MASTER TABLE 2 (PREPROCESSING, MODEL & FUTURE)

Number Id of Paper	Data Preprocessing		Modelling & Algorithms						Evaluation		Experimental Setup & Resources			
	Data Preprocessing	Noise Types	Type of ML	Method	Model	Loss Function	Optimizer	Activation Function	Performance Metrics	Results	Real-time or Offline	Software / Tools (Python, MATLAB)	GPU / CPU / etc	Computational time
1	Bandpass filtering, automatic gain control (AGC)	Environmental, Instrumental	Conventional Techniques	Template Matching	Template Matching	N/A	N/A	ReLU	Accuracy	99.4%	Offline	Python	N/A	N/A
2	Splitting into chunks with ghost zones for parallel computation	Background noise, local environmental vibrations, vehicles, persistent vibrating sources	Conventional Techniques	Local similarity thresholding	Local similarity thresholding	N/A	N/A	N/A	Parallel Efficiency, Computational Runtime	95%	Offline	MATLAB, Python	CPU	10 of hours to 10 of seconds (935x speedup with 1000 CPU cores)
3	Detrending, bandpass filtering	Environmental Noise, Instrumental Noise	Unsupervised	CNN	CNN	N/A	N/A	N/A	Accuracy	99.4%	Offline	N/A	N/A	N/A
4	Gradient map auto-calibration, noise filtration (Savitzky–Golay), peak detection, thresholding	Microseismic, environmental noise, Gaussian noise	Conventional Techniques	Threshold-based event detection	STA/LTA	N/A	N/A	N/A	Response time, accuracy, efficiency	Real-time detection (micro-second accuracy)	Real-time	MATLAB, Python	CPU	Micro-second detection capability
5	Bandpass filtering (5 Hz - 15 kHz), grayscale image conversion	Environmental and coherent fading noise	Supervised	CNN	Inceptiov, AlexNet, LeNet, VGGNet GoogLeNet ResNet	Cross-entropy	RMSProp	ReLU	Accuracy, training speed (steps/s)	96.67%, 35.61 Training Speed	Offline	Python	GPU: NVIDIA Titan X	retraining time ~7 minutes
6	Detrending, bandpass filtering	Urban noise, traffic noise	Unsupervised	CNN	ResNet	N/A	N/A	N/A	Accuracy, cross-correlation coefficient	99.4%	Offline	N/A	N/A	N/A
7	Normalization, binary labelling of P/S phases	Environmental and measurement noise	Supervised	Hybrid CNN+RNN	ARRAYUDF (GRU)	Cross-entropy	N/A	ReLU, Softmax, Tanh & Sigmoid	Hit rate, FAR	Hit rate: 99.83% (P), 93.50% (S); FAR: 0.02% (P), 3.22% (S)	Offline	Python	GPU: NVIDIA V100	Prediction time < 20 seconds
8	Band-pass filtering, normalization, downsampling, noise injection	Ambient noise, instrumental noise	Supervised	CNN	CNN	Binary cross-entropy	N/A	ReLU, Sigmoid	Confusion matrix (TP, FP, FN), Comparison with STA/LTA	(CNN) vs. 218/959 (STA/LTA), 34%	Offline	Python	GPU	Real-time processing
9	Noise correlation, spectral whitening, filtering, interferometry	Ambient noise, instrumental noise	N/A	N/A	N/A	N/A	N/A	N/A	Model fit (L1 misfit in Monte Carlo inversion)	Depth error ±0.8 m; VS30 range 210–280 m/s,	Offline	Python, MATLAB	GPU	Near-real-time
10	Normalization, GAN-based synthetic data refinement	Ambient noise, instrumental and phase noise	Supervised	GAN	GAN	Cross-entropy, L1 loss	Stochastic Gradient Descent (SGD)	ReLU	Accuracy, False Alarm Rate, F1-score	Accuracy: 94–100%, F1: 89.85% (5 km), 87.23% (20 km)	Offline	Python	GPU: NVIDIA GTX 1080Ti	100 MS per fiber segment
11	In-phase/quadrature demodulation, phase extraction, unwrapping, direct current removal	Gaussian noise, instrumental noise	Supervised	ELM, SVM	ELM, SVM	N/A	N/A	N/A	Accuracy, training time, Testing time	ELM-RBF acc: 93.9%, training time: 0.87 s, testing time: 1.07 s	Offline	MATLAB, Python	CPU	Real-time (milliseconds)
12	Downsampling, low-pass filtering, median filtering	DAS noise, hydraulic fracturing noise	Supervised	ANN	ANN	N/A	Back-propagation algorithm	ReLU	Accuracy, F1 score	F1 score: 85% training, 86% val, 79% test set	Offline	Python	N/A	Real-time capable
13	No preprocessing required (raw data used)	Realistic DAS field noise	Supervised	CNN	Yolov3	MSE, Binary cross-entropy	Not mentioned	Sigmoid	Accuracy, Precision, Recall, False detection rate	>80% events identified, FAR: 2%	Offline	Python	GPU	Near real-time
14	Frequency band-pass filtering (1-5 Hz), downsampling	Ambient urban noise, instrumental noise	Supervised	CNN	ResNet	N/A	N/A	ReLU	Accuracy, Loss	Accuracy: 94.67%, Loss: 0.19037	Offline	Python	GPU	<30 mins (parallel computing)

15	Synthetic waveform generation, optical phase simulation	Optical system noise, seismic propagation	Supervised	CNN	FiberNet	N/A	N/A	ReLU	Accuracy, Confusion Matrix	92.8% accuracy	Offline	MATLAB, Python	GPU: NVIDIA GTX 1080Ti	Real-time capable
16	Band-pass filtering (1-5 Hz), data windowing, downsampling	Urban environmental noise, ambient noise	Supervised	CNN	VGG16	N/A	N/A	ReLU, Softmax	Accuracy, Confusion Matrix	83.69%	Offline	Python	GPU	N/A
17	Noise reduction, pulse scanning imaging, envelope extraction	Environmental noise, instrumental noise	Supervised	CNN	CNN	N/A	N/A	ReLU, Softmax	Accuracy, Recall rate, F-measure	Accuracy: 98.2%, Recall: 88.4%, F-measure: 93%	Offline	Python, MATLAB	N/A	0.6 seconds per recognition
18	Data windowing, normalization, label vector design	Background seismic noise, instrumental noise	Supervised	Hybrid CNN+RNN	SC-PSNET	N/A	N/A	ReLU Softmax	Hit rate, FAR, ME, MSE	Hit rate (P: 99.83%, S: 93.50%), FAR (P: 0.02%, S: 3.22%)	Offline	Python	GPU	Near-real-time
19	Resizing and normalization of seismic gathers	Instrumental noise	Unsupervised	Fuzzy clustering (FCM)	Fuzzy clustering (FCM)	N/A	N/A	N/A	Visual comparison with human judgment	Good match with visual inspection, accurate transitional	Offline	Python	N/A	N/A
20	N/A	Random and coherent noise	Unsupervised	Shifted-Matrix Decomposition (SMD)	N/A	N/A	Pattern search	N/A	Dot product between original and reconstructed matrices	Dot product maximized; 80% SNR improved from 1.9 to 4.7 (80%) and 12.3 (95%)	Offline	N/A	N/A	N/A
