

EchoEdge: A Lightweight Deep Learning and IoT Framework for Acoustic Machinery Diagnostics

R. Aravindhan¹, R. Pradeep Kumar², Mrs. Charulatha R.T³

^{1,2,3}Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India, ar7819@srmist.edu.in

Abstract

When a fan or pump stops without warning, the bill is rarely just the spare part. Plants still lose hours to that kind of failure. We listen to the machine instead of bolting on extra probes: a cheap microphone is enough to hear wear, imbalance or a leak before the asset seizes. EchoEdge is the working stack we assembled for that job. A small convolutional net reads log-mel spectrograms of industrial audio. A two-mode console sits on top. The whole thing also ships as a Windows folder that runs offline. Audio comes from MIMII, the public industrial-machine set on Zenodo (record 3384388). MIMII already blends machine sound with plant noise at three signal-to-noise ratios: +6 dB, 0 dB and -6 dB. One net is trained on all three mixes and then, for each 10 s clip, says normal or abnormal. On a held-out slice of unit id_00, mixed across those three ratios, we measured 96.71% accuracy, 97.48% precision and 95.08% recall on the abnormal class at $P(\text{abnormal}) \geq 0.40$. Streamlit drives the console. It can either walk through a live 10 s-per-clip demo, with staged faults and a mix of the three noise packs, or score one uploaded recording. Playback, a live spectrum, a latching silence alarm and the portable folder sit around that core. The numbers are for one unit under MIMII's controlled packs. They are enough for a lab demo and for operator practice on this unit. They are not a licence to drop the same weights on every machine in a plant.

Keywords: Machine Sound Diagnosis, Condition Monitoring, MIMII Corpus, Supervised Convolution, Mel Spectrogram, Noise-Robust Training, Maintenance Support, Local Dashboard, Offline Edge Package

1. Introduction

Fans, pumps and valves keep making sound while they turn. That sound is not leftover noise. Wear, imbalance, leakage and a few other faults leave marks in it. A microphone next to the asset is usually cheaper, and less invasive, than a vibration or current probe, so plants keep coming back to acoustic checks.

Most recent papers on this problem sit under anomalous sound detection (ASD). DCASE has run a machine-condition task (Task 2) for several years. The 2024 and 2025 notes ask a detector to catch a fault the first time a new machine is seen, and to keep working after the acoustic setting changes [1, 2]. Wilkinghoff and colleagues later surveyed how people handle that domain shift [11]. A lot of that literature stays unsupervised or self-supervised [3, 5, 9], mainly because labelled faults are scarce. We

took a narrower path. For one machine type where both healthy and faulty clips already exist, a two-class net is easier to run and easier to explain to a technician than an open-set novelty score.

EchoEdge is the name we gave to the small deep-learning and IoT stack we actually run. Four pieces share one weight file.

1. One convolutional net is trained on industrial audio from the three MIMII noise packs together (+6 dB, 0 dB, -6 dB). We ship one checkpoint, not one file per noise level.
2. We then measured what that mix is worth. A net that only ever saw -6 dB data misses most abnormal clips on its own split. The mixed-noise net keeps abnormal recall high on a noisy held-out mix.
3. A two-mode console sits around the net: a wall-clock 10 s simulation with staged fault stories and an optional noise mix, plus a one-file upload path for a field recording.
4. Playback is peak-scaled across output devices. A silence alarm latches instead of barking again and again. The live log-mel sits next to the decision. A Windows folder starts the stack without a site-wide Python install.

The rest of the write-up follows the usual research path, but we kept the operator pieces in view rather than burying them. Related ASD work is in Section 2. The pipeline and the MIMII slice we used are in Section 3. Features and the net are in Section 4. The console and the offline folder are in Section 5. Numbers are in Section 6. Limits are in Section 7. Section 8 is a short close.

2. Related Work

2.1 Industrial Anomalous Sound Detection

Task 2 of DCASE is still the usual yardstick when the test machine is new and the domain may have moved [1, 2]. The 2025 review by Wilkinghoff et al. walks through the domain-shift recipes that grew up around those editions [11]. A companion study asks a sharper question: how much does assuming a known machine identity actually buy you at test time [10]? Other groups come at the same setting from different corners. Lee et al. try to ignore interfering sources [3]. Zhong et al. push richer spectral front-ends [4]. Huang et al. use contrastive training to cut false alarms when the machine ID changes [5]. Where labels are thin, Zheng et al. fine-tune large audio models with low-rank adapters [6], and Wu et al. skip training with few-shot detectors [12].

