Permutation Entropy for Volcanic Forecasting (scikit-learn baseline)

Martín Ramírez Espinosa, Sergio Alejandro González Osorio

December 10, 2025

Why permutation entropy?

- Seismic waveforms change complexity before energetic events; PE captures this with minimal assumptions.
- Lightweight: counts ordinal patterns, no heavy filtering or feature engineering.
- Robust: deterministic tie-breaking makes PE stable on quantized or low-SNR data.

Theory: PE, WPE, MPE

For embedding dimension m and delay τ:

1. Build ordinal patterns from $[x_t, x_{t+\tau}, \ldots, x_{t+(m-1)\tau}]$ using value-first, index-second ordering.
2. Estimate the pattern distribution p and compute $H = -\sum p \log p$.
3. Normalize by $\log(m!)$ to keep $PE \in [0, 1]$.

Weighted PE multiplies counts by local variance (down-weights flat windows). Multiscale PE repeats across several τ values to capture multi-rate dynamics.

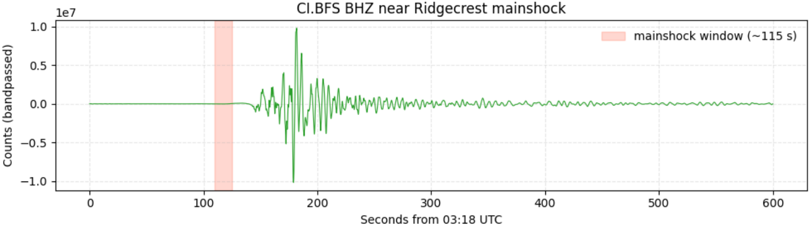
Implementation in this repo

- Core algorithms: `src/permutation_entropy/features.py` (PE/WPE/MPE, deterministic ordinal patterns, sliding windows).
- Feature extraction CLI: `data_features.ingest` (MiniSEED $\rightarrow$ CSV of PE features with aligned start times).
- Model helper: `permutation_entropy.models` (balanced logistic regression, metrics, probabilities).
- Training CLI: `bin/train_pe.py` (loads CSV, stratified split when possible, saves probability plot).

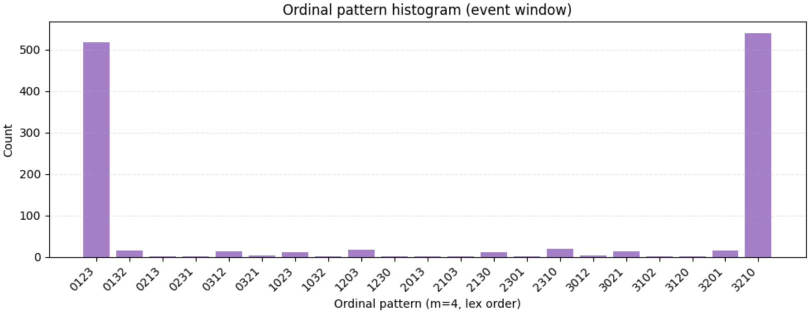
Case study: Ridgecrest mainshock

- Data: IRIS MiniSEED around 2019-07-06 Ridgecrest mainshock (IU.ANMO, CI.BFS for closer view).
- Windows: 30 s length, 5 s hop; labels mark 1180–1220 s from 03:00 UTC.
- Features: PE, WPE, multiscale PE ($\tau = 1..4$); labels recreated if missing in CSV.
- Model: balanced logistic regression; evaluate on a held-out split to avoid optimistic metrics.

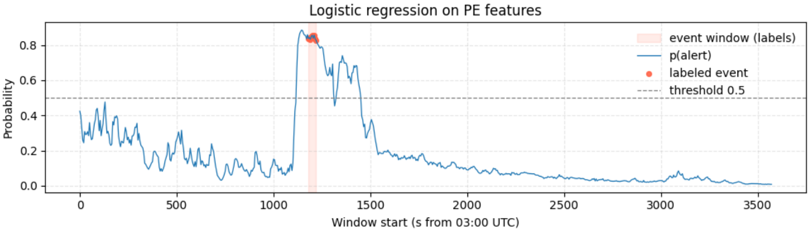
Waveform preview (CI.BFS, closer station)



Ordinal patterns in a window



Probabilities over time (example)



Pipeline summary

1. Download or load MiniSEED; resolve data paths automatically in notebooks/CLI.
2. Segment waveform into overlapping windows; compute PE/WPE/MPE per window.
3. Save CSV with `start_sec` for plotting and `label` for supervised training.
4. Train logistic regression; plot alert probabilities against time to inspect separability.

Takeaways and next steps

- PE-based features provide a fast, transparent baseline for eruption/tremor alerting.
- Deterministic patterns and balanced logistic regression reduce brittleness on small datasets.
- Next: add cross-validation, threshold calibration, and spectral features; package for pip + CI.