vs_observed)

save_data_by_name(score_str,result_folder,model_scores)
```

Lastly we use our stored models to run transfer tests:

```
print("\nPerforming transfer learning, reprinting r2 scores
        \nand plotting score martix:\n")

print(len(models),num_exps)
for i in range(len(models)):
    for j in range(num_exps):
        if j != i:
            dataframe, timespan =
              get_experiment_data(folder, filenames[j], features, target)
            scores =
              test_regression_model(models[i], features, target,dataframe)
            r2_test_matrix[i,j] = scores[-1]

            if abs(r2_test_matrix[i,j]) > 0.7:
                print(f"training data: {experiments[i]},
                testing data: {experiments[j]}, score: {r2_test_matrix[i,j]}")
                plot_feature_vs_prediction(dataframe, ’ep’,
                  models[i].predict(dataframe[features]),
                  [’Evolution of ’+’ep’+’ vs prediction’,
                  ’Sample’,’prediction vs observed ep’])

    print(f"Train score on experiment {experiments[i]}:
          r2 = {r2_train_vector[i]}\nTest score: r2 = {r2_test_matrix[i,i]}")

plot_sns_score_matrix(r2_test_matrix,experiments,experiments,
["XGB Test R2 Score g50", "Testing Data", "Training Data"],
savename=’../Figures1/strain_single_transfer_xgb_g50’)

np.savetxt(result_folder+model_score_matrix,r2_test_matrix)
```

The transfer testing loop is fairly similar for the transfer learning where we use multiple data sorted by rock types. However the training loop is much more complicated, so we will show it here:

```
num_models = 0

r2_train_vector = np.zeros(len(rock_list))

comb_len = sum([len(rock_list)-i for i in range(len(rock_list))])
r2_rock_matrix = np.zeros((comb_len,len(rock_list)))

print(f"\nTraining xgb models on {comb_len} combinations of rocktypes,
testing on each dataset. \nStoring models for later transfer learning:\n")

for i in range(len(rock_list)):
    for j in range(i,len(rock_list)):
        #print(i,j)
```

```
if j == i:
    #train the model on single rock type and store test score
    print(f"Training on and testing on single rock type: {rock_names[i]}")


    filenames = [’strains_curr_’+experiment+’_g’+rad+’0.txt’
                for experiment in rock_list[i]]
    print(filenames)

    dataframe, timespan =
        get_experiment_data(folder, filenames, features, target)

    combined_names.append(rock_names[i])


else:
    #train the model on multiple rock types and store test score
    print(f"Combining experiment: {rock_names[i]} and {rock_names[j]}")

    filenames1 = [’strains_curr_’+experiment+’_g’+rad+’0.txt’
                 for experiment in rock_list[i]]
    filenames2 = [’strains_curr_’+experiment+’_g’+rad+’0.txt’
                 for experiment in rock_list[j]]
    print(filenames1+filenames2)

    dataframe, timespan, index_1 = get_experiment_data(
        folder, filenames1+filenames2, features, target, indices=True)

    rock_str = rock_names[i] + " and " + rock_names[j]
    combined_names.append(rock_str)

df_train, df_test = train_test_split(dataframe,
                    test_size=conf[’model_details’][’test_size’])


plot_data_time_evolutions(dataframe, [features[2],features[11],features[20]],
                        [’Evolution of ’+features[2]+’, ’+features[11]+’, ’
                        +features[20],’Sample’,’feature’])

train_time = time.time()
model = train_xgb_model(df_train, target, features, conf)
#other_model(df_train,target,features)
model_permutations.append(model)
train_time = time.time() - train_time

#with open(saved_models[i], ’wb’) as file:
#    pickle.dump(model,file)
if j == i:
    shap_vals = represent_model_results(model,df_train,features,target,
        'Feature impact on '+rock_names[i]+
        ' xgb model g'+rad+'0', result_folder+
        'rock_type_transfer_shap_vals_xgb_'+rock_names[i]+'_g'+rad+'0')


train_scores = test_regression_model(model,features,target,df_train)
r2_train_vector[i] = train_scores[-1]

if j == i:
    test_scores = test_regression_model(model,features,target,df_test)
    r2_rock_matrix[num_models,i] = test_scores[-1]
    print(f"Train on {rock_names[i]} with training
        score: {r2_train_vector[i]}", f"\nTest on {rock_names[i]}
        with score: {r2_rock_matrix[num_models,i]}")

else:
    ind_list1 = []
    ind_list2 = []
    for ind in df_test.index: #store index array which splits the test points
        if ind < index_1:
            #print(ind)
            ind_list1.append(ind)
        else:
            ind_list2.append(ind)