2.2 MIMII-Oriented Evaluation and Unsupervised Alternatives

Real fault tapes are hard to collect. That is why Purohit et al. later built MIMII-Gen and MIMII-Agent on top of the MIMII recordings: a lab can synthesise extra test cases and compare detectors without waiting for another plant outage [7, 8]. If no abnormal labels exist at all, a one-class deep descriptor of healthy machine audio is still a reasonable fallback [9]. We stay with the original MIMII files, not those generative add-ons, because our setting already has both classes.

2.3 Positioning of EchoEdge

EchoEdge is not another first-shot unsupervised entry. Two choices set it apart. Where we already have labelled healthy and faulty clips for the target unit, we train a small supervised net and treat the output as a yes/no health flag, not an open-set novelty score. And we do not stop at a checkpoint. The same weights drive a simulation-first console and a portable offline folder, so a student or a technician can hear multi-noise detection, watch a fault latch in, and try an uploaded clip without wiring a microphone on the shop floor. The folder is meant for a classroom PC or a booth demo.

3. System Architecture and Dataset

3.1 End-to-End Architecture

Figure 1 is the whole path. Offline, MIMII WAVs from the three noise packs become log-mel spectrograms and one net is trained on the mix. Once that model is frozen, the console does one of two things: it either walks through a queue of 10 s clips, or it waits for a single uploaded file. Every clip goes through the same classifier. Outputs land in a telemetry table, a health gauge, an alert list, a spectrum pane, and—if someone turns it on—the speaker. Figure 2 zooms in on that per-clip path and shows where the silence alarm latches until an operator acknowledges it.

Figure 1. EchoEdge path: MIMII audio at three noise levels, CNN training, the two-mode console, and the offline Windows folder.

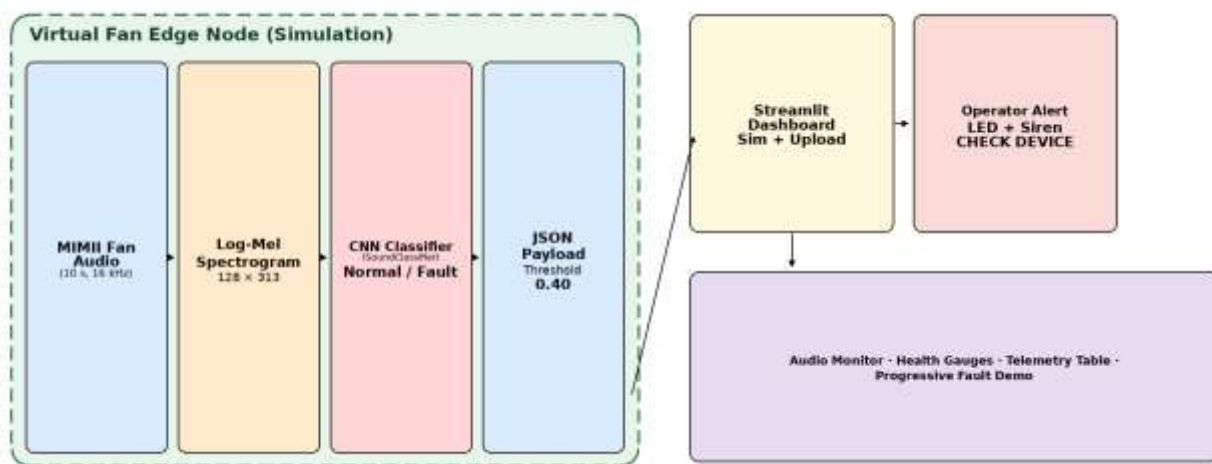
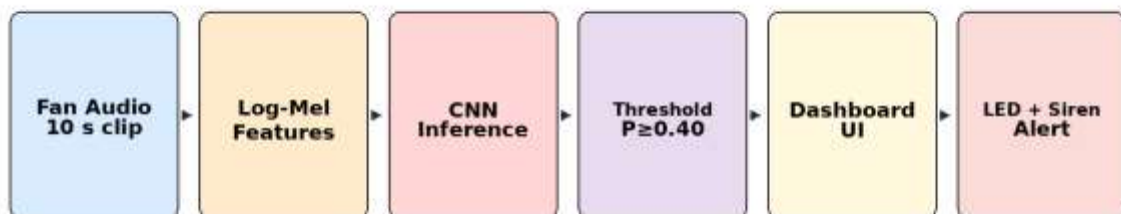


Figure 2. Runtime path for one clip: log-mel features, CNN score, 0.40 cut, dashboard update, and the latching LED/siren with manual re-arm.