21	Bandpass filtering, scaling, RGB image transformation from spatial-temporal matrix	Environmental noise, instrumental noise	Transfer Learning	CNN with Transfer Learning	AlexNet	N/A	Stochastic Gradient Descent with Momentum	ReLU, Softmax	Accuracy, recall, F-measure	Up to 96.16% (4254 samples), 95.56% (1146 samples)	Offline + Online	Python, MATLAB	GPU: NVIDIA GTX 1050Ti	< 5 minutes training time
22	Detrend, demean, bandpass filtering, spectral whitening	Ambient noise	N/A	N/A	N/A	N/A	N/A	N/A	Cross-correlation	N/A	Offline	MATLAB	N/A	N/A
23	Noise addition, data normalization, generation of strain-rate images	Realistic field noise	Supervised	CNN	CNN	Binary cross-entropy, MSE	Adam	ReLU, Sigmoid, Linear	F1 score, accuracy, confusion matrix, MSE	F1: 0.994–1.000; R^2: 0.997–0.998	Offline	Python	GPU	N/A
24	Downsampling, bandpass filtering, median filtering	Field DAS noise	Supervised	CNN	CNN	N/A	Adam	ReLU Softmax	Accuracy, F1-score	Accuracy: 84%, F1-score: 0.85	Offline	Python	GPU	N/A
25	Stacking, RMS normalization, low-pass filtering, 2D median filtering	Ambient noise, instrumental noise, optical noise	Supervised	CNN	CNN	MSE	Adam	ReLU, Softmax	MSE, Standard Deviation, Residual Histograms	MAE: 2.04% x, 0.72% z, 2.76% Vp, 4.19% Vs, 0.97% density	Real time	Python	GPU: NVIDIA GTX 1080Ti	673 MS per 5000 samples
26	Synthetic noise added, random noise amplification	DAS coupling noise, environmental noise	Supervised	Hybrid GAN+CNN	CADN	MSE + Adversarial Loss	Adam	ReLU, Leaky ReLU	SNR, RMSE, ASPER	CADN achieved highest SNR improvement and lowest RMSE	Offline	N/A	GPU: NVIDIA GTX 1080	Training: 12.9 h
27	Cross-correlation with central reference trace	Random noise, coherent noise	Supervised	CNN	CNN	MSE, Cross-entropy	Stochastic Gradient Descent	tanh	Classification accuracy; Reg error	Accuracy: ~97.75% (V(z)); 96.1% (Otway): 15–50m	Offline	Python	GPU	Rapid
28	Median filtering, Butterworth bandpass filtering (0.5–1.0 Hz), decimation	Anthropogenic noise (vehicular), optical noise	Conventional Techniques	Traditional beamforming	Traditional beamforming	N/A	N/A	N/A	N/A	N/A	Offline	N/A	N/A	Real-time detection capability
29	Normalization, DAS-specific processing, event extraction	Field noise, instrumental noise	Unsupervised	CNN + GAN	CNN Autoencoder	N/A	N/A	N/A	N/A	N/A	Offline	Python	N/A	Near-real-time
30	Normalization, filtering, event segmentation, feature extraction	Land machinery, highways, wind turbines	Supervised	CNN, DT, RF, SVM, MLP, KNN	CNN, DT, RF, SVM, MLP, KNN	N/A	Stochastic gradient descent with momentum	ReLU, Softmax	Accuracy, Precision, Recall, F1-score, AUC, Execution Time, Delay	Accuracy ~99.9% for both MLP and CNN; CNN has 12× lower execution time	Real-time	Python, MATLAB	N/A	Significantly faster than classical methods
31	Numerical simulation of noisy signals	Laser frequency drift, phase noise, interference fading	Supervised	CNN	CNN	Log MSE	Gradient descent	N/A	SNR, Phase Standard Deviation (SD), SI-SDR	SNR ↑ from 13.4 dB to 42.8 dB; SI-SDR ↑ by 7.7 dB	Offline	N/A	N/A	N/A
32	Conversion into tensor format	Random noise and coherent noise	Unsupervised	HOSVD	HOSVD	N/A	N/A	N/A	Compression ratio, relative error	3D synthetic: 75% compression at ε = 0.0002, Field DAS: 70.2%, 4D tensor: 82.8% compression	Offline	Python	N/A	Few seconds (4.91 s)

33	Time-frequency transformation (S transform)	Low-frequency traffic noise, weak coupling, ambient noise	Supervised	CNN	CNN with attention gates	L1 loss, Local SSIM loss	Adam	ReLU, Sigmoid (Attention Gates)	PSNR, SSIM	PSNR 38 dB, SSIM 0.96	Real-time	Python	GPU: NVIDIA RTX 2080	150 MS per single patch
34	Normalization, bandpass filtering, scalogram generation	Background seismic noise	Supervised	CNN	SCALODEEP	Binary cross-entropy	Adam optimizer	ReLU, Sigmoid	Accuracy, Precision, Recall, F1-score, Specificity, NPV	Accuracy: 97.69%, Precision: 92.75%, Recall: 95.25%, F1-score: 94.00%, NPV: 98.88%, Specificity: 98.26%	Offline	Python	GPU	Real-time
35	Migration (RTM), data normalization, filtering	Ambient noise, instrumental noise	Supervised	CNN	U-Net	Dice coefficient	N/A	ReLU	Accuracy, Precision, Recall, F1-score, Posterior statistics for VOI	Horizontal DAS F1 = 0.939, Vertical DAS F1 = 0.919, Geophone F1 = 0.877	Offline	Python	GPU	Efficient
36	Bandpass filtering, segmentation into spatial-temporal blocks	Background noise, instrumental noise	Supervised	Hybrid CNN+BiLSTM	CNN + BiLSTM intrinsic denoising	N/A	N/A	ReLU	Effective Detection Rate (EDR), FAR, F1-score, Response time	EDR: 95.1%, FAR: 2.6%, F1-score: 0.926, Response time: 1.78s	Real-time	Python	GPU: NVIDIA Titan Pascal	Real-time (1.78 s)
37	Synthetic modelling	Random, fading, horizontal, checkerboard noise	Supervised	CNN	PA-MRNet	Charbonnier loss	N/A	PReLU, Sigmoid	SNR, SSIM, Visual Comparison	SNR improved to 28.21 dB (synthetic), lowest SSIM (0.0038) on field data	Offline	Python	GPU: NVIDIA GTX 1050Ti	32 hours
38	Bandpass filtering, STA/LTA normalization, data augmentation	Traffic noise, ambient noise, instrumental noise	Supervised	CNN	ADE-Net	Binary cross-entropy	Adam	ReLU, Sigmoid	Accuracy, Recall, F1-score	Accuracy: 80.4%, Recall: 93.18%,	Offline	Python	GPU	< 2.5 seconds per 10 s data
39	Data augmentation	Background noise, random, optical system noise	Supervised	CNN	GC-AB-Unet	MSE	Adam	LeakyReLU, Tanh at Attention Block	SNR	SNR = 14.25 dB	Offline	N/A	GPU: NVIDIA RTX 2080Ti	1.28 hours
