

Deployment-aware acoustic anomaly detection on an embedded microcontroller

Truong Minh Duc, Nguyen Ngoc Minh, Truong Cao Dung

Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology (PTIT), Ha Noi City, Vietnam

Article Info

Article history:

Received Jun 12, 2026

Revised Aug 31, 2026

Accepted Sep 11, 2026

Keywords:

Acoustic anomaly detection
Embedded consistency evaluation
Internet of things edge monitoring
Reconfigurable systems
Tiny machine learning

ABSTRACT

Industrial acoustic anomaly detection is often optimized in off-device benchmarks, while the consistency of exported thresholded decisions on embedded microcontrollers remains less studied. This paper presents a bounded embedded consistency-evaluation case study for an internet of things (IoT)-class acoustic monitoring pipeline on the public MIMII fan 0 dB subset. The pipeline uses a $40 \times 63 \times 1$ log-mel representation, compact convolutional autoencoder (CAE), INT8 TFLite deployment, and latent-space Mahalanobis scoring. The raw Mahalanobis rule defines the on-device decision; a 2-out-of-5 temporal rule is retained only for offline ordered-subset analysis. For fan/id_06/0 dB, repeated stratified evaluation yielded average area under the curve (AUC) 0.96 and F1 0.82. Supporting all-ID evaluation over id_00, id_02, id_04, and id_06 showed the raw Mahalanobis path had the strongest aggregate metrics among tested scoring paths. Stored-sample replay on ESP32-S3 used a fixed 20-sample id_06 batch with generated C audio headers through the canonical DSP path. Both reference-input and digital signal processing (DSP)-path checks produced 20/20 raw-decision agreement with the off-device reference. The results support bounded reference-device consistency for the evaluated batch and provide a reference-device basis for future embedded, IoT-edge, and block-level reconfigurable-system studies, but not cross-ID robustness, streaming, latency, or energy behavior.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Truong Minh Duc

Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology (PTIT)

Ha Noi City, Vietnam

Email: ductm@ptit.edu.vn

1. INTRODUCTION

Industrial machine health monitoring is essential for avoiding costly downtime, and predictive maintenance increasingly relies on acoustic analysis to detect abnormal operating states without interrupting production. In smart factories and connected machinery, such detectors are often expected to move from an off-device development environment to embedded or internet of things (IoT) edge nodes close to the monitored equipment. In this context, unsupervised anomalous sound detection (ASD) has attracted substantial attention because it learns normal sound patterns and flags deviations without requiring labeled fault data, which are often scarce and unpredictable in practice. Progress in industrial ASD was substantially shaped by the release of the MIMII dataset [1], which provides machine recordings under both normal and faulty conditions in realistic environments, and by the subsequent benchmark formalization in DCASE 2020 Task 2 [2], where systems are trained only on normal data and must detect previously unseen anomalies at test time.

The availability of public benchmarks has enabled a wide range of ASD approaches. Recent studies

have reported gains through stronger temporal modeling [3], self-supervised representation learning [4], and compact one-class objectives [5], [6]. Embedded inference research has likewise shown that quantized neural networks and compact Tiny machine learning (TinyML) backbones can execute within tight microcontroller budgets [7], [8]. However, much of the existing literature still focuses on benchmark accuracy, representation learning, or hardware feasibility rather than on whether exported scoring artifacts preserve thresholded raw decisions after quantized transfer to microcontroller unit (MCU)-class devices. Off-device evaluation alone cannot verify whether the exported digital signal processing (DSP) path, quantized inference runtime, and scoring artifacts preserve the same alarm/no-alarm decisions under MCU memory and implementation constraints. This transfer check is important for industrial anomaly detectors because small score perturbations near the operating threshold can change maintenance alarms, operator triage, or downstream logging behavior even when aggregate benchmark metrics remain strong.

Against this background, the present paper studies a narrower embedded consistency-evaluation question. Within a fixed DCASE-style MIMII fan 0 dB setting, it asks whether an off-device trained anomaly detector can be evaluated on ESP32-S3, used here as a concrete MCU-class TinyML target, through stored-sample replay of a fixed test batch while preserving raw-decision agreement when both the INT8 encoder and the exported latent-space Mahalanobis artifacts are frozen. All model fitting and threshold calibration follow the one-class benchmark protocol: only normal MIMII fan recordings are used before evaluation, while abnormal clips are held out for testing. Among the evaluated fan IDs, id_06 was carried forward as the deployment anchor because it combined the strongest repeated-split raw metrics, the largest mean Mahalanobis margin, and the most stable anchor behavior in the bounded per-ID comparison. The weaker ID results are kept as supporting limitations rather than omitted from the evidence chain. The device-side check then replays a frozen id_06 batch derived from the public MIMII benchmark WAV files rather than introducing a new live-data collection.

The claim boundary is limited to raw-decision agreement between the off-device reference and device execution under that protocol. Score and latent discrepancies are treated as bounded diagnostics, the retained 2-out-of-5 rule as an offline ordered-subset policy rather than as a device-validated endpoint, and the device study as fixed-batch reproducibility plus static memory-footprint reporting rather than live microphone acquisition, continuous streaming, timing, or energy characterization.