3.2 Dataset

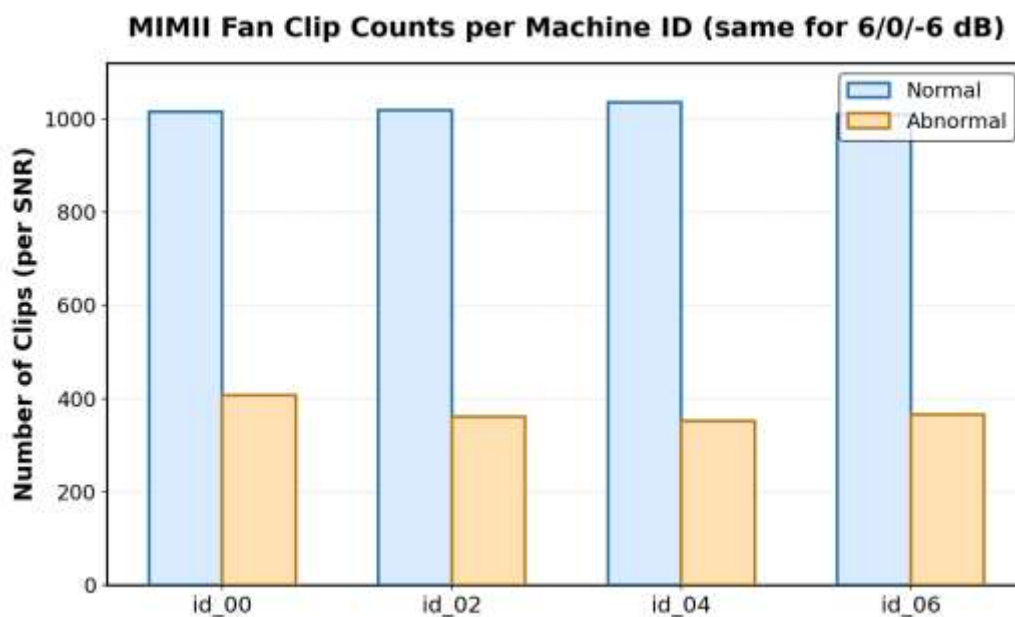
We train and test on the public MIMII industrial-machine set (Zenodo 3384388). The corpus is reused in DCASE-style ASD work because it already mixes machine audio with plant noise at several ratios [1, 2]. MIMII-Gen and MIMII-Agent were later built on the same tapes [7, 8]; we stay with the original files. Three compressed packs—+6 dB, 0 dB and −6 dB—were downloaded and checked by MD5. Together they are about 29.3 GB. We work with one rotating-machine class from those packs.

Each pack is organised by machine unit ID. Inside every unit folder sit normal/ and abnormal/ directories of 10 s, 16 kHz multi-channel WAV files. To keep the first study controlled we train and score mainly on unit id_00:

- Per SNR: 1011 normal + 407 abnormal clips.
- Across three SNRs: $3 \times 1418 = 4254$ clips for id_00.
- Full evaluated machine class (all unit IDs, all SNRs): 16650 WAV files (5550 per SNR).

Figure 3 plots the class counts for id_00 (the same 1011 / 407 split repeats at every noise pack). Multi-channel files are collapsed to mono by averaging channels before any spectrogram is computed. The SNR tags are part of how MIMII was recorded, not something we added. Both people and models find the -6 dB mix much harder; that matches what other ASD papers report when plant noise rises [3, 11]. Nothing in the code is tied to fans specifically. The same steps can be pointed at another equipment class, or at site recordings, once those recordings are labelled.

Figure 3. Clip counts per machine ID in the evaluated class (same 1011 normal / 407 abnormal split at +6 dB, 0 dB and -6 dB).



