Sam Burton and Sofia Catalan

CS545

Professor Ben-Hur

3 November 2025

Seismic Event Classification with CNN and Transformer

Intro

We will train and evaluate both CNN and Transformer models using continuous broadband seismic data from Mount Erebus, focusing on event classification into 18 different eruption types, including "no eruption." We aim to explore whether the Transformer's long attention mechanism can capture the long-term temporal features of eruption data that improve classification accuracy and interpretability. Accurately classifying these signals can contribute to a more robust volcanic event detection framework at Erebus, and eventually other volcanoes.

Dataset

Each training example is a 20-second seismic time series of a volcanic eruption signal from Mount Erebus Volcano. Time series were collected from the 200-station three component 2007-2009 Tomo Erebus Experiment. The final dataset used in this study has ~1,200,000 training examples from 18 clusters (17 known event types, and a "no event" cluster). Each row will contain a label (0-17) in column 0, and then, in columns *1-4001*, a 20-second time series of seismic data sampled at 200 Hz.

Raw signals are pre-processed using standard signal processing techniques such as a 1-10Hz bandpass filter (filters out the less-informative high and low frequencies), de-meaning and normalization by mean absolute deviation (reduces large amplitude discrepancies between stations), and de-trending and dividing by instrument sensitivity (remove instrument response and convert data to units velocity). 17 known eruption signals are cataloged from December 10th–27th, 2008, where waveforms of the same class have highly similar sources (size, origin) and wave propagation paths (path from source to instrument).

Data Exploration and Preprocessing

Before preprocessing, we would like to perform some data exploration on the dataset in order to understand its characteristics. To do that, we will graph 3-5 sample waveforms per class, and then we will analyze the similarities that exist among waveforms belonging to the same class, as well as the differences that exist across waveforms belonging to different classes. In this section, we will also confirm if the class distribution of the dataset is either balanced or unbalanced.

Next, we will be splitting the dataset into train, test, and validation sets. Given that we have a lot of samples available to us, we will define the split to be 80/10/10. Also, due to the high dimensionality of the data, we may perform dimensionality reduction using either PCA or an autoencoder. This will allow us to see if reducing the dimensionality of the data will improve the performance of the model or at least achieve similar results with less training time.

Baseline Model

For the baseline model, we will be implementing a CNN from scratch using PyTorch. We chose a CNN as the baseline model because it can handle time-series data, and it would also be able to detect local patterns in the waveforms, although it may struggle to capture long-term dependencies.

Proposed Model

We will be implementing a Transformer from scratch using PyTorch and evaluating its performance against the CNN. We believe this model will perform significantly better since transformers can capture long-term patterns and dependencies.

For both models, we will be tracking the RMSE loss curves of the train and validation sets to detect if any overfitting occurs.

Evaluation Metrics

Since this is a classification task, we will be evaluating the models using accuracy, F1 score, precision, and recall. We will also visualize where the models get confused by graphing a confusion matrix.

Task Split

Objective	**Tentative Due Dates**	**Teammate**
Data Preprocessing–- - Waveform extraction and preprocessing using event catalog - Normalization - Data split	11/7	Sam
Data Exploration – - verifying class distribution, - plotting waveforms	11/7	Sofia
PCA	11/17	Sofia
Baseline Model - CNN	11/9	Sofia
Transformer Development	11/29	Sofia and Sam
Evaluation Metrics: functions for - Accuracy - f-1 score - Recall - Precision - Confusion matrix	11/15	Sofia
Helper Functions for Model: - Loss curve plotting - Hyper-parameter tuning	11/15	Sam