    print("Checking train test split:", f"Points in {rock_names[i]}:
        {len(ind_list1)},", f"Points in {rock_names[j]}: {len(ind_list2)},",
         f"\nSplit at 20%, Proportion of points:
        {(len(ind_list1)+len(ind_list2))/len(dataframe)}" )

    df_test1 = df_test.loc[ind_list1]
    df_test2 = df_test.loc[ind_list2]

    test_scores1 = test_regression_model(model,features,target,df_test1)
    test_scores2 = test_regression_model(model,features,target,df_test2)

    r2_rock_matrix[num_models,i] = test_scores1[-1]
    r2_rock_matrix[num_models,j] = test_scores2[-1]

    print(f"Train on {rock_names[i]} and {rock_names[j]} with score:
        {r2_train_vector[i]}", f"\nTest on {rock_names[i]}:
        {r2_rock_matrix[num_models,i]}", f"\nTest on {rock_names[j]}:
        {r2_rock_matrix[num_models,j]}")


num_models += 1

plot_feature_vs_prediction(dataframe, 'ep', model.predict(dataframe[features]),
['Evolution of '+'ep'+' vs prediction','Sample','prediction vs observed ep'])
```

Bibliography

Aben, FM, M-L Doan, TM Mitchell, R Toussaint, T Reuschlé, Michele Fondriest, J-P Gratier and F Renard (2016). 'Dynamic fracturing by successive coseismic loadings leads to pulverization in active fault zones'. In: *Journal of Geophysical Research: Solid Earth* 121.4, pp. 2338–2360.

Amoruso, A and L Crescentini (2010). 'Limits on earthquake nucleation and other pre-seismic phenomena from continuous strain in the near field of the 2009 L'Aquila earthquake'. In: *Geophysical research letters* 37.10.

Bakun, WH, B Aagaard, B Dost, WL Ellsworth, JL Hardebeck, RA Harris, C Ji, MJS Johnston, J Langbein, JJ Lienkaemper et al. (2005). 'Implications for prediction and hazard assessment from the 2004 Parkfield earthquake'. In: *Nature* 437.7061, pp. 969–974.

Baud, Patrick, Alexandre Schubnel and Teng-fong Wong (2000). 'Dilatancy, compaction, and failure mode in Solnhofen limestone'. In: *Journal of Geophysical Research: Solid Earth* 105.B8, pp. 19289–19303.

Baud, Patrick, Veronika Vajdova and Teng-fong Wong (2006). 'Shear-enhanced compaction and strain localization: Inelastic deformation and constitutive modeling of four porous sandstones'. In: *Journal of Geophysical Research: Solid Earth* 111.B12.

Bignami, Christian, Emanuela Valerio, Eugenio Carminati, Carlo Doglioni, Pietro Tizzani and Riccardo Lanari (2019). 'Volume unbalance on the 2016 Amatrice-Norcia (Central Italy) seismic sequence and insights on normal fault earthquake mechanism'. In: *Scientific reports* 9.1, pp. 1–13.

Bourbie, Thierry and Bernard Zinszner (1985). 'Hydraulic and acoustic properties as a function of porosity in Fontainebleau sandstone'. In: *Journal of Geophysical Research: Solid Earth* 90.B13, pp. 11524–11532.

Brace, WF (1978). 'Volume changes during fracture and frictional sliding: A review'. In: *Pure and Applied Geophysics* 116.4, pp. 603–614.

Brace, WF, BW Paulding Jr and CH Scholz (1966). ‘Dilatancy in the fracture of crystalline rocks’. In: *Journal of Geophysical Research* 71.16, pp. 3939–3953.

Brantut, Nicolas, MJ Heap, PG Meredith and Patrick Baud (2013). ‘Time-dependent cracking and brittle creep in crustal rocks: A review’. In: *Journal of Structural Geology* 52, pp. 17–43.

Chen, Tianqi and Carlos Guestrin (2016). ‘Xgboost: A scalable tree boosting system’. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Corbi, Fabio, Laura Sandri, Jonathan Bedford, Francesca Funiciello, Silvia Brizzi, Matthias Rosenau and Serge Lallemand (2019). ‘Machine learning can predict the timing and size of analog earthquakes’. In: *Geophysical Research Letters* 46.3, pp. 1303–1311.

Developers, XGBoost (2019). *XGBoost Documentation.*

Fredrich, JT, KH Greaves and JW Martin (1993). ‘Pore geometry and transport properties of Fontainebleau sandstone’. In: *International journal of rock mechanics and mining sciences & geomechanics abstracts.* Vol. 30. 7. Elsevier, pp. 691–697.