4. Signal Processing and Classification Model

4.1 Feature Extraction

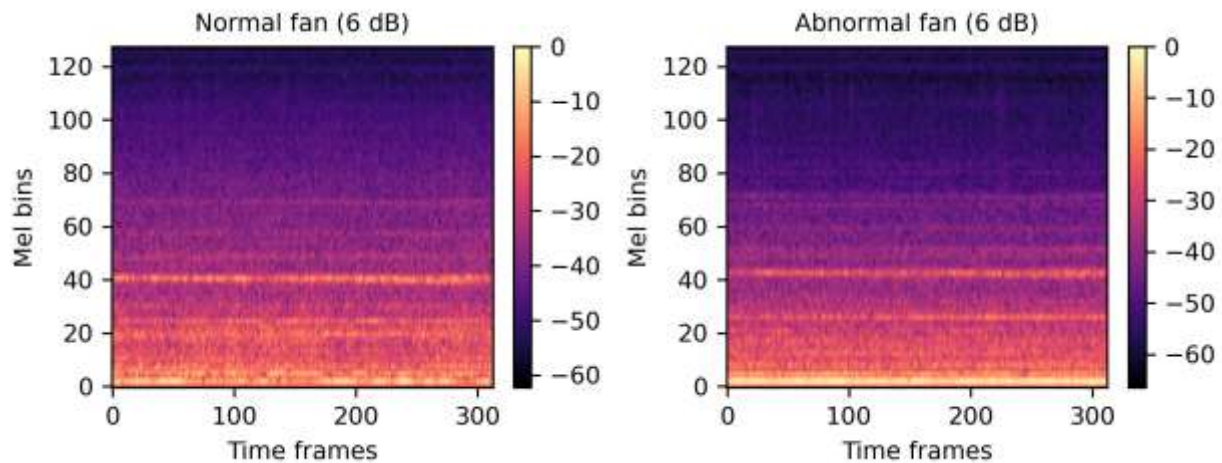
A 10 s mono clip $x[n]$, sampled at $f_s = 16$ kHz, is mapped to a log-mel spectrogram with

$$S = \text{LogMel}(x; N_{\text{FFT}} = 1024, H = 512, M = 128) \quad (1)$$

$$S \in \mathbb{R}^{128 \times 313} \quad (2)$$

During training we compute the spectrogram as we go and cache running mean and standard deviation. At deploy time those two vectors are read from mel_stats.json so the live console normalises the same way the trainer did. Figure 4 puts a normal clip next to an abnormal one at each of the three noise packs. At -6 dB the machine lines are mostly buried in plant noise, which is why we refused to train only on the clean +6 dB pack. Other groups keep pushing richer spectral front-ends and large pre-trained audio towers [4, 6]. We stayed with log-mel plus a small CNN because the whole thing has to run on a laptop without a GPU.

Figure 4. Log-mel views of one normal and one abnormal clip at +6 dB, 0 dB and –6 dB. The fault pattern is hardest to see in the –6 dB pair.



4.2 CNN Architecture

SoundClassifier is a small PyTorch CNN. The stack is:

- Three conv stages, 16 then 32 then 64 maps. Each stage is a convolution, followed by a batch-norm layer, a ReLU, then a max-pool.
- An adaptive average pool then collapses the spatial grid to one fixed size before the dense layers.
- A two-layer head: Linear(128), dropout 0.3, Linear(2).

Healthy clips outnumber faulty ones by about 2.5 to 1, so we use class-weighted cross-entropy. The optimiser is Adam. Every number in this paper comes from a CPU training run; we did that so the experiment can be repeated on a lab PC. A deeper or pre-trained front-end would almost certainly score higher [4, 6], but it would also make the portable folder heavier.

4.3 Decision Rule

Softmax gives $P(\text{abnormal})$. We call the clip abnormal when

$$\hat{y} = \text{abnormal} \quad \text{if} \quad P(\text{abnormal}) \geq 0.40 \quad (3)$$