The paper makes three contributions. First, it frames a bounded replay question as an embedded and IoT-edge consistency-evaluation case study for MIMII fan 0 dB on ESP32-S3, with a block-decomposed DSP, INT8 encoder, Mahalanobis scoring, and thresholding path relevant to later reconfigurable-system studies. Second, it freezes an INT8 encoder together with exported latent-space Mahalanobis artifacts (μ , Σ^{-1} , τ) for a fixed-batch reference-device comparison centered on thresholded raw-decision agreement. Third, it keeps the evidence hierarchy explicit by separating the primary id_06 anchor study, supporting all-ID descriptive context, and the direct fixed-batch device-side consistency evidence. This contribution differs from prior TinyML ASD studies by testing the transferred decision path itself, not by proposing a new ASD backbone or a complete real-time embedded monitoring system.

2. RELATED WORK

Prior work relevant to this study can be grouped by proximity to the present pipeline. The closest group is one-class, reconstruction-based, and lightweight decision-oriented ASD, although several methods differ from the present pipeline in supervision, metadata usage, or score construction. Deep SVDD has been applied to industrial machine audio as a compact one-class baseline [6], [9] while convolutional variational autoencoders and compact self-encoder pipelines remain natural reconstruction baselines [10], [11]. Activity-aware detection and attentive neural-process scoring explore alternative decision structures beyond plain reconstruction [12], [13], hyperbolic latent modeling offers a different geometric view of anomaly separation [14], and metadata-assisted first-shot ASD and prototype-based learning extend this design space when anomaly estimation or partial attribute information is available [5], [15]. Broader off-device ASD studies provide a second comparison group: temporal modeling with recurrent backbones and spectral-temporal fusion remains competitive for industrial audio [3], [16], while recent systems increasingly rely on self-supervised or contrastive representation learning [4], [17]. Joint generative-contrastive learning, hierarchical metadata constraints, and noise-aware contrastive objectives further refine latent representations under more demanding benchmark settings [18]–[20].

A third comparison group concerns embedded feasibility and MCU deployment. Quantized neural-network libraries and embedded inference runtimes establish the software foundation for MCU-side execution [7],

[21], and fully quantized TinyML backbones and related lightweight edge reports show that useful models can fit within strict memory and compute budgets on MCU-class hardware [8], [22]. These embedded-inference studies motivate the paper's narrow question: whether a frozen convolutional autoencoder (CAE)-plus-Mahalanobis detector preserves raw decisions in fixed-batch ESP32-S3 replay. The present study therefore complements prior TinyML and embedded-inference work by auditing the transferred decision path and exported scoring artifacts, while treating hardware-level realization as future work rather than as evidence provided here.

3. RESEARCH METHOD

3.1. Feature representation and convolutional autoencoder architecture

This subsection defines the exportable feature-to-latent path of the proposed pipeline. Each 2 s audio segment is sampled at 16 kHz and converted into a $40 \times 63 \times 1$ log-mel tensor [23] using `n_fft = 1024`, `hop_length = 512`, and `n_mels = 40`. The reproducible feature path uses MIMII channel 0, zero-pads or truncates each clip to the fixed 2 s duration, applies librosa mel-spectrogram defaults for parameters not listed above, and converts power to decibels with `ref = np.max` before the fixed TFLite affine input quantization. Let $x_t \in \mathbb{R}^{40 \times 63 \times 1}$ denote the resulting feature tensor at sequence index t .

A compact CAE maps this tensor to a low-dimensional latent representation used by the ESP32-S3 scoring path studied here. The encoder applies a 3×3 convolution with 8 channels and ReLU activation, a 2×2 max-pooling layer, a second 3×3 convolution with 16 channels and rectified linear unit (ReLU) activation, and a second 2×2 max-pooling layer. A third 3×3 convolution with 16 channels forms a latent tensor, which is reduced by global average pooling to the 16-D latent vector $z_t = f_\theta(x_t) \in \mathbb{R}^{16}$. The resulting CAE has 7,121 trainable parameters. This compact CAE design follows the general direction of lightweight latent-encoder ASD systems that learn compact latent representations useful for downstream anomaly scoring under limited model capacity [10], [11].

During training, the latent tensor is passed through a symmetric decoder composed of two transposed-convolution blocks with upsampling, followed by a cropping layer and a linear single-channel reconstruction head. The autoencoder is optimized off-device with Adam and mean squared reconstruction loss so that the encoder learns a reconstruction-oriented latent space. For embedded deployment, only the encoder branch is quantized and exported as an INT8 TFLite model, while the decoder remains a training-time component.

The device-side consistency study uses a two-stage ESP32-S3 protocol. Stage A drives the MCU with off-device reference tensors to isolate the exported inference and scoring artifacts. Stage B replays the same fixed batch through the canonical MCU DSP path. Figure 1 summarizes the validated path from DSP/log-mel features and INT8 encoder inference to latent Mahalanobis scoring and thresholded raw decisions. Device-side validation stops at raw thresholding; temporal voting and the terminal decision are retained only as offline ordered-subset policy blocks. This block-level separation keeps front-end DSP, quantized encoder inference, compact scoring artifacts, and threshold logic inspectable as implementation units. Thus, the study evaluates fixed-batch raw-decision consistency, not streaming behavior, latency, or energy.