Friedman, Jerome, Trevor Hastie, Robert Tibshirani et al. (2000). ‘Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)’. In: *Annals of statistics* 28.2, pp. 337–407.

Goodfellow, SD, N Tisato, MNMHB Ghofranitabari, MHB Nasseri and RP Young (2015). ‘Attenuation properties of Fontainebleau sandstone during true-triaxial deformation using active and passive ultrasonics’. In: *Rock Mechanics and Rock Engineering* 48.6, pp. 2551–2566.

Griffith, Alan Arnold (1921). ‘VI. The phenomena of rupture and flow in solids’. In: *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character* 221.582-593, pp. 163–198.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media. Chap. 2,7.

Heap, MJ, S Vinciguerra and PG Meredith (2009). ‘The evolution of elastic moduli with increasing crack damage during cyclic stressing of a basalt from Mt. Etna volcano’. In: *Tectonophysics* 471.1-2, pp. 153–160.

Holdsworth, RE, CA Butler and AM Roberts (1997). ‘The recognition of reactivation during continental deformation’. In: *Journal of the Geological Society* 154.1, pp. 73–78.

Hornik, Kurt, Maxwell Stinchcombe and Halbert White (1989). ‘Multilayer feedforward networks are universal approximators’. In: *Neural networks* 2.5, pp. 359–366.

Huang, Lingcao, Patrick Baud, Benoit Cordonnier, Francois Renard, Lin Liu and Teng-fong Wong (2019). 'Synchrotron X-ray imaging in 4D: multiscale failure and compaction localization in triaxially compressed porous limestone'. In: *Earth and Planetary Science Letters* 528, p. 115831.

Hulbert, Claudia, Bertrand Rouet-Leduc, Paul A Johnson, Christopher X Ren, Jacques Rivière, David C Bolton and Chris Marone (2019). 'Similarity of fast and slow earthquakes illuminated by machine learning'. In: *Nature Geoscience* 12.1, pp. 69–74.

Katz, Oded and Ze'ev Reches (2004). 'Microfracturing, damage, and failure of brittle granites'. In: *Journal of Geophysical Research: Solid Earth* 109.B1.

Kerrick, DM (1969). 'K-feldspar megacrysts from a porphyritic quartz monzonite central Sierra Nevada, California'. In: *American Mineralogist: Journal of Earth and Planetary Materials* 54.5-6, pp. 839–848.

Kingma, Diederik P and Jimmy Ba (2014). 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980*.

Labuz, Joseph F and Arno Zang (2012). 'Mohr–Coulomb failure criterion'. In: *The ISRM Suggested Methods for Rock Characterization, Testing and Monitoring: 2007-2014*. Springer, pp. 227–231.

Lash, Gary G and Terry Engelder (2005). 'An analysis of horizontal microcracking during catagenesis: Example from the Catskill delta complex'. In: *AAPG bulletin* 89.11, pp. 1433–1449.

LeNail, Alexander (2019). 'NN-SVG: Publication-Ready Neural Network Architecture Schematics'. In: *Journal of open source software* 4.33, p. 747.

Lion, Maxime, Frédéric Skoczylas and Béatrice Ledésert (2005). 'Effects of heating on the hydraulic and poroelastic properties of bourgogne limestone'. In: *International Journal of Rock Mechanics and Mining Sciences* 42.4, pp. 508–520.

Lockner, David A (1998). 'A generalized law for brittle deformation of Westerly granite'. In: *Journal of Geophysical Research: Solid Earth* 103.B3, pp. 5107–5123.

Lundberg, Scott and Su-In Lee (2017). 'A unified approach to interpreting model predictions'. In: *arXiv preprint arXiv:1705.07874*.

Main, Ian G, Andrew F Bell, Philip G Meredith, Sebastian Geiger and Sarah Touati (2012). 'The dilatancy–diffusion hypothesis and earthquake predictability'. In: *Geological Society, London, Special Publications* 367.1, pp. 215–230.

McBeck, Jessica Ann, John Mark Aiken, Yehuda Ben-Zion and Francois Renard (2020a). 'Predicting the proximity to macroscopic failure using local strain populations from dynamic in situ X-ray tomography triaxial compression experiments on rocks'. In: *Earth and Planetary Science Letters* 543, p. 116344.

McBeck, Jessica Ann, John Mark Aiken, Joachim Mathiesen, Yehuda Ben-Zion and Francois Renard (2020b). 'Deformation precursors to catastrophic failure in rocks'. In: *Geophysical Research Letters* 47.24, e2020GL090255.

