SPEKTRA: Jurnal Fisika dan Aplikasinya
Volume 11, Issue 2, August 2026
Homepage: doi.org/10.21009/SPEKTRA

DOI: doi.org/10.21009/SPEKTRA.112.04

Edge Seismic Denoising: Benchmarking the Hailo-8L on Raspberry Pi 5

Antonia Indriyani Juniar[1], Naura Anbiyya Faka Mursaid[1], Ahmad Kadarisman[1,2], Martarizal[1,*]

[1]*Department of Physics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok 16424, Indonesia*
[2]*Direktorat Instrumentasi dan Kalibrasi, BMKG, Jl. Angkasa I/2 Kemayoran, Jakarta, Indonesia*

*Corresponding Author Email: martarizal@sci.ui.ac.id

Received:
Revised:
Accepted:
Online:
Published:

SPEKTRA: Jurnal Fisika dan Aplikasinya
p-ISSN: 2541-3384
e-ISSN: 2541-3392

ABSTRACT

Real-time seismic monitoring in urban and remote environments is increasingly challenged by strong anthropogenic noise and limited computational resources at field stations. Although deep learning-based seismic denoising methods have demonstrated promising performance, their deployment on low-power edge AI platforms for continuous onsite monitoring remains insufficiently explored. This study investigates the feasibility of implementing a Conv2D-based Denoising Autoencoder (DAE) for real-time seismic signal denoising on a Raspberry Pi 5 integrated with the Hailo-8L AI accelerator. The model was trained using synthetic noise injection on seismic waveform data from the CIKJI station to generate noisy-clean signal pairs. The deployment pipeline included signal preprocessing, Conv1D-to-Conv2D adaptation, ONNX conversion, and INT8 quantization into the Hailo Execution Format (HEF). Experimental evaluation on 21,598 seismic windows demonstrated that the Hailo INT8 implementation achieved a Pearson correlation of 0.9645 and an SNR of 14.95 dB, compared to 0.9727 and 17.08 dB obtained by CPU FP32 inference, respectively. Despite a modest degradation in denoising accuracy, the Hailo-8L significantly reduced CPU utilization from 61.4% to 14.5% and maintained stable sub-millisecond inference latency during 24-hour simulated streaming tests. These results demonstrate that edge AI accelerators provide a practical tradeoff between denoising performance and computational efficiency, supporting the development of low-power real-time seismic monitoring systems for onsite deployment in resource-constrained environments. This study provides the first empirical characterization of NPU-accelerated seismic denoising under continuous real-time streaming conditions, establishing practical

benchmarks for edge AI deployment in resource-constrained seismic monitoring networks.

Keywords: denoising, autoencoder, seismic signal, Hailo-8L, MiniSEED, Raspberry Pi 5

INTRODUCTION

Real-time monitoring of seismic signals in remote areas requires an efficient edge computing system. The Raspberry Pi 5 with Hailo-8L (13 TOPS) offers a solution for implementing deep learning models with low power consumption and at a relatively affordable cost [1]. Several studies have been conducted on deep learning models ranging from autoencoders to self-supervised approaches [2-4]. The results of these studies indicate that deep learning is capable of performing denoising to separate seismic signals from noise. Recent comprehensive reviews have further confirmed that deep learning has become central to modern seismic data processing across a wide range of tasks [5].

A challenge in field signal monitoring is the lack of available noise-clean data pairs; instead, noise-clean data pairs are obtained experimentally. Therefore, a supervised approach using synthetic labels has proven valid. Other studies have also validated the synthetic noise injection approach for seismic signal denoising, where detected seismic event signals are used as proxies for clean signals [6, 7]. More recent work has extended this paradigm to self-supervised approaches that do not require paired clean references [8] and deep learning architectures specifically designed for seismic noise suppression [9], further validating the robustness of data-driven seismic denoising under real-world conditions.

On the hardware implementation side, accelerating deep learning model inference on edge devices using dedicated AI accelerators has become an increasingly relevant topic [10, 11]. However, the process of quantizing models from floating-point (FP32) to integer (INT8) for deployment on hardware accelerators such as the Hailo-8L can cause performance degradation that needs to be characterized [12]. Specific evaluations of the Hailo-8L AI Kit for seismic denoising applications have not been extensively studied in the existing literature.

Beyond the hardware-related challenges discussed above, it is also important to consider the signal-processing methods historically used for seismic denoising. Traditionally, seismic signal processing has generally relied on frequency-based filtering techniques, such as bandpass filters and Wiener filters. Although effective at reducing noise with spectral characteristics different from those of the target signal, these methods often suffer from reduced performance when the noise spectrum overlaps with the seismic signal spectrum. This can lead to distortion of the waveform, which contains information critical for seismic analysis [13].

Advances in deep learning have opened up new, more adaptive, and data-driven approaches. One of the key foundations in this field is the Denoising Autoencoder (DAE), introduced by Vincent et al. [14]. The DAE is trained to reconstruct clean data from noise-contaminated inputs, enabling it to learn latent representations that are more robust to interference. This concept has since been widely adopted in various signal processing applications, including

seismology. Deep convolutional autoencoder architectures have since been applied specifically to seismic random noise suppression [15], demonstrating the versatility and effectiveness of encoder-decoder designs for this task.

The application of deep learning to seismic data has continued to evolve through the PhaseNet model developed by Zhu and Beroza [16]. This research demonstrated that a U-Net-based convolutional encoder-decoder architecture is capable of accurately identifying the phases of P- and S-waves in seismic signals. Subsequent works such as EQTransformer [17] and QuakeFlow [18] further extended deep learning capabilities to simultaneous event detection and scalable monitoring pipelines. The success of PhaseNet demonstrates that deep learning architectures can automatically extract key characteristics from seismic signals without the need for manual feature engineering.