40	Noise addition, image normalization	Gaussian white noise	Supervised	CNN	CNN	Binary cross-entropy, MSE	Adam	ReLU, Sigmoid, Linear	Accuracy, F1-score, MSE, R^2	Accuracy 99%, F1-score 1, R^2 0.99	Real-time	Python	GPU	Real-time
41	Bandpass filtering, median filter, non-local means filtering	Background environmental, instrumental noise	Supervised	CNN	CNN	Cross-entropy	Adam	ReLU, Softmax	Accuracy, F1-score	Accuracy: 85% (high-SNR), 80% (complete dataset), F1-score: 0.85	Real-time	Python	GPU	Real-time
42	Noise extraction and modelling	Coupled noise	Supervised	CNN	CNN	MSE	Adam	ReLU	SNR, Visual inspection, FK spectrum	Synthetic Data SNR improved from 9.58 dB to 26.17 dB	Offline	MATLAB	GPU: NVIDIA GTX 1050Ti	32 hours
43	Data augmentation	Complex noise types	Supervised learning	CNN	CA-MSRNet	MSE	N/A	Leaky ReLU, ReLU, Sigmoid	SNR, MSE, f-k spectrum visual inspection, difference maps	20.9 dB, MSE down to 0.0018, DnCNN	Offline	MATLAB	GPU: NVIDIA GTX 1050Ti	N/A
44	GAN noise synthesis	Random, ringing, checkerboard noise	Supervised	Hybrid (CNN+GAN)	AD-DNNet	Mean squared error (CNN), Wasserstein loss, WGAN	SGD (WGAN), Mini-Batch Gradient Descent (CNN)	ReLU (CNN)	Average SNR improvement, qualitative visual comparison	SNR improvement up to 20.9 dB with AD-DnNet	Offline	Python	GPU	N/A
45	Compressive sensing encoding, image normalization	Environmental noise, random noise	Supervised	CNN	DCNN	MSE	Adamax	Swish	Location error (Euclidean), Moment tensor angular error	Location error < 5 km for 95% of events; MT normalized error 0.25	Near real-time capable	Python	GPU: NVIDIA Tesla A30	7 MS per data frame
46	Band-pass filtering (1–45 Hz), normalization	Seismic noise	Supervised	CNN, ANN, Hybrid (CNN+LSTM)	ANN, LSTM-CNN ResNet	Binary Cross-Entropy	Adam	ReLU, Sigmoid	Accuracy, Recall, Precision, F-score	CNN Acc: 96.94%, CNN+LSTM: 93.86%, FC-ANN: 90.17%	Offline	N/A	GPU: NVIDIA Titan RTX	ANN:1140.5s, CNN:186.3s, CNN+LSTM 1375.9s
47	Singular value decomposition	Random, optical, fading	Supervised	CNN	SVDDCNN (CNN with Singular spectrum)	Frobenius norm-based residual loss	N/A	ReLU	SNR, SSIM	Highest SNR: 12.4907 dB; SSIM: 7.9578×10^{-4}	Offline	MATLAB	GPU: NVIDIA GTX 1050Ti	27 hours 46 mins

48	Median subtraction, band-pass filter (10-200 Hz), decimation, normalization	Background noise	Supervised	CNN	VGGNet and ResNet	Cross-entropy loss	Adam with Bayesian optimization	ReLU and Leaky ReLU	Accuracy, Recall, Precision, AUC	Accuracy 98.6%, Precision 99.92%, Recall 97.58%, AUC: 99.72%	Offline	Python	GPU: NVIDIA V100	Less than 4 minutes for 20 hours data
49	Quadrature demodulation, data augmentation	DAS measurement noise	Supervised	CNN	DCNN	cross-entropy	Stochastic gradient descent with Adam momentum	Temporal-spatial	MIoU, IoU, SNR, Precision Rate (PR)	SNR up to 37.84 dB, MIoU up to 79.37%, PR up to 92.5%	Offline	Python	N/A	N/A
50	Noise extraction	Random, coupled, fading, horizontal, checkerboard noise	Supervised	CNN	DAS VSP Data Denoiser based CNN	MSE	Stochastic Gradient Descent	ReLU	Local SNR	20 dB SNR	Offline	MATLAB	GPU: NVIDIA GTX 1050Ti	N/A
51	Multipatch hierarchy data division	Random, fading, checkerboard, long-period noise	Supervised	CNN	MSDN	Charbonnier Loss + Edge Loss	Adam	Not mentioned	SNR, MSE, visual similarity	MSDN had highest SNR improvement and lowest MSE compared to other methods	Offline	Python	GPU: NVIDIA GTX 1650Ti	44.5 hours
52	Normalization, synthetic noise addition	Random noise, coherent noise	Supervised	CNN	SPSNet	Frobenius norm-based, MSE loss	Adam	ReLU	SNR, MSE, SSIM	Improved SNR: 22 dB, SPSNet outperformed Bandpass, VMD	Offline	Python	GPU: NVIDIA GTX 1050Ti	Training: 32 h
53	Median filtering, normalization, amplitude normalization	LF-DAS background noise	Supervised	RF, SVM, ANN	RF, SVM, ANN	N/A	N/A	N/A	Accuracy, Precision, Training time, Prediction time	RF: 93% acc, 98% precision; SVM: 90% acc; ANN: 93% acc	Real-time capable	Python	N/A	Short (Random Forest)
54	Detrending, resampling, normalization	Low SNR seismic noise	Supervised	CNN, ANN, Hybrid (CNN+LSTM)	CNN, ANN, Hybrid (CNN+LSTM)	Binary cross-entropy	Adam	ReLU, Sigmoid	Accuracy, F-score, TP/FP/FN/TN histogram	Up to 98.8% acc (FC-ANN); CNN: 97.7%, CNN+LSTM: 97.5%	Offline	N/A	N/A	N/A
55	Normalization, framing, short-time energy calculation	Gaussian noise, electronic interference	Supervised	SVM, KNN	SVM, KNN	N/A	N/A	N/A	Accuracy	Ranges from 81% to 94%	Offline	LIBSVM, Python	N/A	N/A
56	Detrending, normalization	Measurement noise	Supervised	CNN	Faster R-CNN	Multi-task loss: cross-entropy + regression (Smooth L1)	Adam	Softmax	mAP, ROC-AUC, accuracy	mAP = 0.5: 77%, accuracy: 98.8%	Designed for embedde d real-time use	Python	GPU: NVIDIA RTX A6000	124.923 ms on GPU, 5.347 s
57	Normalization, conversion to MTF images	Measurement noise	Supervised	CNN	MTF (Markov Transition Fields)	Cross-entropy loss	RMSProp	ReLU, Softmax	Accuracy, ROC, AUC, Loss	Accuracy: up to 99.5% train, 98.75% test; AUC	Offline	Python	GPU	average recognition time 0.3128 s
58	Median subtraction, band-pass filtering, decimation, normalization	DAS channel noise	Supervised	CNN	Mixed	Cross-entropy loss	Adam	N/A	Accuracy, Precision, Recall, AUC	Accuracy: 98.6%, Precision: 99.9%, Recall: 97.6%, AUC: 99.7%	Offline	Python	GPU	N/A
59	Median filtering, band-pass filtering, normalization	DAS noise, anthropogenic noise	Supervised	CNN	ResNet	MSE	Adam	ReLU, Linear	Accuracy, MAPE, Standard deviation	MAPE x=2.21%, z=0.61%; Std Dev 12m; Pearson r > 0.99	Real-time capable	Python	GPU: NVIDIA GTX 1080Ti	7.2–11.8 hours for training
60	Gaussian Pyramid downsampling, normalization	Coherent, fading, horizontal, checkerboard	Supervised	CNN	MPFAN	Charbonnier Loss + Laplacian-based loss	Adam	ReLU + Sigmoid	SNR, MSE, Local Correlation Coefficient, Peak Preservation Accuracy	MPFAN outperforms BPF, VMD, and HSI-DeNet across all metrics	Offline	Python	GPU: NVIDIA GTX 1050Ti	N/A