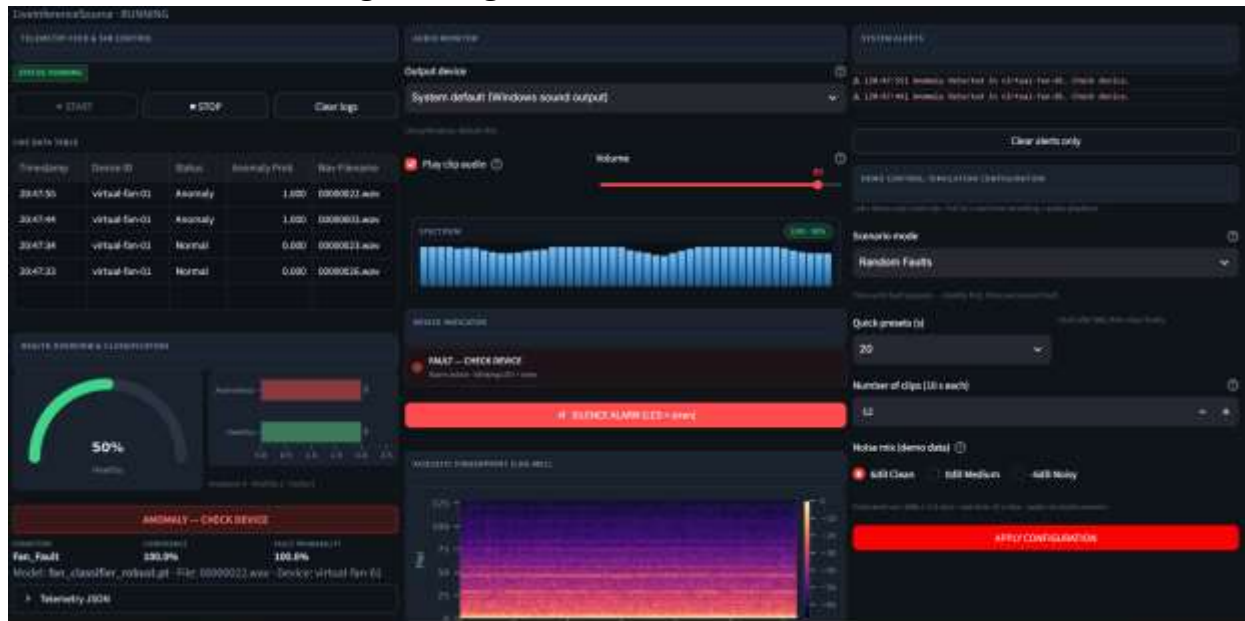
and healthy otherwise. Forty percent is a bit below the usual 0.50 cut. We moved it down after looking at the mixed-noise validation curve: the extra recall matters more to an operator than a handful of extra false alarms, because a missed fault can become downtime, whereas a false alarm can be dismissed. A site that hates nuisance sirens can push the cut back up and re-read the trade-off. They should not copy 0.40 blindly.

5. Dashboard and Deployment

5.1 Dual-Mode Operator Interface

The console is a Streamlit app bound to <http://localhost:8501> on the local machine. Figure 5 is a screenshot of the three-column layout operators see in a live simulation.

Figure 5. Live simulation view. Left: telemetry and health gauges. Centre: playback, device lamp and log-mel. Right: alerts and the scenario controls.



Simulation mode. The operator picks a scenario and hits START. Clips advance in real time—ten seconds of wall clock per file, no fast-forward—so the cadence matches a real MIMII stream. Three canned stories are built in:

- Full Abnormal: every clip is drawn from the abnormal folders.
- Pure Normal: every clip is drawn from the normal folders.
- Random Faults (progressive): the session stays healthy for a user-set time T, then locks into a permanent fault. That is a crude stand-in for a machine that degrades and then stays broken.

A checkbox mix can pull demo clips from any combination of the +6 dB, 0 dB and −6 dB packs. We use that to show that the single robust model does not fall over when the noise level jumps mid-session.

Upload mode. The operator drops in one WAV (or another common audio type). The app scores that file once and shows the result in plain language. SNR tags and corpus folder names stay hidden, so someone who has never heard of MIMII can still use it as a simple “is this recording healthy?” check.

5.2 Audio, Alarms, and Operator Safety

Raw MIMII peaks sit around 0.02, which is inaudible on most laptop speakers. Before the clip reaches sounddevice we peak-scale it. The user can keep the Windows default output or pick a specific device.

If a clip is called abnormal, a silence alarm can latch. It does not fire again on every later abnormal clip. It only arms again after the status has gone back to healthy and then failed, or after the operator presses Re-arm. That stops the siren from barking for the whole length of a sustained fault.

5.3 Portable Offline Package

For a talk or a lab session we ship a folder that already contains Python 3.12, the application, the robust weights, mel_stats.json, and 180 sample WAVs (30 normal and 30 abnormal at each of the three noise packs, all from id_00). Double-clicking the launcher starts the dashboard; nothing has to be installed system-wide. The folder is about 1.7 GB and is meant to be copied as-is onto the target PC. This packaging is a convenience for demos, not a claim that portable runtimes advance few-shot or domain-robust detection on their own.