Automatic seismic event detection is a critical step prior to the denoising process. One of the most widely used methods is the Short-Term Average/Long-Term Average (STA/LTA) algorithm, commonly employed in automatic seismic event detection systems. This algorithm works by comparing short-term signal energy to long-term energy to detect the presence of seismic events. Seismic data processing in this study was performed using ObsPy [19], a Python toolbox that provides standard implementations of the STA/LTA detector and MiniSEED data handling routines. The computational simplicity of the STA/LTA makes it suitable for real-time monitoring systems with limited resources.

As the concept of Edge AI has evolved, various studies have begun to explore the direct application of artificial intelligence on field devices. Geng et al. [20] developed MCU-Quake, an ultra-lightweight deep learning model capable of distinguishing between earthquake signals, explosions, and ambient noise on low-power IoT devices. The results of the study show that deep learning models can be effectively implemented on edge devices with limited memory and power. Recent efforts have also demonstrated real-time AI-assisted seismic monitoring systems deployed on nodal station networks [21], and recent work has demonstrated the importance of hardware-aware computational approaches for accelerating seismic data processing workflows [22], further motivating the deployment approach taken in this study.

One key technique enabling the deployment of deep learning models on edge devices is quantization. Jacob et al. [23] demonstrated that representing weights and activations in INT8 format can maintain model accuracy close to that of FP32 while requiring significantly less memory and achieving higher inference speeds. Subsequent surveys have further characterized INT8 quantization trade-offs across diverse model families [24], and comprehensive reviews of edge AI hardware evolution confirm that INT8-capable NPUs such as the Hailo-8L represent the current standard for power-efficient edge inference [25]. This technique has become the standard for deep learning implementations on devices with limited computational resources.

In this study, the Hailo-8L neural accelerator, which delivers up to 13 TOPS of performance, was used to accelerate deep learning model inference on a Raspberry Pi 5. The Hailo-8L is specifically designed to execute INT8-based neural network operations with high power

efficiency. The integration of the Raspberry Pi 5 and the Hailo-8L provides an attractive edge AI platform for real-time seismic monitoring applications, as it reduces the computational load on the CPU while increasing the inference speed of the proposed denoising model. This study distinguishes itself from prior work by demonstrating that an NPU-based edge AI accelerator can maintain nearly identical seismic denoising quality (ΔSNR = −2.13 dB) while dramatically reducing CPU utilization during continuous 24-hour streaming, enabling practical deployment for unattended seismic field stations operating in resource-constrained environments.

Building on this motivation and the research gap identified above, this study aims to: (1) evaluate the inference performance of Hailo-8L compared to a CPU running FP32 on a Raspberry Pi 5 platform in both single-instance and batch inference scenarios; (2) analyze the degradation in denoising quality resulting from INT8 quantization on synthetic and real data; (3) test real-time streaming capabilities on a 24-hour dataset of seismic data; (4) test the model's generalization across seismic stations; and (5) characterize the effect of the number of INT8 calibration samples on output quality.

METHODS

The Raspberry Pi 5 is used as an edge computing platform with the following specifications: a 2.4 GHz quad-core ARM Cortex-A76 CPU, 8 GB of LPDDR4X RAM, and the Raspberry Pi OS based on Debian Bookworm. Idle power consumption was measured at 0.31 W, which serves as the baseline for system power consumption before inference workloads.

The Hailo-8L AI Kit is installed via an M.2 B+M Key slot with the following specifications: 13 TOPS capacity, M.2 2242 form factor, and PCIe Gen3 interface. The software stack used consists of HailoRT version 4.23.0 and the Hailo Dataflow Compiler (DFC) version 3.33.1.

Seismic data was sourced from two stations in the Indonesian BMKG network. The CIKJI station (IA.CIKJI.00.SHZ) was used for in-distribution training and evaluation, with a total of 23 MiniSEED files from January 2025 (day-of-year 002-030), yielding a total of 21,598 windows after preprocessing. The TSJM station (IA.TSJM.00.SHZ) was used exclusively for cross-station generalization evaluation (cross-domain test) and was not included in the training or calibration processes.

Preprocessing

Preprocessing includes segmenting 400-sample windows (8 seconds) at a sampling rate of 50 Hz, global normalization using a P95 value of 3,819.60 counts with clipping to the range [−1, 1], and window classification based on standard deviation: noise (std < 0.15), ambiguous (0.15 ≤ std < 0.35), and quake (std ≥ 0.35). Synthetic data pairs for training are generated by adding Gaussian noise: noisy = clean + 0.3 × gaussian_noise. The total CIKJI dataset includes 9,019 noise windows, 18,664 quake windows, and 55,992 training pairs.

Model Architecture

The AE-v4 architecture was selected over alternative deep learning models based on three practical constraints specific to Hailo-8L deployment. First, models such as DeepDenoiser [2]

and EQTransformer [17] are substantially more complex than the lightweight AE-v4 architecture and would require additional architectural adaptation and hardware-specific optimization for deployment within the present Hailo-8L pipeline. Second, U-Net-based architectures, while effective for seismic denoising, rely on skip connections and ConvTranspose layers that are incompatible with the Hailo Dataflow Compiler's INT8 graph optimization. Third, frameworks such as SeisBench provide pre-trained models but do not support the Conv1D-to-Conv2D adaptation required for Hailo deployment. The AE-v4 design replacing ConvTranspose1D with Upsample+Conv2D and using a dummy spatial dimension was therefore the minimal viable architecture that satisfies both denoising quality requirements and Hailo-8L hardware constraints.

The AE-v4 model was designed with INT8 quantization compatibility on the Hailo-8L in mind. The fundamental differences from the previous version (AE-v2) are the replacement of the ConvTranspose1D decoder with a combination of Upsample and Conv2D, as well as the migration from Conv1D to Conv2D with a dummy spatial dimension (height=1).