61	Normalization, Synthetic noise addition	Random, horizontal, coherent, fading noise	Supervised	CNN	MRP-Net	Frobenius Norm-based L2 Loss	Adam	Sigmoid, ReLU	SNR, RMSE	SNR improved by >23 dB at −10 dB initial SNR, lowest RMSE among all tested methods	Offline	Python	GPU: NVIDIA RTX 3070	Training: 9.24 h
62	Normalization, patching	ESP-induced coherent and random noise	Supervised	CNN	U-Net	N/A	N/A	N/A	Visual comparison and spectral similarity	ML-denoised output closely matched field data with ESP off	Offline	N/A	GPU	N/A
63	Normalization, synthetic noise addition	OTDR detector noise, electrical noise	Supervised	Hybrid (CNN-based Autoencoder & BiLSTM)	DCAE+BiLST M	MSE	N/A	ELU (Exponential Linear Unit)	MSE, SNR Improvement, RMSE, PRD, Accuracy, Localization Error	DCAE achieves up to 21 dB SNRimp at −5 dB input; BiLSTM achieves up to 96.37% accuracy with denoising	Offline	N/A	N/A	N/A

64	Synthetic noise modelling, normalization	Random, coherent DAS noise	Supervised	CNN	L-FM-CNN	New MSE loss function combined with Energy Ratio Matrix (ERM)	Small-batch gradient descent	Leaky ReLU	SNR, RMSE	SNR improvement up to 19 dB, RMSE reduced to 0.0066	Offline	Python	GPU: NVIDIA GTX 1050Ti	N/A
65	Synthetic noise modelling, normalization	Random, coherent, fading, horizontal noise	Supervised	CNN	RCEN	MSE	N/A	N/A	SNR, RMSE	RCEN: SNR = 18.12 dB, lowest RMSE	Offline	Python	GPU: NVIDIA RTX 3070	N/A
66	Bandpass filtering, normalization	Random noise, marine environment noise	Supervised	CNN	Residual CNN (ResUNet)	MSE	N/A	ReLU	RE (Relative Error), SSIM, MSE	CNN: RE = 0.2925, SSIM = 0.92, MSE = 0.0204 ResUNet: RE = 0.1604, SSIM = 0.96, MSE = 0.0074	Offline	Python	GPU	N/A
67	Windowing, FFT, normalization, resampling	Seismic noise, anthropogenic disturbances	Supervised	MLP	Multilayer Perceptron (MLP)	Cross-entropy loss	Adam	Tanh	Accuracy, precision, recall, F1-score, FoEW	Accuracy (89.74%–95.98%), F1-score (91%), FoEW (mean > 85% for MMI ≥ VI)	Real-time	Scikit-learn	CPU	1.55 MS per window processing time
68	Synchrosqueezing transform, normalization	Random noise, coherent noise, fading noise	Transfer Learning	CNN	CNN	Custom loss, L2 norm	Adam	ReLU	SNR, SSIM, visual inspection	SNR improvement: up to +20 dB; SSIM ~0.0056 for best denoising (lowest signal leakage)	Offline	Python	GPU: NVIDIA GTX 1080Ti	N/A
69	Median filtering, normalization, framing, short-time energy calculation	DAS measurement noise, Gaussian noise	Unsupervised	Autoencoder	Autoencoder	MSE	Adam	N/A	AUC and ΔSNR (SNR improvement)	AUC: up to ~99.9% for most events; ΔSNR: up to 15 dB	Offline	Python	N/A	Short, optimized for real-time applications
70	Mini-batch training, dropout, normalization	Gaussian noise, DAS measurement noise	Supervised	Hybrid Attention CNN - HACNN	CNN (HACNN)	Binary cross-entropy	Adam	Tanh	Accuracy, ROC-AUC	Accuracy = 99.92% (clean), 67.63% (SNR=0), ROC/AUC	Offline	Python	GPU: NVIDIA RTX 3090	N/A
71	Normalization, synthetic noise addition	Random, fading, coherent, horizontal noise	Supervised	GAN	Autoencoder, DuGAN (CNN-Based)	MSE + Adversarial Loss (Wasserstein GAN-GP)	Adam	ReLU, Leaky, ReLU	SNR, MSE, visual inspection (difference maps, spectra, local similarity)	DuGAN outperforms bandpass, F-X, SVD, DnCNN, U-Net, and original GAN on SNR and MSE	Offline	Python	GPU	N/A
72	Normalization, synthetic noise addition	Random noise, coherent noise, fading noise	Supervised	CNN	MDD-Net	L2 norm (MSE/Frobenius norm)	Adam	Sigmoid	SNR, RMSE	SNR +22.56 dB, RMSE ↓ to 0.07	Offline	Python	GPU	N/A
73	Normalization	Spatially incoherent, random noise	Self-Supervised	CNN	U-Net with J-invariance	MSE +L2 norm	Adam	Swish	Scaled variance of residuals, waveform coherence gain	Residual variance improves with SNR; coherence gain consistently > 1	Offline	Python	GPU	N/A
74	Normalization, FFT denoising	Oceanic background noise	Conventional Techniques	FK Filtering	FK Filtering	N/A	N/A	N/A	SNR, Classification Rate, Relative Error	SNR improvement: up to 8.42 dB; Magnitude estimation error: 0.43% to 13%; Smallest earthquake	Real-time	Python	N/A	N/A
75	Radon transform preprocessing, normalization	Random noise, coherent noise	Supervised	CNN	U-Net	L1 loss + MSE	Adam	ReLU	SNR, Absolute and Relative	Relative S/N: Up to 28.8 dB after 9 epochs	Offline	Python	GPU	N/A
76	Median removal, bandpass filtering, normalization	Coherent noise	Unsupervised	DBSCAN clustering	DBSCAN clustering	N/A	N/A	N/A	Detection accuracy, SNR	Detected more true events than benchmark methods, validated with geophones	Real-time	Python	CPU	3 seconds per 15-second data window on 20-core CPU
77	Median subtraction, detrending, band-pass filtering	Spurious signals, instrument noise	Unsupervised	Autoencoder and GMM clustering	Autoencoder and GMM clustering	MSE	Adam	ReLU, Linear	Silhouette coefficient, MSE	N/A	Offline	Python	GPU	Python
78	FK filtering, ray tracing, reflection extraction	Not noise-based, signal extraction	Conventional Techniques	FK Filtering	FK Filtering	N/A	N/A	N/A	N/A	N/A	Offline	Python	N/A	N/A
79	Remove mean, normalization, patching into 256×256 windows, data augmentation.	Random noise, fading noise, checkerboard noise, horizontal noise	Supervised	Hybrid (Transformer CNN)	Custom deep-learning method (CP-SANet) with self-attention	Charbonnier loss	AdamW	GELU and Leaky ReLU	SNR, MSE	S/N: 31.31 dB; MSE: 1.65e-4 (best among baselines)	Offline	Python	GPU: NVIDIA RTX 3090	Training took about 13.5 hr. Inference times, but GPU-accelerated.