McBeck, Jessica Ann, Yehuda Ben-Zion and Francois Renard (2020). 'The mixology of precursory strain partitioning approaching brittle failure in rocks'. In: *Geophysical Journal International* 221.3, pp. 1856–1872.

McBeck, Jessica Ann, Benoıt Cordonnier, Sergio Vinciguerra and Francois Renard (2019). 'Volumetric and shear strain localization in Mt. Etna basalt'. In: *Geophysical Research Letters* 46.5, pp. 2425–2433.

McBeck, Jessica Ann, Maya Kobchenko, Stephen A Hall, Erika Tudisco, Benoit Cordonnier, Paul Meakin and Francois Renard (2018). 'Investigating the onset of strain localization within anisotropic shale using digital volume correlation of time-resolved X-ray microtomography images'. In: *Journal of Geophysical Research: Solid Earth* 123.9, pp. 7509–7528.

Mjachkin, VI, WF Brace, GA Sobolev and JH Dieterich (1975). 'Two models for earthquake forerunners'. In: *Earthquake prediction and rock mechanics.* Springer, pp. 169–181.

Moro, Marco, Michele Saroli, Salvatore Stramondo, Christian Bignami, Matteo Albano, Emanuela Falcucci, Stefano Gori, Carlo Doglioni, Marco Polcari, Marco Tallini et al. (2017). 'New insights into earthquake precursors from InSAR'. In: *Scientific reports* 7.1, pp. 1–11.

Nasseri, MHB, SD Goodfellow, L Lombos and RP Young (2014). '3-D transport and acoustic properties of Fontainebleau sandstone during true-triaxial deformation experiments'. In: *International Journal of Rock Mechanics and Mining Sciences* 69, pp. 1–18.

Nielsen, Michael A (2015). *Neural networks and deep learning.* Vol. 25. Determination press San Francisco, CA. Chap. 1,2.

Nur, Amos (1972). 'Dilatancy, pore fluids, and premonitory variations of ts/tp travel times'. In: *Bulletin of the Seismological society of America* 62.5, pp. 1217–1222.

— (1974). 'Matsushiro, Japan, earthquake swarm: Confirmation of the dilatancy-fluid diffusion model'. In: *Geology* 2.5, pp. 217–221.

Pan, Sinno Jialin and Qiang Yang (2009). 'A survey on transfer learning'. In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Paterson, Mervyn S and Teng-fong Wong (2005). *Experimental rock deformation-the brittle field.* Springer Science & Business Media, pp. 45–47, 68–76, 111.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg et al. (2011). 'Scikit-learn: Machine learning in Python'. In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Reches, Ze'ev and David A Lockner (1994). 'Nucleation and growth of faults in brittle rocks'. In: *Journal of Geophysical Research: Solid Earth* 99.B9, pp. 18159–18173.

Renard, Francois (2017). 'Critical evolution of damage toward system-size failure in crystalline rock [Data set]'. In: *Norstore*. DOI: https://doi.org/10.11582/2017.00025.

— (2018a). 'Dynamic in situ three-dimensional imaging and digital volume correlation analysis quantify strain localization and fracture coalescence in sandstone [Data set]'. In: *Norstore*. DOI: https://doi.org/10.11582/2018.00022.

— (2018b). 'Volumetric and shear processes in crystalline rock during the approach to faulting [Data set]'. In: *Norstore*. DOI: https://doi.org/10.11582/2018.00023.

— (2018c). 'Volumetric and shear strain localization in Mt. Etna basalt [Data set]'. In: *Norstore*. DOI: https://doi.org/10.11582/2018.00036.

Renard, Francois, Benoit Cordonnier, Dag K Dysthe, Elodie Boller, Paul Tafforeau and Alexander Rack (2016). 'A deformation rig for synchrotron microtomography studies of geomaterials under conditions down to 10 km depth in the Earth'. In: *Journal of synchrotron radiation* 23.4, pp. 1030–1034.

Renard, Francois, Benoıt Cordonnier, Maya Kobchenko, Neelima Kandula, Jérôme Weiss and Wenlu Zhu (2017). 'Microscale characterization of rupture nucleation unravels precursors to faulting in rocks'. In: *Earth and Planetary Science Letters* 476, pp. 69–78.

Renard, Francois and Jessica Ann McBeck (2018). 'Investigating the onset of strain localization within anisotropic shale using digital volume correlation of time-resolved X-ray microtomography images [data set]'. In: *Norstore*. DOI: https://doi.org/10.11582/2018.00005.