The encoder consists of three cascaded convolution blocks: Conv2D(1→16, kernel=(1,7), padding=(0,3)) followed by ReLU and MaxPool2D(1,2); Conv2D(16→32, kernel=(1,5), padding=(0,2)) followed by ReLU and MaxPool2D(1,2); and Conv2D(32→64, kernel=(1,3), padding=(0,1)) followed by ReLU and MaxPool2D(1,2).

The decoder consists of three cascaded upsampling blocks: Upsample(scale=(1,2), mode=nearest) followed by Conv2D(64→32, kernel=(1,3)) and ReLU; Upsample(scale=(1,2), mode=nearest) followed by Conv2D(32→16, kernel=(1,3)) and ReLU; and Upsample(scale=(1,2), mode=nearest) followed by Conv2D(16→1, kernel=(1,3)) and Tanh.

The model's input shape is (1,1,1,400): batch, channel, dummy height, seismic width. The AE-v4 model has a total of 17,185 parameters, an ONNX size of 70.1 KB, and a HEF compilation size of 868 KB. FIGURE 1 illustrates the complete encoder–decoder architecture, including layer-by-layer kernel and channel configurations.

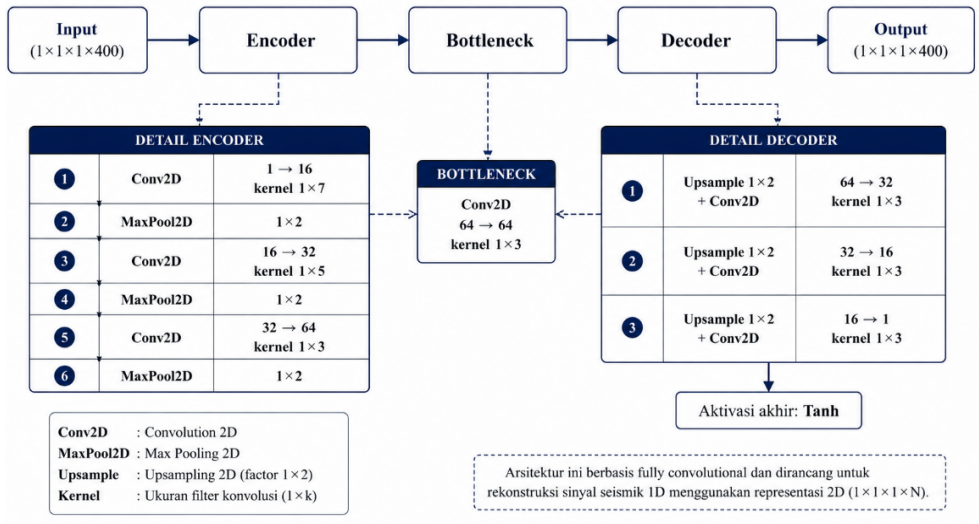

FIGURE 1. Autoencoder Architecture

Conversion to HEF

The conversion from PyTorch to HEF is performed in four stages. First, since the Hailo Dataflow Compiler (DFC) does not natively support Conv1d, all Conv1d and ConvTranspose1d layers are converted to Conv2d and ConvTranspose2d by adding a dummy spatial dimension (height=1), changing the input from (1,1,400) to (1,1,1,400). Second, the Conv2d model is exported to ONNX (opset 18) with the kernel_shape attribute explicitly added. Third, the ONNX file is parsed into HAR format using the Hailo parser and then optimized with INT8 quantization using 256 calibration samples from the original CIKJI data, targeting the Hailo-8L architecture. Fourth, the HAR file is compiled to HEF using the Hailo compiler, resulting in a 0.839 MB file ready for deployment.

Experimental Setup

The experiment was conducted on a Raspberry Pi 5 with the Hailo-8L AI Kit (13 TOPS), using HailoRT v4.23.0 and DFC v3.33.1. CPU inference used the ONNX Runtime with the CPUExecutionProvider. The evaluation covered six comprehensive aspects. First, the inference performance benchmark used 200 warmup iterations followed by 200 measurement iterations, with metrics including latency (mean, P50, P95, P99), throughput (samples per second), CPU usage, RAM, temperature, power estimation, and inference efficiency per Watt. Second, denoising quality on synthetic data using the metrics MSE, RMSE, MAE, PSNR, SNR, Pearson r, SSIM, PSD error, and band-limited SNR per octave. Third, a quality test on real data using original CIKJI data without added synthetic noise to measure quality under real-world operational conditions. Fourth, real-time streaming across 10,800 windows (equivalent to 24 hours of CIKJI data) to measure time series latency, jitter, dropped frames, and CPU usage during continuous operation. Fifth, cross-station generalization testing by evaluating the model trained on CIKJI data against TSJM data. Sixth, INT8 calibration ablation using four HEF configurations with different numbers of calibration samples: 128, 256, 512, and 1,024.

RESULTS AND DISCUSSION

As shown in TABLE 1, prior studies on seismic edge AI have primarily focused on event classification on ultra-low-power MCUs or CPU-only inference without dedicated AI accelerators. This work is the first, to the authors' knowledge, to characterize real-time seismic denoising inference on an NPU-assisted edge platform (Hailo-8L) under continuous 24-hour streaming conditions, with explicit CPU load profiling and cross-station generalization evaluation.