6. Experiments and Results

6.1 Experimental Setup

Unless a paragraph says otherwise, every number below is for machine unit id_00. Each training corpus is split 80/20, stratified on the label. We trained four nets:

1. +6 dB-only: trained and checked only on the +6 dB pack.
2. 0 dB-only: trained and checked only on the 0 dB pack.
3. -6 dB-only: trained and checked only on the -6 dB pack.
4. Robust: trained on the union of all three packs and checked on a mixed held-out set.

Architecture, features and the 0.40 cut are the same in all four runs.

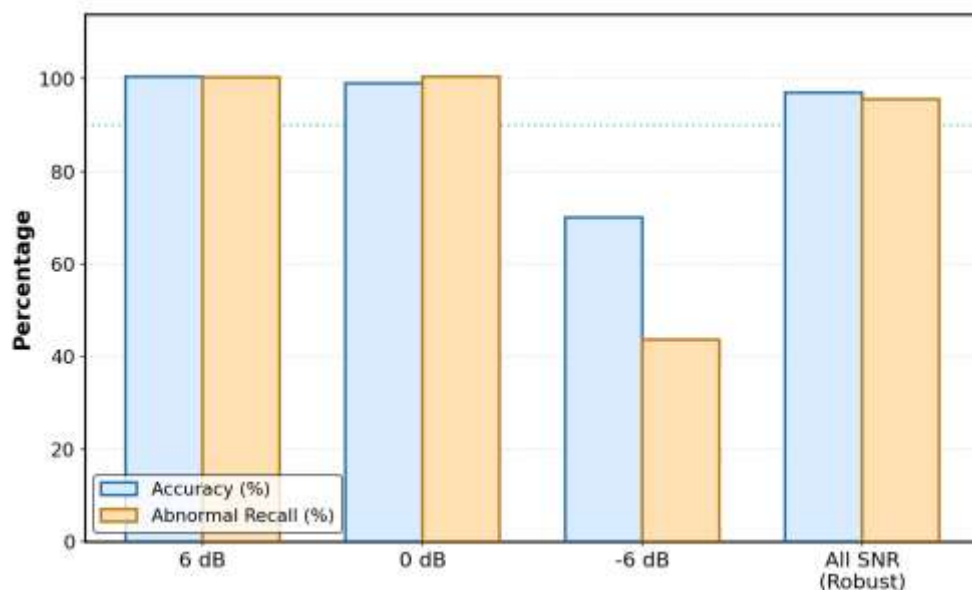
6.2 Single-SNR versus Multi-SNR Performance

Table 1 and Figure 6 put the four validation results next to each other. We deploy the mixed-noise net because, on the mixed held-out set of id_00, it is the only one that keeps both accuracy and abnormal recall in a range we would show an operator.

Table 1. Validation metrics by training regime (machine unit id_00, cut 0.40)

Model	Acc. (%)	Prec. (%)	Abn. Rec. (%)	Notes
6 dB only	100.0	100.0	100.0	High SNR
0 dB only	98.9	97.6	100.0	Moderate SNR
-6 dB only	69.8	79.6	43.2	Noise-limited
Robust (all SNR)	96.71	97.48	95.08	Deployed

Figure 6. Accuracy and abnormal recall for the three single-pack nets and the mixed-noise (robust) net.



What stands out in that table:

- At +6 dB the CNN is essentially done: every validation clip is classified correctly. That pack is the easy end of MIMII.

- A net that only ever trained on -6 dB data reaches just 43.2% abnormal recall on its own -6 dB validation split. Accuracy falls to 69.8%. We read this as the plant-noise floor hiding most of the fault cue from this small log-mel CNN, which is consistent with how other ASD studies describe low-SNR mixes [3, 11]. That is an observation about this architecture and this feature set, not a physical claim that the noise “dominates the signal” in every detector.
- Training on all three packs together recovers 95.08% abnormal recall and 96.71% accuracy on the mixed held-out set of id_00. That is the model we ship. It is still one file, and it is far safer than the -6 dB-only checkpoint for a demo that must not miss faults.