80	Mean removal, normalization, split into patches (128×96), data augmentation via random flipping.	Strong random "blue noise" in DAS data; tests vs. bandpass, Wiener	Supervised	CNN	Noise2Noise (UNet)	MSE	Adam	N/A	Semblance-based SNR (local SNR estimates using Neidell & Taner method)	DAS-N2N yields highest SNR among all methods compared (bandpass, Wiener,	Offline	Python	GPU: NVIDIA RTX 2080Ti	Processes 30 s of data (1 km cable, 985 channels) in under 1 s on a single GPU (after training).
81	Normalization, down-sampling, synthetic noise addition	Coupled noise, horizontal, random, fading	Supervised	CNN	DPA-Net	MSE + L2 loss	Adam	SoftMax (attention module)	MSE loss	SNR, RMSE	Offline	Python	GPU: NVIDIA RTX 2060 Super	Training: 60 epochs
82	Normalization, resampling	High-frequency and low-frequency noise	Supervised	Hybrid (Bidirectional LSTM (RNN))	SERM	N/A	Adam	Sigmoid, Tanh	RMSE	RMSE: 0.806 (DL-based), 2.838 (physics-based)	Offline	Python	GPU	N/A
83	Matched filtering, synthetic perturbation generation	DAS intrinsic system noise, environmental noise	Supervised	CNN	CNN combined demodulation-denoising (GRU)	N/A	N/A	Tanh	Noise level in pε/√Hz, SNR gain, response linearity	Noise reduced from 215 pε/√Hz (SERM) to 74 pε/√Hz (DL); SNR gain up to 8.4 dB	Offline	N/A	GPU	Training: variable with early stopping
84	Synthetic noise modelling, random geological model generation	Random, coherent, optical, resonance, and tube waves	Supervised	CNN	RRCAN (regularized attention denoising)	MSE + TV + Kernel Norm	Adam	Leaky ReLU	PSNR, SSIM, Visual Comparison	RRCAN_1 outperformed DnCNN in PSNR and SSIM on synthetic and field data	Offline	Python	GPU	N/A
85	Patching, normalization	Random, horizontal, coupled, erratic noise	Supervised	CNN	ULDNet	MSE	Adam	Leaky ReLU	SNR, Visual Inspection	Synthetic: ULDNet = 12.23 dB; Benchmark: PATCHUNET = 9.48 dB, DDUL = 9.92 dB	Offline	Python	GPU	Training: 100 epochs
86	Patching, normalization	Random, horizontal, coupled, erratic noise	Supervised	CNN	FCDNet	MSE	Adam	ReLU, Linear	SNR, Local Similarity	FCDNet: 18.29 dB (synthetic), better preservation and less signal leakage than benchmarks	Offline	Python	GPU	Training: 100 epochs
87	Band-pass filtering, median filtering	Cultural, environmental, optical, coupling noise	Conventional Techniques	Moving-rank-reduction filtering	Moving-rank-reduction filtering	N/A	N/A	N/A	Detection performance	N/A	Offline	Python	N/A	High computational cost
88	Signal decomposition using EEMD	Coupled noise, horizontal noise, random noise	Supervised	Hybrid (Stacking ensemble (RF, GBDT, extra tree))	Stacking ensemble (RF, GBDT, extra tree)	Exponential loss, Logarithmic loss (GBDT)	N/A	N/A	SNR, RMS error	Best among compared methods: S/N = 7.70 dB, RMS = 0.0097	Offline	Python	N/A	N/A
89	Generate synthetic OTDR signals with random intensities, convolve with pulse. Add noise. For real data, measure real pulse shape, do repeated scanning.	General DOFS random noise in OTDR/BOTDR, up to strong amplitude fading noise.	Supervised	CNN	1DDCNN	N/A	N/A	ReLU	PSNR, Deconvolution Gain (DG)	DG: HQS = 15.8 dB, SSRNet = 15 dB, TVD = 7.9 dB	Offline	Python	GPU: NVIDIA Titan	10 hours to train each specialized 1DDCNN, total iteration for HQS is 10 steps.
90	Summation of synthetic signals + real noise to form noisy training pairs. Possibly random flips.	Multi-type strong noise in DAS data,	Supervised	GAN	Multi-scale adversarial approach (GAN) with U-Net	Content constraint + adversarial constraint	N/A	N/A	Qualitative visual comparison and signal preservation; some discussion on recovery in time/frequency domain	Demonstrated good signal preservation and denoising in figures	Offline	Python	N/A	N/A
91	Basic denoising for raw VSP, up/down going separation, align first arrivals. For ML: group nearby DAS channels to match single geophone.	Not purely denoising.	Supervised	LSTM	LSTM neural network or plain physics formula.	RMSE	Adam	ReLU	MSE, Spectrum Matching, RTM Image Alignment	MSE (DL: 1.34, Physics-based: 3.96), improved RTM phase alignment	Offline	Python	N/A	N/A
92	Patch extraction, produce "clean/noisy" pairs. Possibly random flips or normalizations. Then train.	Complex DAS noise: random, strong amplitude erratic, vertical/horizontal stripes.	Supervised	CNN	CNN-based (encoder–decoder).	N/A	Adam	N/A	SNR	From -7.29 dB to 26.6 dB improvement on synthetic data	Offline	Python	GPU: NVIDIA RTX 2080Ti	4 minutes total GPU time for training. Inference presumably near real-time.