3.2. Latent-space scoring and temporal decisions

This subsection defines the score artifacts and decision rules that operate on the 16-D latent vector. Three score families are considered in the study: reconstruction mean squared error (MSE), latent L2-to-mean, and Mahalanobis scoring in the latent space. Only Mahalanobis scoring followed by thresholding defines the deployed raw detector; reconstruction MSE and latent L2-to-mean are off-device comparison scores used in the ablation analysis. The focus on a compact latent-space detector is complementary to recent self-supervised and contrastive ASD strategies [4], [18]. Prototype-based adaptation has also shown promise when additional attribute information is available [5].

Mahalanobis is retained as the deployed raw score because it has a compact exported-artifact form through μ , Σ^{-1} , and τ . For each export, the INT8 encoder output is dequantized back to $z_t \in \mathbb{R}^{16}$; μ is the train-normal latent mean in that space, Σ^{-1} is obtained from a regularized latent covariance using a pseudoinverse when needed, and τ is set by a percentile rule on train-normal Mahalanobis scores. When a single frozen export bundle is used for raw reference-versus-device agreement checks, these artifacts are held fixed, whereas repeated-split analyses regenerate their own score artifacts. The deployed latent-space score is written as (1):

$$s_t = (z_t - \mu)^\top \Sigma^{-1} (z_t - \mu), \quad (1)$$

where μ and Σ^{-1} denote the exported train-normal latent mean and regularized inverse covariance in the dequantized latent space. Since s_t is used as a ranking and thresholding quantity rather than as a normalized

likelihood, we refer to it throughout the paper as a Mahalanobis score. A raw anomaly decision is then defined as (2):

$$\hat{y}_t^{\text{raw}} = \begin{cases} 1, & s_t > \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where τ denotes the operating threshold. Higher values of s_t indicate larger deviation from the normal latent distribution. Temporal voting is evaluated as a second design layer on top of the raw detector in offline ordered-subset analysis; it is not the endpoint used for the device raw-decision consistency check. Here, t indexes successive 2 s segments only after filename restoration within an ordered evaluation subset, so temporal voting is characterized in that setting rather than as a generic streaming benchmark. For a temporal rule with required vote count k and window length w , the aggregated decision at sequence index t is written as (3):

$$\hat{y}_t^{(k,w)} = \begin{cases} 1, & \sum_{i=\max(1, t-w+1)}^t \hat{y}_i^{\text{raw}} \geq k, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

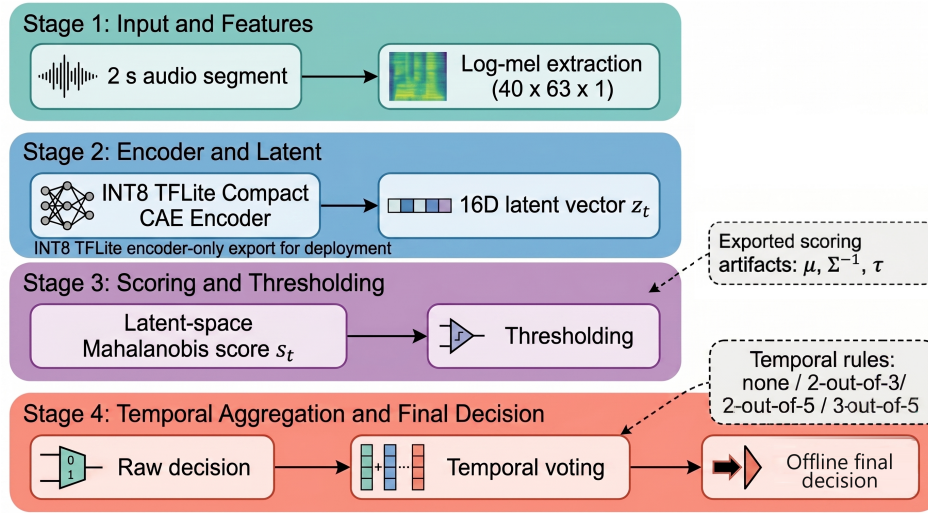


Figure 1. Proposed TinyML pipeline for acoustic anomaly detection: feature extraction, INT8 encoder inference, Mahalanobis scoring, thresholding, and temporal decision blocks

The paper retains the 2-out-of-3 and 2-out-of-5 temporal rules for offline ordered-subset analysis. A stricter 3-out-of-5 variant was checked only as a sensitivity case and was not used to reselect the detector, threshold, deployment anchor, or device-side claim. The truncated-prefix definition makes the first $w - 1$ positions explicit without zero padding. Among the retained rules, 2-out-of-5 is treated as a pre-fixed offline smoothing policy, not as evidence beyond the raw Mahalanobis detector.

The experiments are organized around three evidence layers. The repeated fan/id_06/0 dB study provides the primary single-ID anchor evidence. The repeated all-ID study provides supporting descriptive evidence about cross-ID variation and remaining limitations. The fixed 20-sample test batch provides direct bounded reference-device consistency evidence for the deployed ESP32-S3 path. A separate score-family ablation provides single-run design-selection evidence and is not treated as a fourth claim-bearing layer.

3.3. Dataset, training, and export setup

The study follows the one-class MIMII/DCASE setting on the fan 0 dB subset, with id_00, id_02, id_04, and id_06 used in the repeated all-ID evaluation and id_06 used as the primary single-ID anchor for deployment-consistency analysis [1], [2]. Within this scope, the study evaluates 4075 normal and 1475 abnormal clips across the four fan IDs.