Renard, Francois, Jessica Ann McBeck, Benoıt Cordonnier, Xiaojiao Zheng, Neelima Kandula, Jesus R Sanchez, Maya Kobchenko, Catherine Noiriel, Wenlu Zhu, Paul Meakin et al. (2019). 'Dynamic in situ three-dimensional imaging and digital volume correlation analysis to quantify strain localization and fracture coalescence in sandstone'. In: *Pure and Applied Geophysics* 176.3, pp. 1083–1115.

Renard, Francois, Jérôme Weiss, Joachim Mathiesen, Yehuda Ben-Zion, Neelima Kandula and Benoıt Cordonnier (2018). 'Critical evolution of damage toward system-size failure in crystalline rock'. In: *Journal of Geophysical Research: Solid Earth* 123.2, pp. 1969–1986.

Ronneberger, Olaf, Philipp Fischer and Thomas Brox (2015). 'U-net: Convolutional networks for biomedical image segmentation'. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rouet-Leduc, Bertrand, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J Humphreys and Paul A Johnson (2017). 'Machine learning predicts laboratory earthquakes'. In: *Geophysical Research Letters* 44.18, pp. 9276–9282.

Scholz, Christopher H, Lynn R Sykes and Yash P Aggarwal (1973). 'Earthquake prediction: A physical basis'. In: *Science* 181.4102, pp. 803–810.

Shimodaira, Hidetoshi (2000). 'Improving predictive inference under covariate shift by weighting the log-likelihood function'. In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Sibson, Richard H (1985). 'Stopping of earthquake ruptures at dilational fault jogs'. In: *Nature* 316.6025, pp. 248–251.

Sigrist, Fabio (2021). 'Gradient and newton boosting for classification and regression'. In: *Expert Systems With Applications* 167, p. 114080.

Skelton, Alasdair, Margareta Andrén, Hrefna Kristmannsdóttir, Gabrielle Stockmann, Carl-Magnus Mörth, Árny Sveinbjörnsdóttir, Sigurjón Jónsson, Erik Sturkell, Helga Rakel Guðrúnardóttir, Hreinn Hjartarson et al. (2014). 'Changes in groundwater chemistry before two consecutive earthquakes in Iceland'. In: *Nature Geoscience* 7.10, pp. 752–756.

Tapponnier, Paul and WF Brace (1976). 'Development of stress-induced microcracks in Westerly granite'. In: *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*. Vol. 13. 4. Elsevier, pp. 103–112.

Tudisco, Erika, Edward Andò, Rémi Cailletaud and Stephen A Hall (2017). 'TomoWarp2: A local digital volume correlation code'. In: *SoftwareX* 6, pp. 267–270.

Verberne, Berend A, Jianye Chen, André R Niemeijer, Johannes HP de Bresser, Gillian M Pennock, Martyn R Drury and Christopher J Spiers (2017). 'Microscale cavitation as a mechanism for nucleating earthquakes at the base of the seismogenic zone'. In: *Nature communications* 8.1, pp. 1–8.

Vinciguerra, Sergio, Concetta Trovato, Philip G Meredith and Philip M Benson (2005). 'Relating seismic velocities, thermal cracking and permeability in Mt. Etna and Iceland basalts'. In: *International Journal of Rock Mechanics and Mining Sciences* 42.7-8, pp. 900–910.

Waskom, Michael L (2021). ‘Seaborn: statistical data visualization’. In: *Journal of Open Source Software* 6.60, p. 3021.

Weiss, Karl, Taghi M Khoshgoftaar and DingDing Wang (2016). ‘A survey of transfer learning’. In: *Journal of Big data* 3.1, pp. 1–40.

Wyss, Max (2001). ‘Why is earthquake prediction research not progressing faster?’ In: *Tectonophysics* 338.3-4, pp. 217–223.

Wyss, Max and David C Booth (1997). ‘The IASPEI procedure for the evaluation of earthquake precursors’. In: *Geophysical Journal International* 131.3, pp. 423–424.

Zhu, Wei, Patrick Baud, Sergio Vinciguerra and Teng-fong Wong (2016). ‘Micromechanics of brittle faulting and cataclastic flow in Mount Etna basalt’. In: *Journal of Geophysical Research: Solid Earth* 121.6, pp. 4268–4289.

Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong and Qing He (2020). ‘A comprehensive survey on transfer learning’. In: *Proceedings of the IEEE* 109.1, pp. 43–76.