93	Amplitude normalization, patch extraction, random combination of signal/noise	Background noise, fading noise, horizontal noise, checkerboard noise, optical	Supervised	CNN	ACPNet	L1 loss + MSE	Adam	ReLU, Sigmoid	SNR, Local Similarity Maps	SNR improved from 4.78 dB to 21.28 dB in best test case; superior to U-Net and DnCNN	Offline	N/A	GPU: NVIDIA GTX 1080Ti	N/A
94	Synthetic waveforms, noise augmentation, normalization	DAS background noise	Supervised	CNN	JointNet	Joint loss (MSE + Cross-entropy)	Stochastic Gradient Descent	ReLU	Accuracy, Precision, Recall, F1	Field DAS: Accuracy = 0.91, Recall = 1.0, Precision = 0.50, F1 = 0.67	Real-time	Python,	GPU	.5 s per file
95	Bias removal, band-pass filtering, normalization	Traffic noise, coupling noise	Supervised	CNN	U-Net	Cross-entropy	N/A	CNN: ReLU, Sigmoid; U-Net: ReLU, Softmax	Accuracy, precision, recall, F1-score	CNN: Accuracy 96.62%, Precision 98.73%, Recall 95.51%	Offline	Python	GPU	Near-real-time
96	Patch slicing (64×64), amplitude normalization, random removing traces to emulate incomplete data, combining real noise	Optical noise, fading noise, horizontal noise, random noise, plus missing traces	Supervised	CNN	MSI-Net	MSE + Frobenius Norm	Adam	ReLU	SNR, RMSE, Local SNR (LSNR)	SNR > 20 dB, RMSE = 0.1261 (best among all methods)	Offline	Python	GPU: NVIDIA RTX 3080	Training time around 7.79 h,
97	Amplitude normalization, patch slicing, data augmentation with random noise amplitude	Background noise (random, fading, horizontal, checkerboard, coupling	Supervised	CNN	Multi-scale parallel CNN with feature interactions (MSI-Net)	L2 loss + MSE	Adam	ReLU	SNR, MAE, MSE, SSIM	SNR: 19.0006 dB, MAE: 0.0274, MSE: 0.0014, SSIM: 0.4127	Offline	Python	GPU: NVIDIA GTX 1080Ti	Training cost not explicitly stated, but mention 200 epochs, some hours on GPU
98	Patch extraction, normalization, noise–signal superposition in different SNR levels	Complex background noise in DAS-VSP (random, fading, horizontal, checkerboard, optical, coupling)	Supervised	CNN	RGSA-Net (Recurrent-guided multi-scale CNN with attention blocks)	MSE	Adam	Sigmoid	SNR, RMS Error	RGSA-Net achieved highest: S/N improvement up to +23 dB, lowest RMS error among all methods	Offline	Python	GPU: NVIDIA GTX 1080Ti	Training time around 9.24 h, testing is fast (on GPU)
99	Patch extraction (64×64), random weighting, combine synthetic & noise	Time-varying optical, fading, horizontal noise, random	Supervised	CNN	MSR-Net	Frobenius norm (L2)	Adam	ReLU (used in CSFE module)	S/N (Signal-to-Noise Ratio), RMS Error	up to +29.51 dB S/N improvement, lowest RMS error	Offline	Python	GPU: NVIDIA RTX 3060	5.55 h training (60 epochs)
100	Split into 64×64 patches, random amplitude, pair with clean signals	Multicomponent noise (random, optical abnormal, horizontal, fading)	Supervised	GAN	AMGAN	MSE + Adversarial Loss	Adam	ReLU, Sigmoid	SNR, MSE	S/N: 16.3343 dB (best among compared methods); MSE: 0.0034	Offline	MATLAB	GPU: NVIDIA GTX 1050Ti	50 epochs training; time not explicitly given
101	Patch-based (256×256), random noise addition, pair with pure data	Random noise, optical abnormal noise, fading noise, horizontal noise	Supervised	CNN	ADNet	MSE	Adam	ReLU, Tanh	SNR, SSIM	ADNet: SNR ↑ to 15.3275 dB, SSIM ↑ to 0.7864 (better than VMD, band-pass, DnCNNs)	Offline	Python	GPU: NVIDIA GTX 1050Ti	Trained up to 1,000 epochs
102	Patch-based (64×64), random amplitude scale for noise	Complex DAS noise: time-variant optical noise, horizontal	Supervised	CNN	RMCHN	L2 norm + MSE	N/A	N/A	SNR, RMSE	RMCHN achieves highest SNR gain and lowest RMSE among methods (Table III)	Offline	N/A	N/A	N/A
103	Subsampling pairs from noisy data; random shift for self-supervision; amplitude normalization	Primarily random noise	Self-Supervised	CNN	Sample2Sample	Combined: L2 (Reconstruction Loss) + Regularization Loss	Adam	Leaky ReLU	SNR	SNR improved from 1.31 dB → 17.56 dB (Synthetic)	Offline	Python	GPU	N/A
104	Normalization, noise augmentation	Anthropogenic, instrumental, traffic	Semi Supervised	CNN	U-Net (PhaseNet-DAS)	uses Gaussian-shaped label smoothing	AdamW	ReLU	Phase pick accuracy, association rate, SNR, pick consistency	N/A	Offline	Python	GPU: NVIDIA V100	3.5 hours for 180 hours of data
105	Amplitude balancing, data augmentation, patch extraction	Incoherent random noise, ground roll, second-source land noise	Supervised	Hybrid: self-supervised + supervised	CNN (Inception U-Net)	Not mentioned	Not mentioned	Not mentioned	SSIM, NMAE, MAE, MSE	SSIM ~0.93-0.95; NMAE <5%	Offline	Python	GPU: NVIDIA A100	1 hour training on GPU, 1 s/gather inference

106	Amplitude normalization to [−1,1], patch creation (256×256)	Spatially correlated DAS noise, background noise in VSP data	Supervised	CNN	BSN	Self-supervised loss (modified MSE using asymmetric PD)	Adam	ReLU	SSIM, SNR, MSE	SNR and SSIM improved vs. baseline; MSE reduced	Offline	Python	GPU: NVIDIA RTX 3060	50 epochs training, time depends on data size
107	Normalization, patch extraction (128×128), random transformations	DAS background noise, large missing-trace data	Supervised	CNN	MCANet (multicascade attention-based)	Frobenius norm (L2 loss)	Adam	ReLU, Softmax (in ECA block)	SNR, RMS Error, SSIM	S/N: +22.19 dB (best), SSIM: 0.0036 (lowest)	Offline	MATLAB	GPU: NVIDIA RTX 3080	7.05 h training, 1.96 s inference per gather
108	Transform t-x data to frequency domain Hankel matrix, normalization, patching	Ringing noise, horizontal noise, background noise in DAS-VSP	Supervised	CNN	RRU-Net (Unet-Hankel matrices)	MSE	Adam	ReLU	SNR, waveform visual comparison	Improved SNR from −17.09 dB to 25.69 dB on synthetic data	Offline training, fast inference on large data sets	Python	GPU	25 h training, a few seconds for inference on shot gathers
109	Patching, normalization, data augmentation with noise for big variety	High-amplitude erratic, random, horizontal, fade, background	Supervised	CNN	SLKNet	MSE	Adam	ReLU	SNR, Visual comparison, FK spectra	SNR improvement: from -17.09 dB to 25.69 dB; outperforms U-net and SVD methods	Offline	Python	GPU: NVIDIA GTX 1050Ti	Training some hours; after training 11 s per test gather
110	Patching, cycle-consistency constraints, unpaired approach	Random, fading, horizontal, low frequency	Supervised	CNN	CycleGAN + attention	Adversarial loss + Cycle consistency loss	N/A	ReLU and Leaky ReLU	MSE and SNR	S/N = up to 27.29 dB, MSE = as low as 2.74e-5	Offline	Python	GPU	196 h for training