TABLE 1. Comparison with Related Studies on Edge AI Deployment for Seismic and Signal Processing Applications

Study	Hardware	Model	Latency	CPU Load	SNR/Acc	Focus	Key Contribution
Geng et al.	MCU (IoT, low-power)	MCU-Quake, lightweight	N/A	Low	>95%	Event classification on ultra-low power MCU	First ultra-lightweight DL model for seismic event discrimination on IoT devices
Kadarisman et al.	Raspberry Pi 5 (CPU only)	Lightweight ML	<1 ms	<20%	N/A	Anomaly detection, no AI accelerator	Edge inference on RPi5 without NPU; no streaming evaluation
Jacobe et al.	General GPU/edge devices	Various CNN	N/A	N/A	FP32 ≈ INT8	INT8 quantization Methodology, Real-time edge AI inference	Foundational INT8 quantization framework adopted across edge AI deployment
This work	RPi5 + Hailo-8L	AE-v4 Conv2D INT8	0.27 ms (Hailo)	14.5%	SNR 14.95 dB	Real-time seismic denoising with NPU acceleration	NPU-accelerated seismic denoising with 24-hour streaming validation and CPU offload profiling

CPU vs. Hailo-8L Performance Benchmark

TABLE 2 shows a comparison of inference performance between the CPU and Hailo-8L on the Raspberry Pi 5. Based on a comparison of CPU inference performance with the Hailo-8L, the CPU's absolute FP32 latency (0.17 ms) was found to be lower than that of the Hailo INT8 (0.27 ms) due to the communication overhead between the CPU and NPU via the PCIe bus. However, Hailo's CPU usage (2.7%) is significantly lower than that of the CPU's FP32 operations (7.8%), freeing up CPU resources for parallel data acquisition processes. Hailo's estimated power consumption (3.29 W) is also lower than that of the CPU FP32 (3.49 W), with the RPi5's idle power baseline at 0.31 W. It should be noted that the estimated power consumption represents the total system power, so the difference between the inference and idle states reflects the actual power consumption of the inference process. A standalone Hailo benchmark script (200 iterations, 20 warmups) under isolated conditions showed an average

TABLE 2. Comparison of CPU vs. Hailo-8L Inference Performance

Metric	CPU FP32	Hailo-8L
Latency mean (ms)	0.17	0.27
Latency P95 (ms)	0.19	0.3
Throughput(sps)	3,408.0	2,498.6
CPU Usage (%)	7.8	2.7
RAM (MB)	68.4	93.0
Temperature (°C)	49.08	49.42
Power	3.49	3.29
Inference/Watt	976.5	759.5

latency of 0.28 ms, throughput of 2,035.83 sps, CPU usage of 3.5%, and a temperature of 49.9°C, consistent with the comparative benchmark results.

Parallel load testing was conducted to evaluate the stability of inference when seismic data acquisition runs concurrently with model inference. The test results show that CPU-based inference experienced an increase in CPU usage from 21.7% to 22.5% when parallel acquisition was run. Meanwhile, inference using Hailo-8L only experienced an increase in CPU usage from 4.0% to 5.6%. Although Hailo-8L's latency increased slightly from 0.33 ms to 0.35 ms under parallel conditions, CPU usage remained significantly lower compared to pure CPU-based inference. This indicates that Hailo-8L is effective at maintaining system efficiency when inference runs concurrently with real-time data acquisition. CPU temperature across all scenarios remained relatively stable in the range of 51-53°C, with no signs of thermal throttling observed during testing. This temperature stability indicates that the Raspberry Pi 5 is still capable of continuously handling parallel inference and data acquisition workloads under the current experimental configuration.

These results demonstrate that the primary advantage of the Hailo-8L lies not in increased raw throughput, but in its ability to significantly reduce the CPU load in real-time, multi-process edge systems. This characteristic is crucial for field seismic monitoring systems that require continuous inference alongside data acquisition, buffering, network communication, and local storage.

As summarized in TABLE 3, it is worth noting that the Hailo-8L's latency consistently exceeds that of the CPU FP32 across all batch sizes (0.78x speedup at batch=1, approximately 0.61x at batch=32). This behavior is attributed to the fixed PCIe Gen3 communication overhead between the host CPU and the NPU, which imposes a baseline transfer cost regardless of computational workload. At the model scale used in this study (17,185 parameters, input shape 1×1×1×400), the per-sample compute demand is too small to amortize this bus overhead even at larger batch sizes. Consequently, the Hailo-8L does not offer a latency advantage for this lightweight model in isolated inference benchmarks. However, for edge seismic monitoring systems where inference must coexist with continuous data acquisition, buffering, and network communication, the critical metric is CPU availability rather than raw inference latency, and in this regard, the Hailo-8L's consistent reduction of CPU utilization (from 65.0% to 13.0% at batch=16, and from 100% to 14.1% at batch=32) represents a substantial operational advantage.

TABLE 3. CPU vs. Hailo-8L Performance Benchmark by Batch Size

Batch	CPU Lat (ms)	CPU (%)	CPU W	Hailo Lat (ms)	Hailo (%)	Hailo W	Speedup
1	0.21	6.4	3.41	0.27	5.8	3.41	0.78x
4	0.65	18.7	3.93	1.05	8.7	3.54	0.62x
8	1.25	33.9	4.55	2.08	12.5	3.70	0.60x
16	2.52	65.0	5.82	4.15	13.0	3.73	0.61x
32	5.04	**100.0**	7.23	8.23	**14.1**	3.78	0.61x

TABLE 4. Parallel Load Benchmark 500 Windows

Metric	CPU Baseline	CPU+Load	Hailo+Load
Latency mean (ms)	0.1582	0.1600	0.2588
Latency P95 (ms)	0.1718	0.1751	0.2722
Latency P99 (ms)	0.1800	0.1987	0.3237
Latency max (ms)	0.4192	0.3196	0.3956
CPU mean (%)	42.0	42.0	**9.0**
CPU max (%)	100.0	100.0	50.0
Temp mean (°C)	54.8	55.1	53.6
RAM (MB)	911.2	912.6	901.7

As shown in TABLE 4, when the system performs data acquisition and inference simultaneously, Hailo's average CPU usage is only 9.0%, compared to 42.0% for the CPU, a difference of 33%. CPU degradation under parallel load: latency increases by only +0.0018 ms (+1.1%), with no change in CPU usage. This demonstrates that Hailo frees up CPU capacity for logging, communication, and other parallel processing tasks.