Following the benchmark protocol, the CAE is trained from normal recordings only. In this one-class construction, abnormal clips are evaluation-only and validation is formed solely from train-normal data. Model

fitting uses Adam [24], mean squared reconstruction loss, a maximum of 50 epochs, batch size 32, and early stopping on validation loss with patience 8 and best-weight restoration. Within each claim-bearing split, 20% of the train-normal partition is carved out as a shuffled validation subset for early stopping only. After training, post-training quantization uses representative training data to export an INT8 TFLite encoder whose input and output types are both INT8 [25]. The exported INT8 encoder, together with the fixed Mahalanobis artifacts μ , Σ^{-1} , and τ , is synchronized with the MCU firmware for the raw-decision consistency check.

3.4. Repeated evaluation protocol

The main reported metrics are area under the curve (AUC), pAUC@0.1, and F1, with precision and recall used to compute the thresholded F1 summaries. Here pAUC@0.1 denotes the standardized partial receiver operating characteristic AUC (ROC AUC) over the false-positive-rate range $[0, 0.1]$. The mean margin reported for anchor selection is the held-out class-separation margin, computed as the mean abnormal Mahalanobis score minus the mean normal Mahalanobis score and then averaged across repeated splits. The primary anchor study evaluates fan/id_06/0 dB with repeated stratified shuffle split [26], n_splits=5, and test_size=0.25. Within the evaluated fan IDs, id_06 serves as the deployment anchor because the repeated offline study identifies it as the strongest raw single-ID case with the largest mean Mahalanobis margin and the most stable anchor behavior.

For each split, the encoder is trained on train-normal data and then held fixed for evaluation. The Mahalanobis path is calibrated in the dequantized latent space of the exported INT8 TFLite encoder: μ is the train-normal latent mean, Σ^{-1} is the pseudoinverse of the train-normal covariance after $1 \times 10^{-4} I$ regularization, and τ is the 95th percentile of the corresponding train-normal Mahalanobis scores. This label-free τ calibration uses train-normal data only; no abnormal clips, evaluation labels, or fixed-batch samples enter threshold setting.

The sensitivity check in this study is deliberately limited to operating parameters that affect the exported decision path or the offline temporal policy. The train-normal percentile threshold, the $1 \times 10^{-4} I$ covariance regularization, and the retained 2-out-of-3 and 2-out-of-5 temporal rules were kept explicit across the repeated evaluations. A stricter 3-out-of-5 temporal variant was additionally checked offline as a sensitivity variant but was not promoted to the main operating-policy summary because the paper's deployment claim is tied to raw Mahalanobis thresholding rather than to temporal voting. This compact sensitivity scope preserves reproducibility without turning the study into a broad hyperparameter-search benchmark.

The retained Mahalanobis + 2-out-of-5 offline operating policy was selected from repeated held-out comparisons and frozen before any Stage A/B device tests. After each held-out subset is restored to deterministic filename order, temporal voting is applied as ordered-subset smoothing rather than as contiguous streaming evidence.

The same repeated protocol is then applied independently to id_00, id_02, id_04, and id_06 in the all-ID study, yielding 20 evaluated split rows in the aggregate summary. Repeated summaries are reported as empirical mean and standard deviation, with $R=5$ for the id_06 anchor study and $R=20$ for the pooled all-ID study.

3.5. Embedded consistency-evaluation setup

The batch reference-device study directly evaluates bounded consistency for the deployed detector on a fixed 20-sample id_06 test batch containing 10 normal and 10 abnormal clips. To avoid post hoc drift during device testing, this balanced id_06 batch was frozen before Stage A/B execution and then reused unchanged in both stages; no batch items were used in τ calibration or in the earlier offline choice of the retained 2-out-of-5 operating policy.

Stage A and Stage B both use one frozen id_06 deployment bundle consisting of the quantized encoder, synchronized μ , Σ^{-1} , and τ , and the same fixed 20-sample test batch. For reproducibility, the retained bundle contains the INT8 TFLite encoder, TFLite input/output scale and zero-point parameters, the Mahalanobis artifacts, the operating threshold, the fixed-batch filename list, generated C audio/header files, and Stage A/B comparison logs. In Stage A, the device input tensor is forced from the off-device reference so that the controlled input_q $\rightarrow$ latent $\rightarrow$ score $\rightarrow$ raw decision path can be tested directly. In Stage B, the same fixed batch is re-evaluated through the full canonical DSP path audio $\rightarrow$ DSP $\rightarrow$ input_q $\rightarrow$ latent $\rightarrow$ score $\rightarrow$ raw decision using provenance-clean embedded C audio headers generated from the public MIMII benchmark WAV files. This design separates an artifact-transfer consistency check from fixed-batch DSP-path retesting on the device. Stage B is a stored-sample batch replay; it does not include live microphone acquisition, I2S buffering, continuous streaming, latency measurement, or energy measurement. The device evidence in this paper therefore covers raw-decision reproduction under Stage A/B for that fixed bundle, not full on-device reproduction of the temporal policy.

The embedded deployment target is ESP32-S3 (esp32-s3-devkitc-1) through PlatformIO and Arduino. These embedded experiments evaluate fixed-batch reproducibility and static memory-footprint reporting rather than deployment robustness or production performance. Accordingly, this paper reports static build footprint and fixed-batch reference-device reproducibility, while leaving rigorous latency and energy characterization to future work.

