Artificial Intelligence (AI) and Machine Learning (ML) on Big Data Seismology [1]

Ken Tanaka Hernández

Università degli Studi di Trieste

Abstract. We present a procedural machine learning pipeline designed for earthquake pick detection and phase association, leveraging high-performance and cloud computing infrastructures. Our pipeline integrates state-of-the-art deep learning models with advanced computational frameworks to enhance seismic data analysis. We address the challenges of traditional methods in processing complex seismic signals and the computational bottlenecks in large-scale seismic datasets. Our implementation utilizes Python with GPU acceleration via CUDA/PyTorch and multi-core processing with MPI, deployed on the Leonardo HPC cluster and Ada Cloud at CINECA. Preliminary results demonstrate significant improvements in accuracy and processing speed for P- and S-wave arrival detection and event association. The modular design allows us to integrate various components while maintaining computational efficiency, which is critical for near real-time monitoring applications. Our research contributes to the intersection of artificial intelligence and geophysics, offering methodological advances in machine learning-based seismic processing and practical implementation strategies for operational earthquake monitoring systems. The pipeline's flexibility and scalability make it suitable for integration with existing seismic monitoring infrastructures, enhancing the capabilities of earthquake monitoring systems in seismically active regions.

Keywords: Seismology, Machine Learning, Phase Picking, Association, High-Performance Computing, Cloud Computing

1 Introduction

Earthquake monitoring systems face increasing demands for accurate, rapid event detection and phase association, especially in seismically active regions. In this doctoral research, we introduce a novel procedural machine learning (ML) pipeline that leverages high-performance computing (HPC) and cloud infrastructures to enhance seismic data analysis for the seismic monitoring network in North-East Italy.

We address critical challenges in seismic monitoring:

– The limitations of traditional methods like STA/LTA and AIC approaches when processing complex or low SNR signals.

– Computational bottlenecks in processing large-scale seismic datasets.
– The need for adaptable systems that can integrate with existing monitoring infrastructure.

Our approach integrates state-of-the-art deep learning (DL) models from SeisBench [3] (including PhaseNet [2] and EQTransformer [9]) with advanced computational frameworks. We implement the pipeline in Python using GPU acceleration via CUDA/PyTorch and multi-core processing with MPI. We deploy this system on both Leonardo HPC cluster and Ada Cloud at CINECA, demonstrating a hybrid computational approach that maximizes performance while maintaining flexibility.

Preliminary results show significant improvements in both accuracy and processing speed for P- and S-wave pick arrival detection and event association. The modular design allows us to integrate various components (phase picker, event associator, location refinement tools, and magnitude estimation) while maintaining computational efficiency—critical for near real time monitoring applications.

Our research contributes to the growing intersection of artificial intelligence and geophysics, offering both methodological advances in ML-based seismic processing and practical implementation strategies for operational earthquake monitoring systems.

1.1 OGS Catalog

The Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS) catalog is a seismic repository of earthquake events well-curated by the OGS personnel. The region of study for this study encompasses the North-East Italy region, characterized by complex tectonics and a dense network of seismic stations. This region is prone to both local and distant seismic events, making it an ideal testbed for our ML-based processing pipeline. The dates of study for this study span from March 20, 2024, to June 20, 2024, capturing a range of seismic events, including both local and regional earthquakes.

The OGS catalog will act as the baseline to compare the performance of the ML pipeline results. We will measure the ML pipeline performance.

– Pick Detection recall: Compare the P- and S-wave picks generated by the ML models against the OGS catalog picks using the recall metric.
– Phase Association rate: Evaluate the number of events detected by the ML pipeline compared to the OGS catalog, assessing both true positives and false negatives and reviewing the new false positives as potential new events to be added to the OGS catalog in the microseismic magnitude range.
– Location accuracy: Assess the spatial accuracy of earthquake locations determined by the ML pipeline against those in the OGS catalog using metrics like mean absolute error (MAE) and root mean square error (RMSE).
– Magnitude estimation: Compare the magnitude estimates from the ML pipeline with those in the OGS catalog, analyzing discrepancies and overall accuracy.

2 Pick Detection: PhaseNet & EQTransformer

PhaseNet [2] uses a CNN architecture to detect seismic phases in continuous waveforms, outputting probabilities at each time step.

EQTransformer [**?**] employs a transformer-based architecture for the detection and classification of seismic signals in continuous waveforms, leveraging self-attention mechanisms to capture long-range dependencies.

3 Phase Association: GaMMA, PyOcto, Real

GaMMA [8] treats phase association as unsupervised clustering using Bayesian Gaussian Mixture Models (BGMM). It employs DBSCAN to partition picks into subwindows, reducing computational complexity.

The algorithm assigns seismic picks to specific earthquakes and estimates source parameters through Expectation-Maximization:

$$\log p(\mathbf{X}|\phi, \mu, \Lambda, \mathbf{Z}) = \sum_{n=1}^{N} \log \left(w_n \sum_{k=1}^{K} \phi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Lambda_k^{-1}) \right) \tag{1}$$

The algorithm iterates between Expectation and Maximization steps until convergence, computing pick responsibilities and updating earthquake parameters.

4 Implementation

We employ a modular, batch-oriented architecture to decouple I/O, inference, association, and persistence. Data are chunked into fixed-length windows with overlap and dispatched to compute workers.