As shown in TABLE 5, the performance degradation of Hailo INT8 compared to the FP32 CPU at -2.13 dB SNR and -0.008 Pearson r is relatively minimal. Interestingly, Hailo's Pearson r value (0.9645) is better than that of the CPU INT8 (0.9599), indicating that Hailo's quantization pipeline is more controlled than the standard ONNX Runtime INT8.

FIGURE 2 presents a visual comparison across three window categories: Noise window: CPU (SNR=-6.3 dB, r=0.429) vs. Hailo (SNR=-6.3 dB, r=0.435); Ambiguous window: CPU (SNR=-1.3 dB, r=0.593) vs. Hailo (SNR=-1.3 dB, r=0.594); Quake window: CPU (SNR=+1.6 dB, r=0.711) vs. Hailo (SNR=+1.6 dB, r=0.711).

The positive SNR in the quake window alone indicates that the model successfully preserves meaningful signals. The consistency of the CPU and Hailo metric values across all three categories demonstrates functional equivalence.

TABLE 5. Denoising Quality Metrics: CPU FP32 vs. CPU INT8 vs. Hailo INT8

Metric	CPU FP32	CPU INT8	Hailo INT8	ΔHailo vs FP32
MSE	0.0016	0.0024	0.0027	+68.8%
RMSE	0.0404	0.0489	0.0517	+27.9%
MAE	0.0262	0.0334	0.0331	+26.3%
PSNR (dB)	27.87	26.22	25.73	-2.14
SNR (dB)	17.08	15.44	14.95	-2.13
Pearson r	0.9727	0.9599	0.9645	-0.008
Latency (ms)	0.159	0.353	0.267	—

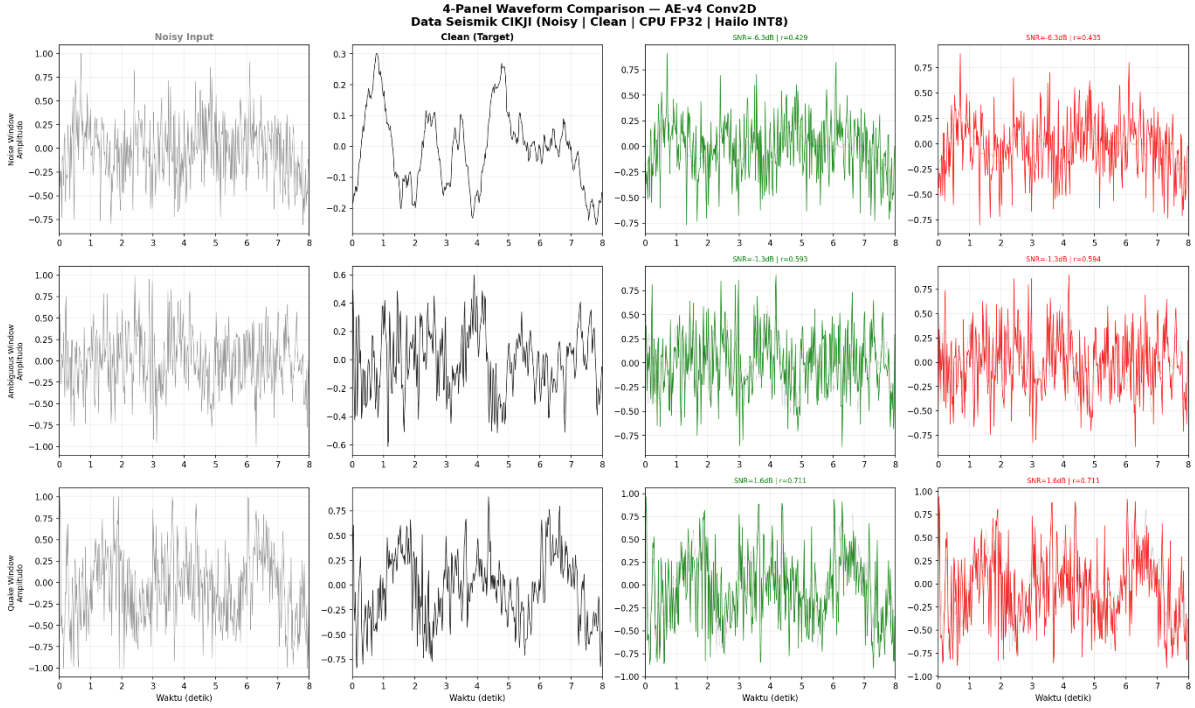

FIGURE 2. 4-panel waveform comparison (Noisy Input, Clean Target, CPU FP32, Hailo INT8)