4. RESULTS AND DISCUSSION

4.1. Strongest single-ID anchor

The primary deployment anchor is the fan/id_06/0 dB case. Table 1 reports the per-ID repeated-split screening summary used to choose this anchor from the evaluated fan IDs.

Table 1. Per-ID repeated-split raw Mahalanobis summary for deployment-anchor selection

ID	AUC	pAUC@0.1	F1	Mean margin	Selection
id_00	0.626±0.051	0.544±0.025	0.243±0.067	0.726	Support
id_02	0.721±0.028	0.600±0.026	0.394±0.062	1.721	Support
id_04	0.639±0.036	0.530±0.008	0.190±0.034	0.448	Support
id_06	0.965±0.017	0.865±0.054	0.844±0.051	3.419	Anchor

Within this bounded fan-only pool, id_06 combines the strongest raw metrics with the largest mean Mahalanobis margin. The dedicated id_06 stability run used for Figure 2 is reported separately because it characterizes the selected anchor in more detail: the raw Mahalanobis detector achieved 0.958 ± 0.015 AUC, 0.858 ± 0.034 pAUC@0.1, and 0.815 ± 0.045 F1, with a split-wise threshold of 5.582 ± 0.154 . Together, the per-ID screening summary and the dedicated anchor run provide the main offline evidence that id_06 is a suitable anchor for the subsequent fixed-batch device study. The weaker id_00 and id_04 rows remain visible in the all-ID evidence as limitations, so the selected anchor is positioned as the strongest bounded case rather than as proof of cross-ID robustness.

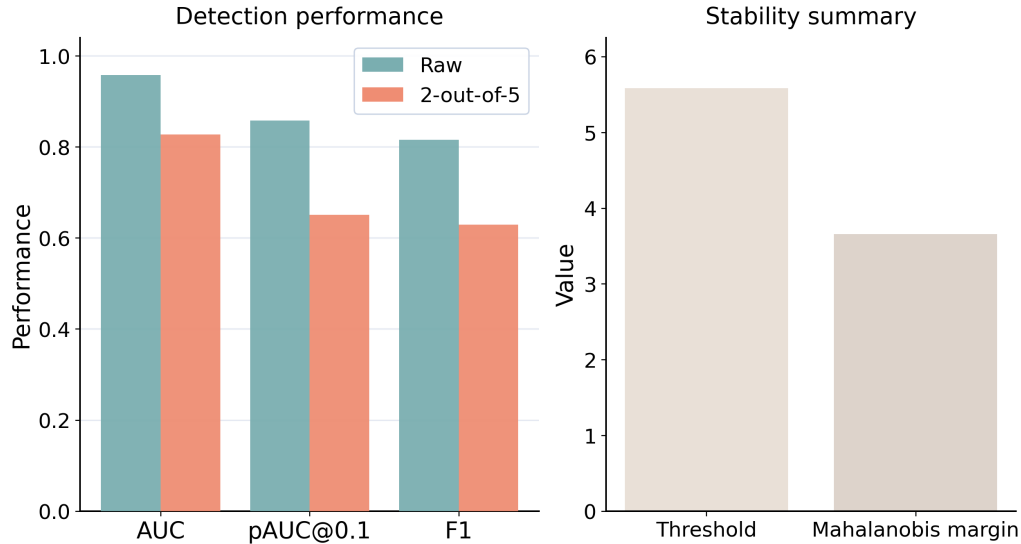


Figure 2. Repeated-split stability summary for the selected fan/id_06/0 dB anchor

The repeated all-ID study provides descriptive evidence about aggregate cross-ID variation and remaining limitations beyond the primary anchor case. For compactness, the repeated all-ID block of Table 2 reports the raw Mahalanobis path together with the two temporal rules carried forward into the operating-policy comparison, namely 2-out-of-3 and 2-out-of-5.

Table 2. Core offline results supporting the proposed detector across the repeated all-ID study and the single-run score-family ablation

Method	AUC	pAUC@0.1	F1
<i>Repeated all-ID aggregate (fan 0 dB)</i>			
Raw Mahalanobis	0.738±0.140	0.635±0.139	0.418±0.263
2-out-of-3	0.660±0.124	0.573±0.062	0.261±0.238
2-out-of-5	0.673±0.119	0.568±0.049	0.337±0.209
<i>Score-family ablation (aggregate single run)</i>			
Mahalanobis	0.733	0.624	0.427
Latent L2	0.641	0.541	0.213
Reconstruction MSE	0.656	0.589	0.368

Note. The score-family ablation block is a single-run design-selection check, not repeated-split benchmark evidence.

Under the repeated-split protocol, the raw Mahalanobis path produced the highest aggregate metrics among these reported detector variants. The 2-out-of-5 rule is retained as a pre-fixed offline ordered-subset policy fixed before device testing, not as a claim that temporal voting improves on the raw path. Because Table 2 is aggregate, the repeated all-ID study is interpreted as descriptive evidence of cross-ID variability and present limits rather than as evidence of robust deployment across IDs.