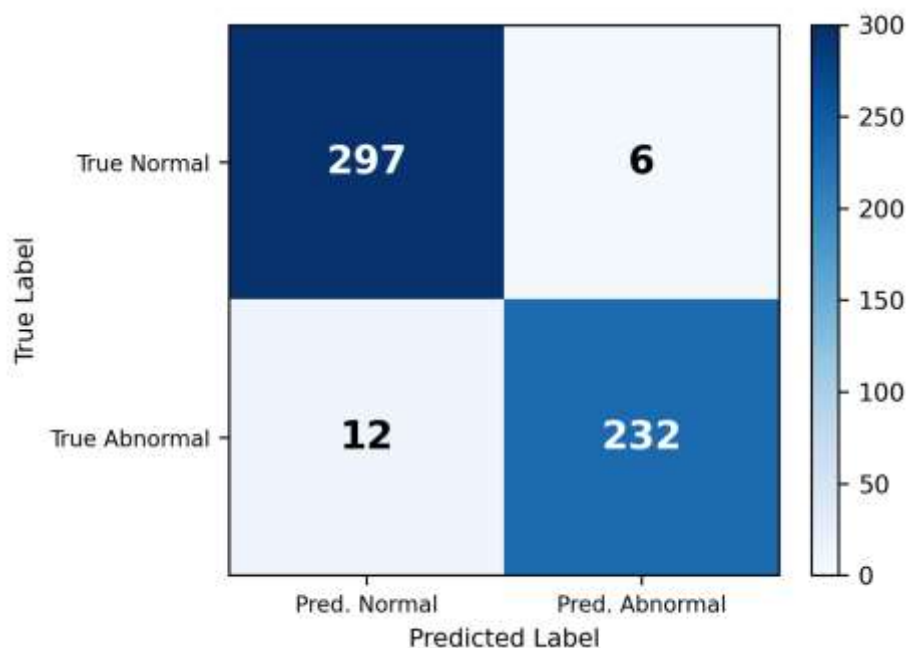
6.3 Confusion Matrix (Robust Model)

Table 2 lists the raw counts for the robust model on the mixed-noise validation split of id_00. Figure 7 is the same matrix drawn as a heatmap.

Table 2. Confusion counts for the robust mixed-noise model (id_00 validation)

	Predicted Normal	Predicted Abnormal
Actual Normal	TN = 297	FP = 6
Actual Abnormal	FN = 12	TP = 232

Figure 7. Confusion matrix of the robust mixed-noise classifier on the mixed held-out split.



From those four cells:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 96.71\% \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} = 97.48\% \quad (5)$$

$$\text{Recall}_{\text{abn}} = \frac{TP}{TP + FN} = 95.08\% \quad (6)$$

Twelve false negatives are the number we care about most: those are abnormal clips the operator would not have been warned about. Six false positives are small enough that, together with the latching alarm, they should not wear the operator out on this validation split.

6.4 Comparison with Unsupervised Baselines

If a site has almost no labelled faults, unsupervised ASD is still the right first tool. One-class deep SVDD [9], contrastive setups [5], and the first-shot DCASE systems [1, 2] were all built for that case. The catch, for an operator-facing binary lamp, is that those detectors are usually tuned to keep false alarms down, and that tuning can hide weak faults. We already have both labels for id_00, so we stay with the supervised mixed-noise CNN. MIMII-Gen / MIMII-Agent [7, 8] and training-free few-shot detectors [12] are useful next steps; they are not what EchoEdge runs today.

6.5 System-Level Evaluation

The console was also tried the way an operator would use it, not just scored offline.

- The simulation really does wait 10 s per clip, so the detection cadence matches the MIMII file length.
- The progressive-fault story is easy to narrate in a lab: the machine is fine, then it is not, and it stays that way.
- Upload mode gives one answer and does not ask the user which noise pack or unit folder the file came from.
- The portable folder started on a clean Windows machine that did not have Python on the PATH.

7. Discussion and Limitations

Unit specificity. The strongest numbers in this paper belong to one machine unit, id_00. We have not shown that the same weights survive a change of unit ID, let alone a change of machine type. That limit is not a surprise: machine identity is known to move ASD scores around at test time [10], and domain shift is the main theme of recent DCASE reviews [11]. Anyone who wants to take EchoEdge onto a different asset will need to relabel and retrain, or at least fine-tune.

Simulation versus live microphones. The system is simulation-first. A live microphone can reuse the same feature code and the same CNN, but we have not done the acoustic calibration or the plant-floor validation that would make a live-sensor claim honest.

SNR realism. MIMII's noise packs are carefully mixed. A real hall will have different spectra and bursts that are not stationary [3]. Treating our -6 dB result as a plant-noise guarantee would be a mistake.