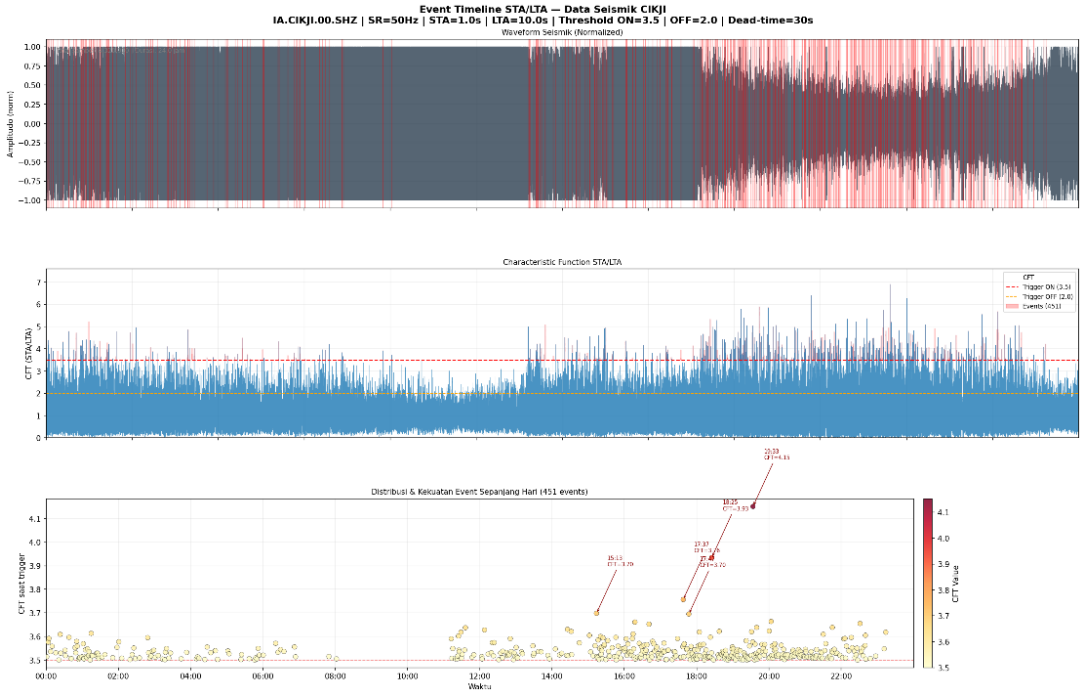

FIGURE 3. 24-hour event timeline of CIKJI seismic data (STA/LTA

As shown in FIGURE 3, the event timeline shows 451 detected events using STA/LTA parameters (threshold ON=3.5, OFF=2.0, STA=1.0 s, LTA=10.0 s, dead-time=30 s), with a maximum CFT of 9.509 and a peak in activity between 15:00 and 20:00 WIB. Note that this threshold configuration differs from the stricter ON=4.0/OFF=2.5 setting applied in the end-

TABLE 6. Full Streaming Test, 10,800 Windows

Metric	CPU FP32	Hailo INT8
Realtime ratio	3.1×10^{-5}	4.6×10^{-5}
Latency mean (ms)	0.1674	0.2698
Latency std (ms)	0.0162	0.0349
Latency P95 (ms)	0.1769	0.3006
Latency P99 (ms)	0.1889	0.3474
Latency max (ms)	0.9517	1.1058
Jitter mean (ms)	0.0047	0.0119
Jitter max (ms)	0.7866	0.8058
CPU mean (%)	**61.4**	**14.5**
CPU max (%)	100.0	66.7
Temp mean (°C)	58.5	55.6
Temp max (°C)	61.7	60.0
Events detected	**748**	**748**

to-end pipeline validation, which accounts for the difference in total detected event counts between the two analyses. A comparison of the 24-hour signals shows that the denoised CFT is sharper and better defined than the raw CFT, indicating that denoising enhances the contrast of seismic events. Residual noise appears very small compared to the denoised signal, demonstrating the effectiveness of signal-to-noise separation.

As presented in TABLE 6, during the 24-hour simulated streaming test (10,800 windows, STA/LTA threshold ON=3.5/OFF=2.0), the CPU FP32 and Hailo INT8 detected 748 identical events, with matching window indices and CFT values. This result is independent of the single-day end- to-end validation (34 vs. 33 events, threshold ON=4.0/OFF=2.5) and the exploratory event timeline analysis (451 events), as each test was conducted under distinct threshold configurations and data scopes. The Hailo CPU's mean (14.5%) was significantly lower than that of the FP32 CPU (61.4%), a difference of 46.9%, with a temperature that was 2.9°C lower.

TABLE 7. Cross-Station Generalization Test, CIKJI → TSJM

Metric	CPU→CIKJI	CPU→TSJM	Hailo→CIKJI	Hailo→TSJM
MSE	0.0746	0.0747	0.0754	0.0757
RMSE	0.2729	0.2731	0.2744	0.2750
MAE	0.2174	0.2171	0.2176	0.2171
PSNR (dB)	11.2850	11.2792	11.2386	11.2200
SNR (dB)	-6.7813	-11.0813	-6.8276	-11.1406
Pearson r	0.4291	0.2917	0.4278	0.2893

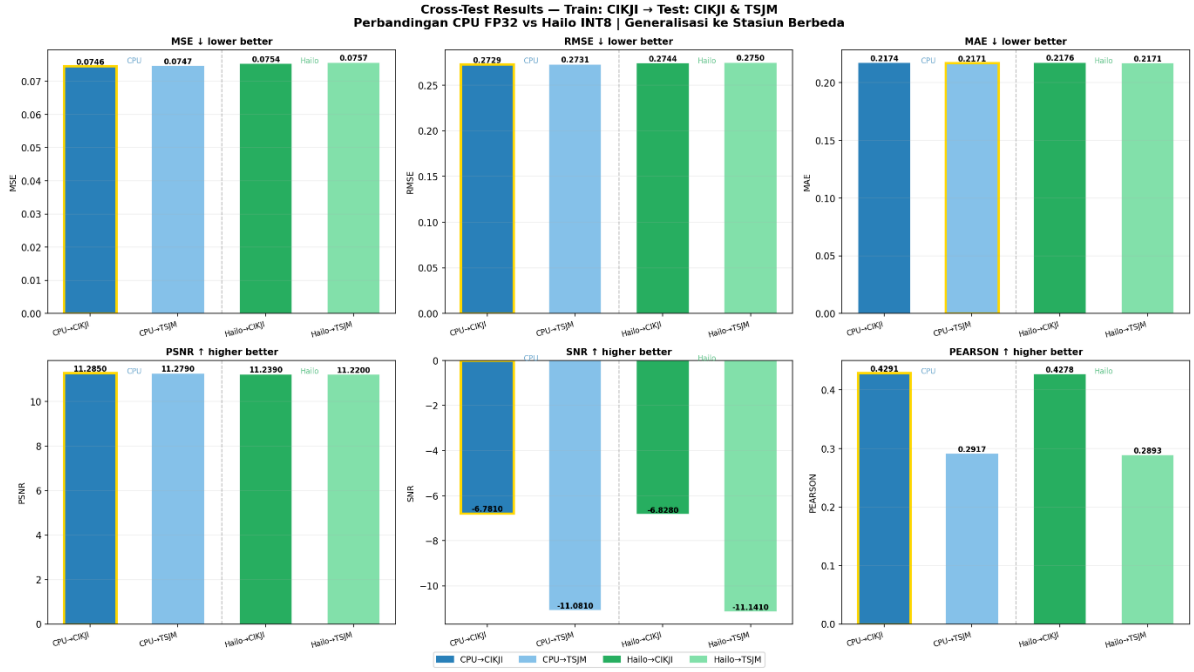

FIGURE 4. Comparison bar chart of cross-test results (Train: CIKJI → Test: CIKJI & TSJM) across six evaluation metrics