4.2. Anchor temporal stability

The pre-fixed 2-out-of-5 temporal operating policy is examined alongside the deployment study because it provides ordered-subset smoothing and a reproducible offline rule, even though it is not the numerically strongest detector in the offline tables. Under the same ordered-sequence protocol, the 2-out-of-5 rule yielded 0.828 ± 0.032 AUC, 0.651 ± 0.016 pAUC@0.1, and 0.629 ± 0.043 F1. These values show that temporal behavior remains sensitive to the evaluation protocol and operating objective even on the primary single-ID anchor. Figure 2 summarizes repeated-split behavior for this anchor.

4.3. Score-family ablation

The score-family block of Table 2 supports selecting Mahalanobis within this pipeline because it gives the highest aggregate single-run metrics among the tested score families. Because this ablation is reported in a different evaluation context from the repeated-split results, it is used here as design-selection evidence rather than as a broader benchmark claim.

4.4. Fixed-batch reference-device consistency

The fixed-batch study directly compares the off-device reference and ESP32-S3 behavior on a fixed id_06 test batch. Raw agreement is reported here as the fraction of batch items for which the reference and device raw decisions match; it is an implementation-consistency measure, not ground-truth accuracy. Score consistency and latent consistency are summarized by batch-averaged absolute score and latent errors. Table 3 consolidates the fixed 20-sample id_06 batch study with the static build footprint.

Table 3. Fixed-batch ESP32-S3 consistency and static build footprint

Stage	Raw agree.	Mean score err.	Mean latent err.
<i>Fixed 20-sample id_06 consistency batch</i>			
Stage A ref. input	20/20	0.030	0.001
Stage B DSP replay	20/20	0.127	0.005
<i>Static build footprint</i>			
ESP32-S3 static build	28.7% RAM/17.2% Flash		

In Stage A, the ESP32-S3 was driven by the off-device reference tensor and achieved 20/20 raw-decision agreement with mean score and latent errors of 0.030 and 0.001. In Stage B, the same fixed batch passed through the full DSP path and again achieved 20/20 agreement, with mean score and latent errors of 0.127 and 0.005. Relative to the anchor threshold scale reported above (5.582 ± 0.154), the Stage B score shift remains small on the same score axis, without implying unchanged ranking or calibration. The ESP32-S3 static build occupied 28.7% RAM and 17.2% Flash (94,136/327,680 RAM bytes; 575,317/3,342,336 Flash bytes). The reported memory figures are static build-footprint values; runtime heap/stack high-water marks, latency, and energy were not measured. These results therefore support only fixed-batch id_06 raw-decision agreement

with bounded mean score and latent drift across the evaluated batch. Figure 3 separates the Stage A/Stage B summary from the ordered per-sample Stage B margins.

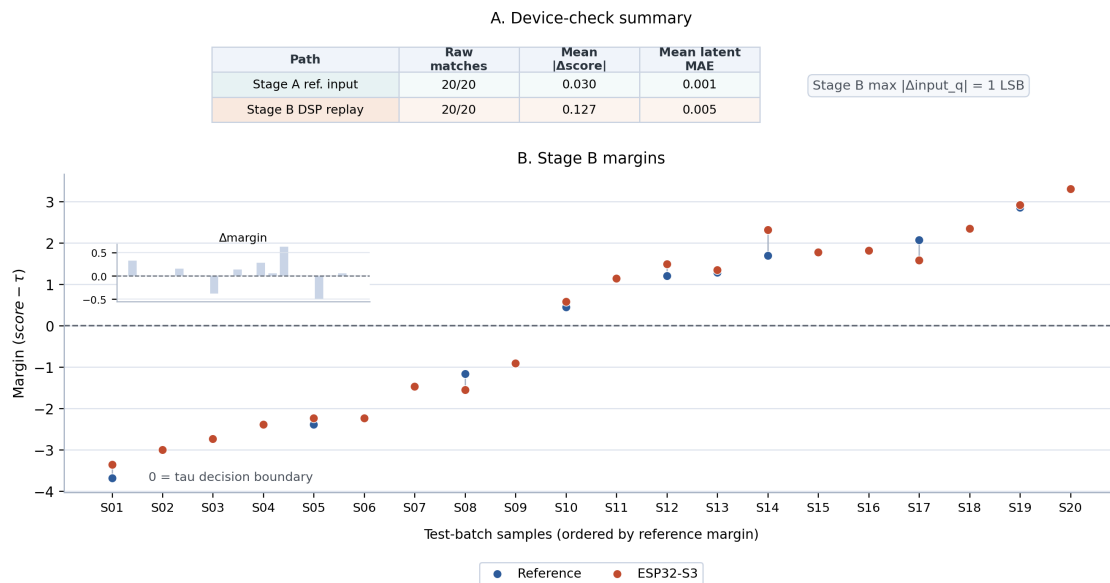


Figure 3. Reference-device consistency summary for the fixed id₀₆ batch on ESP32-S3

4.5. Discussion

The Stage A/Stage B contrast defines the safe deployment interpretation in this paper. Stage A shows near-numerical parity along the input_q → latent → score → raw decision path because the device is driven from the off-device reference tensor. Stage B shows that reintroducing the canonical DSP front end changes intermediate scores and latents, yet the fixed id₀₆ batch still preserves 20/20 raw decisions. Across the tested detector variants in Table 2, the raw Mahalanobis path gives the strongest aggregate metrics, whereas the pre-fixed 2-out-of-5 rule is retained only as an offline ordered-subset policy. The score-family ablation supports design selection within this pipeline, not a broader benchmark result, and the deployment result remains limited to fixed-batch id₀₆ raw-decision agreement with bounded latent and score drift rather than cross-ID robustness, score-ranking invariance, calibration invariance, or field-performance guarantees.