111	FFT, STFT, DWT	Gaussian, DAS measurement noise	Supervised	Hybrid (CNN + LSTM)	LSTM-CNN	Huber loss for LSTM, Categorical_ crossentropy for LSTM-CNN	SGDM (for LSTM), Adam (for LSTM-CNN)	ReLU, Softmax	Accuracy, precision, recall, F1	val_accuracy = 93.87%; validation accuracy = 93.92%	Offline	Python	GPU	42 min training
112	Down sample for the top method, sub-chirped pulses in example, patch-based training	Laser frequency drift, AWGN, stripe noise	Supervised	Hybrid (CNN+RNN(GRU)	2DSenseNet	MSE + phase-transition constraint loss	N/A	ELU	Noise level (pε/√Hz), spatial resolution (m), STFT clarity	Noise ↓59%, spatial resolution 0.9 m, demodulation linearity preserved	Offline	Python	GPU	N/A
113	Patch-based extraction, combine synthetic signals with field noise	Random, fading, horizontal, checkerboard, low-frequency	Supervised	CNN	DADN	L1 loss	Adam	ReLU, Sigmoid	SNR improvement, Visual Inspection, Grad-CAM Interpretability	DADN outperformed BP filter, FX Deconvolution, ADNet, RIDNet, HINet, MPRNet	Offline	Python	GPU	N/A
114	Random mask application, normalization, patch-based training	Random, Gaussian-like, checkerboard, fading, horizontal, low-frequency	Supervised	CNN	Triplet Masking Denoiser (Tri-MD)	Cross-Entropy Loss, Charbonnier Loss	Adversarial loss + Cycle consistency loss	N/A	SNR, MSE, LS (Local Similarity)	Superior performance in all metrics vs. baselines (SNR ↑, MSE ↓, LS ↑)	Offline	Python	GPU	500 epochs
115	Patch cropping, normalization, merging real noise	Random, fading, horizontal, Gaussian-like, abnormal optic noise	Supervised	CNN	Diffusion generative model (Mean-SDE) + NAFnet	Modified objective based on optimal trajectory	Lion optimizer	NAFnet + SimpleGate multiplicative	SNR, MSE, SSIM, Signal Resolution (SR), Spectral SNR	SNR ↑ up to 20 dB, SSIM ↓ in noise (improved), SR ↑, MSE ↓ (quantified in Table II)	Offline	Python	GPU	12 h training
116	DCT block splitting, data augmentation, patch-based training	Random, fading, abnormal optical noise, horizontal, others	Supervised	CNN	CSANet (Unet+DCT+Attention)	Charbonnier loss	AdamW	PReLU (in network), Sigmoid (in attention module)	SNR, MSE, local SNR	CSANet outperformed BPF, U-Net-T, and U-Net-F in SNR and MSE	Offline	Python	GPU	Trained 100 epochs
117	Windowing data into patches, normalization, mixing real noise	Non-uniform random, checkerboard, fading, horizontal, abnormal optical, peak noise	Supervised	Hybrid (Transformer CNN)	Urefiner	Charbonnier loss	N/A	LeakyReLU	SNR, MSE, Correlation Coefficient	SNR improved from 3.04 dB to 30.13 dB	Offline	Python	GPU	N/A

118	Normalization, patch-based training	Background, random, horizontal	Supervised	Hybrid(Transformer CNN)	Hybrid(Transformer CNN)	Charbonnier Loss	N/A	N/A	SNR, MSE, SSIM	SNR: 24.53 dB, MSE: 3.86e-5, SSIM: 0.0899 (on synthetic data)	Offline	Python	GPU	N/A
119	Band-pass filtering, median filtering	Background, cultural, environmental noise	Supervised	CNN	DASEventNet (ResNet)	Binary cross-entropy	Adam	ReLU	Accuracy, precision, recall	Accuracy 100%	Offline	Python	GPU	0.2 s per 2 s data window
120	Normalization, sliding window, stacked traces, partial manual labelling for noise	Coupled noise primarily, also random or mixed	Supervised	CNN	U-Net	MAE	Adam	ReLU	Visual inspection, Residual plots, and F-K spectrum analysis	Effective suppression of coupled noise with partial signal preservation	Offline	Python	GPU	N/A
121	Geophone-derived labelling, filtering	DAS noise, ambient noise	Supervised	CNN	U-Net & PhaseNet	N/A	N/A	ReLU in layers; final output uses Softmax	Precision, Recall, TP/FP/FN/TN classification	Precision = 92%, Recall = 92%, detection down to MW = –1.6	Offline	Python	GPU	Several seconds per minute of data
122	Synthetic data embedding, amplitude scaling	DAS background noise	Supervised	CNN	BASNet	Hybrid: Binary Cross Entropy, Structural Similarity, Intersection-over-Union	N/A	N/A	Event detection counts vs. geophone catalogue	Lower detection count than geophones but reliable for events close to fiber	Offline	Python	GPU: NVIDIA Quadro GV100	1 min per 10 s data
123	Normalization, patch extraction, separate synthetic signals and real noise	Background noise, random interference, single-trace low-frequency, etc.	Supervised	Hybrid (Transformer CNN)	SKUformer	Charbonnier	Adam	LeakyReLU, GELU	SNR, SSIM	SNR: Up to 28 dB, SSIM: Up to 0.98 (best among compared methods)	Offline	Python	GPU	N/A
124	Normalization, patch-based training with two distinct SNR ranges for teacher/student	Mixed random, background, horizontal, strong abnormal	Supervised	CNN	HKD (ResNet+ Teacher-Student)	MSE + HKD loss	Adam	ReLU	SNR, MSE, SSIM, Spectral SNR	HKD S/N: 33.24 dB, MSE: 1.36×10^{-4}, SSIM: 0.0052	Offline	Python	GPU	N/A
125	Polarization conversion, SOPAS calculation	External disturbances	Supervised	CNN	Temporal Fusion Attention Network (TFAN) (Hybrid LSTM, Attention Mechanism & TCN	Categorical Cross-Entropy	Adam	Softmax	Accuracy, precision, recall	95% accuracy for training/validation; 98% for testing	Offline	Python	GPU	Real-time capable (1 s detection time)
126	Smoothing, filtering, normalization	Background, instrumental	Supervised	CNN	QRE-net	Cross-entropy	Adam	Leaky ReLU, Softmax	Accuracy, recall, FPR	Accuracy: 88.74% (1s); Recall: 81.70%; FPR: 4.22%	Offline	Python	GPU	<0.001 s per sample
127	Filtering, normalization, segmentation	Environmental, operational noise	Supervised	GNN	GAT, GCN	N/A	N/A	N/A	Accuracy, precision, recall, F1	GAT: Acc 82%, Prec 78%, Rec 80%, F1 79%; GCN lower; CNN baseline at 78.7% acc	Offline	Python	GPU	Real-time capable
128	Patch extraction (256×256), masked blind spots for self-supervised	Random noise, large amplitude background noise	Self-Supervised	CNN	Blind spot visualization (BSV)	Empirical Risk Minimization +Standard L2 Loss	N/A	N/A	SNR, MSE, SSIM	SNR improvement up to 22 dB, good MSE/SSIM	Offline	Python	GPU: NVIDIA GTX 1060Ti	N/A
129	Patch extraction (256×256) from synthetic & field data	Random, horizontal, checkerboard, fading, optical noise	Supervised	CNN	LDAFNet	L2 Loss	N/A	N/A	SNR, Signal-to-Noise Improvement Ratio (SNIR)	S/N improved from -2.9385 dB to 22.8245 dB (Total SNIR: 25.763 dB)	Offline	Python	GPU: NVIDIA GTX 1060Ti	N/A
130	Patch extraction (128×128), normalization	Random noise, horizontal noise	Supervised	CNN	GDONet	Charbonnier loss (variation of L1/L2)	N/A	PReLU	SNR, MSE	Up to 25.76 dB SNR improvement	Offline	Python	GPU: NVIDIA GTX 1050Ti	N/A