As shown in TABLE 7 and FIGURE 4, the decrease in generalization for the CPU (-4.30 dB) and Hailo (-4.31 dB) is nearly identical, confirming that the bottleneck lies in data distribution mismatch between stations rather than in the INT8 quantization process. However, the magnitude of this degradation warrants careful consideration for operational deployment. The Pearson correlation drops from 0.4291 (CIKJI) to 0.2917 (TSJM) for CPU, and from 0.4278 to 0.2893 for Hailo, a reduction of approximately 32% in waveform correlation. Combined with a 4.3 dB SNR drop, these results indicate that the model is highly localized to the CIKJI station's noise profile and instrument response characteristics. In practical terms, this means the model cannot be directly transferred to a new station without per-station retraining or domain adaptation. For multi-station network deployment across the BMKG network, each target station would require either fine-tuning on local data or an explicit domain adaptation step. This limitation is consistent with findings from other station-specific seismic deep learning studies and is addressed as a priority in the future work direction.

TABLE 8. Effect of the Number of INT8 Calibration Samples on Output Quality

N Calibration	MSE	SNR (dB)	Pearson r
128	0.069579	0.57	0.6743
256	0.073172	0.35	0.6692
512	0.077312	0.11	0.6672
1024	0.072918	0.37	0.6719
CPU FP32(Ref)	0.001633		

TABLE 8 presents the effect of the number of INT8 calibration samples on output quality. It should be noted that the SNR values in TABLE 8 differ substantially from those reported in TABLE 5, despite both configurations using N=256 calibration samples. This discrepancy arises because the calibration ablation was evaluated on a dedicated 500-window validation subset, whereas Table 5 reports results averaged across all 21,598 windows spanning noise, ambiguous, and quake categories. The absolute SNR values in Table 8 therefore reflect the characteristics of this specific subset and should not be compared directly to Table 5. The key finding from Table 8 is the relative stability across calibration sizes (ΔMSE = 0.007, ΔPearson = 0.003), confirming that quantization quality is robust to calibration sample count within the range of 128-1024.

CONCLUSION

This study contributes the first systematic, empirically grounded characterization of NPU-accelerated (Hailo-8L) seismic denoising inference under continuous real-time streaming conditions, bridging deep learning-based seismic signal processing research and practical edge AI deployment for field seismic monitoring networks.

This study yielded seven key findings that provide practical guidance for the implementation of edge AI for seismic monitoring: The Upsample+Conv2D (AE-v4) architecture successfully addressed the incompatibility of ConvTranspose with Hailo-8L's INT8 quantization, paving the way for the deployment of seismic denoising models on edge AI accelerators. Hailo-8L delivers 4.2x better CPU efficiency during 24-hour streaming (14.5% vs. 61.4%), with stable results under high load (14.1% vs. 100% on batch-32). The degradation in denoising quality due to INT8 quantization is minimal (ΔSNR = -2.13 dB, ΔPearson = -0.008), making it still suitable for seismic monitoring applications. The number of INT8 calibration samples had only a marginal effect on output quality across the tested range of 128-1,024 samples (ΔMSE = 0.007, ΔPearson = 0.003), indicating that the quantization pipeline is robust to calibration set size and does not require extensive calibration data collection for deployment. 748 STA/LTA detections were obtained identically by CPU FP32 and Hailo INT8, proving functional equivalence for practical seismic event detection applications. The Hailo-8L end-to-end pipeline improved detection sensitivity on a single-day validation file (day 002, threshold ON=4.0/OFF=2.5), with 34 events detected in the denoised signal versus 33 in the raw signal, demonstrating the added value of AI-based denoising in a stricter detection regime. Cross-station generalization consistently decreased on both the CPU and Hailo (ΔSNR ≈ -4.3 dB), indicating the need for domain adaptation for multi-station deployment. For future research, we recommend: (1) exploring domain adaptation for cross-station generalization, (2) testing multi-model pipelines to optimally utilize the Hailo-8L's 13 TOPS capacity, and (3) implementing the system on a real seismic station network with actual power measurements.

This study is limited to a single small model (17,185 parameters), synthetic noise conditions, estimated rather than directly measured power consumption, and a single accelerator platform. Future work will explore domain adaptation for cross-station generalization, concurrent multi-model deployment (PhaseNet, EQTransformer, AE-v4) within the Hailo-8L compute budget,

and field deployment on an operational BMKG station network with direct power measurement.

ACKNOWLEDGMENTS

The authors wish to thank all parties who contributed to this research, especially the Meteorology, Climatology, and Geophysics Agency (BMKG) “Bantuan Operasional Pendukung Riset” under contract 010/PKS/FMIPA/UI/2025.