4.6. Embedded, reconfigurable, and internet of things implications

For industrial IoT monitoring, raw-decision consistency is practically important because an edge node typically forwards an alarm state, event log, or maintenance trigger rather than a full benchmark curve. A detector that preserves aggregate AUC but changes thresholded decisions near τ may still produce different alarm/no-alarm behavior after deployment. The Stage A/B protocol therefore acts as a fixed-batch deterministic raw-decision consistency check before broader field trials: it verifies whether the exported INT8 encoder, synchronized Mahalanobis artifacts, and threshold logic preserve the fixed reference decisions on the target embedded device.

The same decomposition is also relevant to future hardware-oriented and networked deployment studies. The pipeline separates feature extraction, quantized encoder inference, compact Mahalanobis scoring, and a comparator-style threshold decision. This block-level structure may help frame future FPGA, ASIC, or multi-node IoT studies, but the present paper does not evaluate hardware synthesis, distributed deployment, latency, energy, or security behavior; instead, it reports bounded reference-device consistency evidence for the evaluated ESP32-S3 batch.

4.7. Limitations and threats to validity

The evidence should be read as a bounded case study. The fixed-batch device result uses one id₀₆ test batch and stored-sample replay through generated headers rather than live acquisition. It does not establish live streaming behavior, runtime performance, secure or distributed deployment, cross-ID robustness, calibration or

ranking invariance, or on-device reproduction of the temporal policy. The all-ID rows expose weaker id_00 and id_04 behavior; those rows are descriptive warnings rather than deployment evidence.

5. CONCLUSION

This study evaluated a bounded embedded consistency-evaluation question for a TinyML ASD pipeline on MIMII fan 0 dB using an ESP32-S3. The main finding is restricted to the fixed 20-sample fan/id_06/0 dB test batch. For this batch, the exported INT8 encoder and synchronized latent-space Mahalanobis artifacts achieved complete agreement in raw decisions between the off-device reference and the ESP32-S3. Stage A exercised the controlled artifact-transfer path by feeding off-device reference tensors through the device inference/scoring path, while Stage B retested generated C audio headers as fixed stored-sample replay through the canonical DSP path. Both stages showed 20/20 raw-decision agreement; bounded score and latent drift are reported only as mean diagnostics, not as evidence of calibration or ranking invariance. Within this evidence boundary, we conclude reference-device raw-decision consistency for the evaluated fixed fan/id_06/0 dB batch.

The result is relevant to embedded and IoT-edge deployment studies because it checks whether fixed binary alarm decisions survive transfer to MCU-class execution and exposes a block-level decision path for future reconfigurable-hardware investigations. These results do not establish live microphone acquisition, continuous streaming, embedded reproduction of the offline ordered-subset temporal-voting policy, broader calibration invariance, score-ranking invariance, cross-ID robustness, or complete field characterization. Future work should extend the same protocol to additional test batches, machine IDs, acoustic conditions, continuous streams, networked deployments, security-aware integration, and reproducible runtime characterization.

ACKNOWLEDGMENTS

The authors acknowledge institutional support from AVIS Lab, Posts and Telecommunications Institute of Technology (PTIT), Vietnam.

FUNDING INFORMATION

This research received no specific external grant; institutional support was provided by AVIS Lab, PTIT, Vietnam.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Truong Minh Duc			✓			✓		✓	✓					
Nguyen Ngoc Minh		✓		✓						✓				
Truong Cao Dung		✓								✓		✓		

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal Analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project Administration

Fu : Funding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest related to this work.

DATA AVAILABILITY

The MIMII dataset used in this study is publicly available from Zenodo. Its DOI is 10.5281/zenodo.3384388. The DCASE one-class benchmark protocol is described in the cited DCASE sources. The original

MIMII audio files are not redistributed by the authors. Author-generated materials that support checking the reported fixed-batch reference-device result, including the id.06 batch filename list, threshold and scoring-configuration notes, summary tables, and Stage A/B comparison logs, are available from the corresponding author upon reasonable request.