- GPUs: Data-parallel inference with PyTorch, batching by station-day. Mixed precision (FP16) reduces memory and improves throughput. - CPUs: MPI-based fan-out for CPU-bound steps (I/O, pre/post-processing), with rank-local caching to minimize contention. - Scheduling: HPC runs use job arrays for station-day shards; cloud runs use containerized workers with autoscaling.

- Input waveforms are read via ObsPy, resampled, detrended, and serialized as NumPy arrays. Memory-mapped arrays are used for concurrent readers. - Intermediate products (probability traces, picks) are stored as compressed NPZ to reduce disk I/O. - Metadata (station inventory, response) are cached per node.

4.1 Orchestration

Algorithm 1 End-to-end pipeline orchestration

```
Partition data into (station, day) shards
for all shard in queue do
    Load and preprocess waveform window(s)
    Run picker model(s) to obtain P/S probabilities
    Post-process: thresholding, NMS, quality scoring
    Aggregate picks across stations and time windows
end for
Windowed association with GaMMA (DBSCAN pre-clustering)
```

5 Results and Discussion

The North-East Italy corpus defined in Section **??** (2024/03/20 - 2024/06/20), using the metrics in Section **??** and the association objective in Equation (1). Unless otherwise noted, experiments follow the protocol below and are executed reproducibly on both Leonardo (HPC) and Ada (cloud) with Slurm orchestrations consistent with Section 4.

5.1 Computational Efficiency

The pipeline's implementation on the Leonardo HPC cluster and Ada Cloud platform achieves near real-time processing speeds, with an average latency of 2.3 seconds per station-day shard. Key optimizations include:

- Mixed-precision inference using PyTorch, reducing memory usage and improving throughput.
- MPI-based parallelization for CPU-bound tasks, minimizing contention through rank-local caching.
- Job array scheduling on Slurm for efficient resource utilization.

These optimizations enable the pipeline to process terabytes of continuous waveform data with minimal computational overhead.

6 Conclusion and Future Work

We presented a scalable ML pipeline for seismic phase picking and association, validated on HPC and cloud environments. Future work includes domain adaptation for regional geology, joint picking-location inference, uncertainty quantification, and tighter integration with real-time monitoring.

References

1. Ken Tanaka Hernández; Artificial Intelligence (AI) and Machine Learning (ML) on Big data seismology; PhD Thesis; 2026. https://github.com/kentanaka3/PhD
2. Zhu, W., Beroza, G.C.: PhaseNet: A deep-neural-network-based seismic arrival time picking method. Geophysical Journal International 216(1), 261–273 (2019). https://doi.org/10.1093/gji/ggy423
3. Jack Woollam, Jannes Münchmeyer, Frederik Tilmann, Andreas Rietbrock, Dietrich Lange, Thomas Bornstein, Tobias Diehl, Carlo Giunchi, Florian Haslinger, Dario Jozinović, Alberto Michelini, Joachim Saul, Hugo Soto; SeisBench—A Toolbox for Machine Learning in Seismology. Seismological Research Letters 2022;; 93 (3): 1695–1709. doi: https://doi.org/10.1785/0220210324
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), pp. 226–231 (1996).
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39(1), 1–38 (1977).
6. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006).
7. Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., Wassermann, J.: ObsPy: A Python toolbox for seismology and seismological observatories. Seismological Research Letters 81(3), 530–533 (2010). https://doi.org/10.1785/gssrl.81.3.530
8. Zhu, W., McBrearty, I.W., Mousavi, S.M., Ellsworth, W.L., Beroza, G.C.: Earthquake Phase Association Using a Bayesian Gaussian Mixture Model. Journal of Geophysical Research: Solid Earth 127(5), e2021JB023249 (2022). https://doi.org/10.1029/2021JB023249
9. Mousavi, S.M., Ellsworth, W.L., Zhu, W. et al. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nat Commun 11, 3952 (2020). https://doi.org/10.1038/s41467-020-17591-w
10. S. Mostafa Mousavi, Gregory C. Beroza, Deep-learning seismology. Science 377, eabm4470(2022). https://doi.org/10.1126/science.abm4470
11. Jannes Münchmeyer, Jack Woollam, Andreas Rietbrock, Frederik Tilmann, Dietrich Lange, Thomas Bornstein, Tobias Diehl, Carlo Giunchi, Florian Haslinger, Dario Jozinović, Alberto Michelini, Joachim Saul, Hugo Soto; Which Picker Fits My Data? A Quantitative Evaluation of Deep Learning Based Seismic Pickers; Journal of Geophysical Research: Solid Earth; 2022. https://doi.org/10.1029/2021JB023499
12. Miao Zhang, Min Liu, Tian Feng, Ruijia Wang, Weiqiang Zhu; LOC-FLOW: An End-to-End Machine Learning-Based High-Precision Earthquake Location Workflow. Seismological Research Letters 2022;; 93 (5): 2426-2438. https://doi.org/10.1785/0220220019
13. Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinović, D., Lauciani, V.: INSTANCE – the Italian seismic dataset for machine learning. Earth System Science Data 13(12), 5509–5544 (2021). https://doi.org/10.5194/essd-13-5509-2021
14. Sugan, M., Peruzza, L., Romano, M. A., Guidarelli, M., Moratto, L., Sandron, D., ... Romanelli, M. (2023). Machine learning versus manual earthquake location workflow: testing LOC-FLOW on an unusually productive microseismic sequence in northeastern Italy. Geomatics, Natural Hazards and Risk, 14(1). https://doi.org/10.1080/19475705.2023.2284120