REFERENCES

[1] Hailo Technologies Ltd, "Hailo-8/Hailo-8L AI Accelerator Datasheet," URL: https://hailo.ai/products/hailo-8l.
[2] W. Zhu, S. M. Mousavi, and G. C. Beroza, "Seismic Signal Denoising and Decomposition Using Deep Neural Networks," Nov. 2018, doi: 10.1109/TGRS.2019.2926772.
[3] J. Chen et al., "SelfDenoiser: Self-supervised Seismic Signal Denoiser for Continuous and Contactless Cardiac Monitoring," Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 9, no. 4, 2025, doi: 10.1145/3770701.
[4] D. Liu, Z. Deng, C. Wang, X. Wang, and W. Chen, "An Unsupervised Deep Learning Method for Denoising Prestack Random Noise," IEEE Geoscience and Remote Sensing Letters, vol. 19, 2022, doi: 10.1109/LGRS.2020.3019400.
[5] S. M. Mousavi and G. C. Beroza, "Machine Learning in Earthquake Seismology," Annual Review of Earth and Planetary Sciences, vol. 51, pp. 105-129, 2023, doi: 10.1146/annurev-earth-071822-100323.
[6] A. Steinberg, P. Gaebler, G. Hartmann, J. Lehr, and C. Pilger, "Deep Neural Networks Based Denoising of Regional Seismic Waveforms and Impact on Analysis of North Korean Nuclear Tests," Pure Appl. Geophys., vol. 182, pp. 4977-4997, 2025, doi: 10.1007/s00024-024-03491-3.
[7] S. Zhang, G. Wang, L. Lu, M. Wei, X. Qin, and L. Zhang, "TF-UNet: a time-frequency feature network for seismic noise attenuation using hybrid synthetic-field data," Journal of Supercomputing, vol. 81, no. 18, 2025, doi: 10.1007/s11227-025-08113-w.
[8] J. Li, D. Trad, and D. Liu, "Robust seismic data denoising via self-supervised deep learning," Geophysics, vol. 89, no. 5, pp. V437-V451, 2024, doi: 10.1190/geo2023-0762.1.
[9] M. Ding, Y. Zhou, and Y. Chi, "Seismic signal denoising using Swin-Conv-UNet," J. Appl. Geophy., vol. 223, 2024, doi: 10.1016/j.jappgeo.2024.105355.
[10] G. P. Pereira and M. Z. Chaari, "A Look into the Performance of the Raspberry Pi AI HAT+ for Computer Vision Applications," in Proceedings of the 2025 11th International Conference on Communication and Signal Processing, ICCSP 2025, IEEE, 2025, pp. 1620-1625, doi: 10.1109/ICCSP64183.2025.11089235.
[11] D. Minott, S. Siddiqui, and R. J. Haddad, "Benchmarking Edge AI Platforms: Performance Analysis of NVIDIA Jetson and Raspberry Pi 5 with Coral TPU," in Conference Proceedings - IEEE SOUTHEASTCON, IEEE, 2025, pp. 1384-1389, doi: 10.1109/SoutheastCon56624.2025.10971592.
[12] A. Kadarisman, I. Fachruddin, S. Soekirno, H. Andi Nugraha, and B. Heryanto Rusanto, "Early Detection of Seismic Signal Anomalies Using Raspberry Pi 5 and Lightweight Machine Learning Models," Prosiding IPS dan Seminar Nasional Fisika, vol. 14, no. 1, pp. 107-114, 2026, doi: 10.21009/03.1401.FA14.
[13] A. Burzynska, K. Abratkiewicz, and M. J. Mendecki, "U-Net-Based Raw Data-Driven Approach for Seismic Signal Processing and Spectrogram Segmentation," IEEE Sens. J., vol. 25, no. 24, pp. 44351-44362, 2025, doi: 10.1109/JSEN.2025.3625057.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in Proc. 25th Int. Conf. Mach. Learn. (ICML), Helsinki, Finland, 2008, pp. 1096-1103.
[15] N. Iqbal, "DeepSeg: Deep Segmental Denoising Neural Network for Seismic Data," IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 7, pp. 3397-3404, 2023, doi: 10.1109/TNNLS.2022.3205421.
[16] W. Zhu and G. C. Beroza, "PhaseNet: A deep-neural-network-based seismic arrival-time picking method," Geophys. J. Int., vol. 216, no. 1, pp. 261-273, Jan. 2019, doi: 10.1093/gji/ggy423.
[17] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking," Nat. Commun., vol. 11, no. 1, Dec. 2020, doi: 10.1038/s41467-020-17591-w.
[18] W. Zhu et al., "QuakeFlow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing," Geophys. J. Int., vol. 232, no. 1, pp. 684-693, Jan. 2023, doi: 10.1093/gji/ggac355.
[19] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, "ObsPy: A python toolbox for seismology," Seismological Research Letters, vol. 81, no. 3, pp. 530-533, May 2010, doi: 10.1785/gssrl.81.3.530.
[20] Z. Geng, Y. Wang, W. Pan, C. Yu, Z. Bai, and H. Zhang, "Real-time discrimination of earthquake signals by integrating artificial intelligence technology into IoT devices," Commun. Earth Environ., vol. 6, no. 1, Dec. 2025, doi: 10.1038/s43247-025-02003-y.
[21] J. Li et al., "A real-time AI-assisted seismic monitoring system based on new nodal stations with 4G telemetry and its application in the Yangbi MS 6.4 aftershock monitoring in southwest China," Earthquake Research Advances, vol. 2, no. 2, 2022, doi: 10.1016/j.eqrea.2021.100033.
[22] W. Wang et al., "Accelerating three-dimensional seismic wave simulations on ARM using a hybrid half-precision and scalable vector extension approach," Geophys. J. Int., vol. 244, no. 2, 2026, doi: 10.1093/gji/ggaf496.
[23] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 2704-2713.
[24] H. Hu, "A Review of Joint Optimization Methods for Neural Network Compression," International Journal of Computer Science and Information Technology, vol. 8, no. 3, pp. 118-124, Mar. 2026, doi: 10.62051/ijcsit.v8n3.11.
[25] T. M. Khan, Q. E. Ul Haq, S. Iqbal, and T. A. Soomro, "Edge-based artificial intelligence: Understanding the evolution of hardware and software and future trends," Eng. Appl. Artif. Intell., vol. 174, 2026, doi: 10.1016/j.engappai.2026.114526.