REFERENCES

- [1] H. Purohit *et al.*, “MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019, pp. 209–213, doi: 10.33682/m76f-d618.
- [2] Y. Koizumi *et al.*, “Description and discussion on DCASE 2020 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring,” in *Proceedings of the 5th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020)*, 2020, pp. 81–85.
- [3] T.-T.-H. Le, A. A. Adiputra, J. Yun, and H. Kim, “Anomaly detection in industrial machine sounds using high-frequency features and gate recurrent unit networks,” *IEEE Access*, vol. 13, pp. 77165–77186, 2025, doi: 10.1109/ACCESS.2025.3565812.
- [4] K. Wilkinghoff, “Self-supervised learning for anomalous sound detection,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 276–280, doi: 10.1109/ICASSP48485.2024.10447156.
- [5] A. Jiang *et al.*, “Adaptive prototype learning for anomalous sound detection with partially known attributes,” in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5, doi: 10.1109/ICASSP49660.2025.10889514.
- [6] S. Kilickaya *et al.*, “Audio-based anomaly detection in industrial machines using deep one-class support vector data description,” in *2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems Companion (CIES Companion)*, pp. 1–5, 2025, doi: 10.1109/CIESCompanion65073.2025.11010815.
- [7] A. Garofalo, M. Rusci, F. Conti, D. Rossi, and L. Benini, “PULP-NN: A computing library for quantized neural network inference at the edge on RISC-V based parallel ultra low power clusters,” in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 33–36, doi: 10.1109/ICECS46596.2019.8965067.
- [8] J. Moosmann, M. Giordano, C. Vogt, and M. Magno, “TinyissimoYOLO: A quantized, low-memory footprint, TinyML object detection network for low power microcontrollers,” in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2023, pp. 1–5, doi: 10.1109/AICAS7966.2023.10168657.
- [9] L. Ruff *et al.*, “Deep one-class classification,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 4393–4402.
- [10] M.-H. Nguyen, D.-Q. Nguyen, D.-Q. Nguyen, C.-N. Pham, D. Bui, and H.-D. Han, “Deep convolutional variational autoencoder for anomalous sound detection,” in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, 2021, pp. 313–318, doi: 10.1109/ICCE48956.2021.9352085.
- [11] W. Dong, F. Guo, and T. Cheng, “Machine anomalous sound detection based on a multi-dimensional feature extraction self-encoder model,” in *2024 5th International Conference on Computer Engineering and Application (ICCEA)*, 2024, pp. 1165–1169, doi: 10.1109/ICCEA62105.2024.10604121.
- [12] T. Nishida, K. Dohi, T. Endo, M. Yamamoto, and Y. Kawaguchi, “Anomalous sound detection based on machine activity detection,” in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 269–273, doi: 10.23919/EUSIPCO55093.2022.9909901.
- [13] G. Wichern, A. Chakrabarty, Z.-Q. Wang, and J. L. Roux, “Anomalous sound detection using attentive neural processes,” in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 186–190, doi: 10.1109/WASPAA52581.2021.9632762.
- [14] F. G. Germain, G. Wichern, and J. Le Roux, “Hyperbolic unsupervised anomalous sound detection,” in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023, pp. 1–5, doi: 10.1109/WASPAA58266.2023.10248092.
- [15] H. Zhang *et al.*, “First-shot unsupervised anomalous sound detection with unknown anomalies estimated by metadata-assisted audio generation,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1271–1275, doi: 10.1109/ICASSP48485.2024.10448451.
- [16] Y. Liu, J. Guan, Q. Zhu, and W. Wang, “Anomalous sound detection using spectral-temporal information fusion,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 816–820, doi: 10.1109/ICASSP43922.2022.9747868.
- [17] H. Chen, Y. Song, L.-R. Dai, I. McLoughlin, and L. Liu, “Self-supervised representation learning for unsupervised anomalous sound detection under domain shift,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 471–475, doi: 10.1109/ICASSP43922.2022.9747863.
- [18] X.-M. Zeng *et al.*, “Joint generative-contrastive representation learning for anomalous sound detection,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095568.
- [19] H. Lan, Q. Zhu, J. Guan, Y. Wei, and W. Wang, “Hierarchical metadata information constrained self-supervised learning for anomalous sound detection under domain shift,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 7670–7674, doi: 10.1109/ICASSP48485.2024.10446044.
- [20] S. Huang, Z. Fang, and L. He, “Noise supervised contrastive learning and feature-perturbed for anomalous sound detection,” in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5, doi: 10.1109/ICASSP49660.2025.10888995.
- [21] R. David *et al.*, “TensorFlow Lite Micro: Embedded machine learning for TinyML systems,” *Proceedings of Machine Learning and Systems 3 (MLSys 2021)*, 2021, pp. 800–811.
- [22] J. Moosmann, H. Müller, N. Zimmerman, G. Rutishauser, L. Benini, and M. Magno, “Flexible and fully quantized lightweight TinyissimoYOLO for ultra-low-power edge systems,” *IEEE Access*, vol. 12, pp. 75093–75107, 2024, doi: 10.1109/ACCESS.2024.3404878.
- [23] B. McFee *et al.*, “librosa: Audio and music signal analysis in Python,” in *Proceedings of the 14th Python in Science Conference*

- (SciPy 2015), 2015, pp. 18–24, doi: 10.25080/Majora-7b98e3ed-003.
- [24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [25] B. Jacob *et al.*, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.
- [26] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

BIOGRAPHIES OF AUTHORS



Truong Minh Duc     received the B.Sc. and M.Sc. degrees in Electronics from the Posts and Telecommunications Institute of Technology (PTIT), Vietnam, in 2014 and 2022, respectively. He is currently a lecturer with the Faculty of Electronics Engineering 1 and AVIS Lab, PTIT. His research interests include embedded signal processing, TinyML deployment, and acoustic anomaly detection. He can be contacted at email: ductm@ptit.edu.vn.



Nguyen Ngoc Minh     is a lecturer with the Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology (PTIT), Vietnam. His research interests include telecommunications systems, embedded computing, and intelligent signal-processing applications. He can be contacted at email: minhnn@ptit.edu.vn.



Truong Cao Dung     is an associate professor with the Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology (PTIT), Vietnam. His research interests include electronic engineering, embedded systems, and applied signal processing. He can be contacted at email: dungtc@ptit.edu.vn.