Threshold choice. The 0.40 cut is a recall-first choice for this validation set. A site that pays more for a false siren than for a missed fault should move the cut and re-read Table 1, not copy our number.

Compute. Training on CPU was fine for this model. If we scale to several units or several machine types, a GPU run would save calendar time; it would not change the architecture.

8. Conclusion

We described EchoEdge as we actually use it: one mixed-noise CNN on log-mel spectrograms, a two-mode Streamlit console, and a Windows folder that runs without a developer install. On the held-out mix of $+6$ dB, 0 dB and -6 dB clips from machine unit id_00, the robust net reached 96.71% accuracy and 95.08% abnormal recall. The -6 dB-only net, by contrast, recalled only 43.2% of abnormal clips on its own split. Those figures justify using the mixed-noise weights for teaching, operator practice, and

further work on this unit. They do not, by themselves, justify a live-sensor rollout across other plant equipment. That next step needs new labels, a microphone calibration, and a noise environment that is not MIMII's.

Acknowledgement

We are grateful to CSE at SRMIST, Vadapalani campus, for lab time and supervision. The MIMII recordings (Zenodo 3384388) are used under the terms posted by the dataset authors. The DCASE Task 2 write-ups [1, 2] are the documents we followed when we needed the current ASD evaluation vocabulary.

References

1. T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, Y. Kawaguchi, "Description and discussion on DCASE 2025 Challenge Task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring", Proceedings of the DCASE Workshop, Barcelona, Spain, 2025. <https://arxiv.org/abs/2506.10097>
2. T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, Y. Kawaguchi, "Description and discussion on DCASE 2024 Challenge Task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring", Proceedings of the DCASE Workshop, 2024. <https://arxiv.org/abs/2406.07250>
3. Y. Lee, J. Kim, J. Ok, "Activity-guided industrial anomalous sound detection against interferences", arXiv:2409.01885, 2024. <https://arxiv.org/abs/2409.01885>
4. G. Zhong, Q. Wang, J. Du, L. Wang, M. Cai, X. Fang, "An enhanced audio feature tailored for anomalous sound detection based on pre-trained models", arXiv:2508.15334, 2025. <https://arxiv.org/abs/2508.15334>
5. S. Huang, Z. Fang, L. He, "Noise supervised contrastive learning and feature-perturbed for anomalous sound detection", arXiv:2509.13853, 2025. <https://arxiv.org/abs/2509.13853>
6. X. Zheng, A. Jiang, B. Han, Y. Qian, P. Fan, J. Liu, W.-Q. Zhang, "Improving anomalous sound detection via low-rank adaptation fine-tuning of pre-trained audio models", arXiv:2409.07016, 2024. <https://arxiv.org/abs/2409.07016>
7. H. Purohit, T. Nishida, K. Dohi, T. Endo, Y. Kawaguchi, "MIMII-Gen: Generative modeling approach for simulated evaluation of anomalous sound detection system", arXiv:2409.18542, 2024. <https://arxiv.org/abs/2409.18542>
8. H. Purohit, T. Nishida, K. Dohi, T. Endo, Y. Kawaguchi, "MIMII-Agent: Leveraging LLMs with function calling for relative evaluation of anomalous sound detection", arXiv:2507.20666, 2025. <https://arxiv.org/abs/2507.20666>
9. S. Kilickaya, et al., "Audio-based anomaly detection in industrial machines using deep one-class support vector data description", arXiv:2412.10792, 2024. <https://arxiv.org/abs/2412.10792>
10. K. Wilkinghoff, K. Imoto, Z.-H. Tan, "How much does machine identity matter in anomalous sound detection at test time?", arXiv:2602.16253, 2026. <https://arxiv.org/abs/2602.16253>
11. K. Wilkinghoff, T. Fujimura, K. Imoto, J. Le Roux, Z.-H. Tan, T. Toda, "Handling domain shifts for anomalous sound detection: A review of DCASE-related work", Mitsubishi Electric Research Laboratories, Technical Report TR2025-157, Nov. 2025. <https://www.merl.com/publications/docs/TR2025-157.pdf>

12. H.-H. Wu, W.-C. Lin, A. Kumar, L. Bondi, S. Ghaffarzadegan, J. P. Bello, “Towards few-shot training-free anomaly sound detection”, Proceedings of Interspeech, 2025. https://www.isca-archive.org/interspeech_2025/wu25b_interspeech.pdf