131	Patch extraction (64×64), normalization, synthetic + scaled noise superposition	Various background noise: random, horizontal, optical, time-varying, coupling noise	Supervised	CNN	MSAACNN	Frobenius norm +L2 loss	Adam	ReLU and Sigmoid	SNR, RMSE	SNR improvement >18 dB, lowest RMSE vs. other methods	Offline	MATLAB	GPU: NVIDIA RTX 3060	N/A
132	Patch extraction (128×128 or 256×256), normalization, overlap with Hann windows maybe, real noise addition	Various random, horizontal, erratic, strong background noise	Supervised	CNN	DCRCDNet	MSE	Adam	Leaky ReLU, Linear	SNR, Local Similarity (qualitative)	Improved SNR from −10.21 dB to 15.61 dB	Offline	Python	GPU	N/A
133	Normalization, sliding-window patch extraction, random weighting of noise levels	DAS background noise	Supervised	CNN	RDA-Net	MSE +L2 norm	Adam	N/A	SNR, RMSE	RDA-Net achieved SNR improvement up to 24 dB and lowest RMSE among benchmarks	Offline	MATLAB	GPU: NVIDIA RTX 3070	Training 5.08 hours, 1 second per record for inference
134	For supervised: create semi-synthetic data by adding real noise to synthetic wavefields; for N2N: separate down going/upgoing parts, align channels, normalizing, patching	Mainly DAS instrumental (hardware) noise	Supervised	CNN	Noise2Noise (UNet)	MSE	Adam	LeakyReLU	SNR	SNR improved (e.g., from 2.10 to 9.55 for SL, to 10.89 for N2N on test data)	Offline	Python	GPU	Training 250 epochs for each approach
135	Synthetic forward modelling for label data, artificially remove consecutive traces to form missing data, patching 64×64 with overlap	Strong background noise + large missing traces	Supervised	Transformer	seismic interpolation transformer (SIT)	SSIM-based loss	N/A	Leaky ReLU	Q value, SSIM, Local Similarity	Q ≈ 30 dB, High SSIM	Offline	Python	GPU: NVIDIA RTX 4090	Training 56 hours; inference a few seconds
136	Rotations for handling horizontal/vertical correlated noise, masked neighbourhood creation, plus TTV weighting for local texture.	Spatially correlated noise and random noise	Self-Supervised	CNN	U-Net style with masked-spot modifications	Custom loss with Texture Total Variation (TTV) regularization	N/A	Leaky ReLU	SNR, SSIM, MSE, local similarity	SelfTexture outperforms BM3D, BP, DIP, DDAE, BSN	Offline	Python	GPU: NVIDIA RTX 4090	N/A
137	Possibly normalization, no strong mention of others	Random noise, fading noise, low-frequency noise, horizontal noise	Semi Supervised	CNN	Semi-supervised denoising network (SSDN)	Supervised: L1 Loss, Unsupervised: KL Divergence Loss	N/A	LeakyReLU	Global and Local SNR	SSDN outperforms traditional, supervised, and unsupervised baselines	Offline	Python	GPU: NVIDIA GTX 3060	Epochs up to 100
138	Optional clipping and bandpass on the "removed noise" section	Various random noise also recovers leaked seismic signals from the "removed noise"	Unsupervised	Dictionary learning and sparse coding	SigRecover	N/A	N/A	N/A	Qualitative visual inspection, energy fidelity	Qualitative improvement in signal fidelity shown through figures	Offline	Python	N/A	N/A
139	FFT, filtering, normalization	Background noise	Supervised	SNN	SNN	N/A	N/A	N/A	Accuracy, precision, recall, F1	SNN: Accuracy 81.1%, F1-score 80%; CNN: Accuracy 78.7%, F1-score 77.8%	Offline	Python	GPU	Real-time capable
140	Possibly bandpass or normalization, not clearly detailed	Coupling noise, fading noise, checkerboard noise, background noise	Supervised	CNN	Attention-Based Deep CNN (SPANet)	L1 Loss	Stochastic Gradient Descent (SGD) with cosine annealing	LeakyReLU	SNR, MSE, Difference Map, f-k Spectrum	S/N up to 24.14 dB, MSE as low as 1.2e-3	Offline	N/A	N/A	N/A
141	FFT, filtering, STFT	Background noise	Supervised	CNN	DSRNet	Cross-entropy	Adam	ReLU, Softmax	Accuracy, Precision, F1 Score, FPR, Inference Time	Accuracy (5s: 88.29%, 10s: 93.15%, 20s: 93.88%), F1 Score (up to 91.5%)	Offline	Python	GPU	<0.5 s per sample
142	Yes, some bandpass and time-windowing before CITD	Optical noise, fading noise, background noise	Supervised	Hybrid (decomposition + ensemble classification)	Random forest / GBDT-based classification plus improved ITD	N/A	N/A	N/A	Accuracy, Precision, Recall, F1-score, SNR, RMSE	>99% accuracy, ~10–14 dB SNR improvement	Offline	N/A	N/A	N/A

143	Bandpass filtering, normalization	Anthropogenic, instrumental	Transfer Learning	Convolutional Transformer	EQCCT	MAE	Adam	N/A	Precision, Recall, F1-score, MAE, Standard Deviation	Precision: 100%, Recall: P-wave 91.5%, S-wave 88.48%, F1-score: 94.73%, MAE (P): 0.009 s, MAE (S): 0.096s	Offline	Python	GPU	Real-time capable
144	Filtering, normalization	Background noise, operational	Transfer Learning	CNN	ResUNet with EQTransformer	N/A	N/A	ReLU	MAE, STD, accuracy within error thresholds	MAE = 0.15s, STD = 0.36s, Accuracy = 97.9% (for P-phase)	Offline	Python	GPU	Real-time capable
145	Possibly bandpass filtering and time-window segmentation for building training patches.	Ocean waves, random noise, background anomalies, ignoring high-freq content.	Supervised	LSTM (SHRED (Shallow Recurrent Decoder), INR (SIREN, RFFN))	SHallow REcurrent Decoder (SHRED)	MSE	Adam	ReLU (SHRED, RFFN), Sinusoidal (SIREN)	MSE, PSNR	PSNR good for ocean waves; poor for >0.15 Hz seismic content	Offline	Python	GPU: NVIDIA A100	Training about tens of minutes for a 10 min dataset example
146	Possibly normalization, no strong mention.	Coupling noise, fading noise, random background noise, horizontal noise, multiple random types in downhole environment	Supervised	CNN	Attention-Based Deep CNN (NSKMRNet)	Charbonnier loss	N/A	Leaky ReLU	SNR, MSE	SNR improved 30 dB; MSE lowest compared to baselines	Offline	N/A	GPU	N/A
147	Bandpass filter for removing out-of-range frequencies, optional downsampling to reduce sample rate, then patch generation for the deep net	Random/erratic noise, background anomaly, some high amplitude outliers.	Self-Supervised	Transformer + MLP	Transformer-based autoencoder	MSE/PSN	N/A	ELU	PSNR, SSIM, RMSE	PSNR: up to 60.45 dB, SSIM: 1.00, RMSE: 50–90	Offline	Python	GPU	Training 500 epochs with patches, around half an hour for large examples
148	Possibly minor amplitude scaling or patch chunking for dictionary learning.	Complex mixture of random and erratic noise, large amplitude outliers, multi-type noise	Unsupervised	Dictionary Learning	via SVD-Free	Huber-based robust norm with iterative	Custom iterative update rules	N/A	SNR	Improvement from 15.1 dB to over 10 dB (synthetic)	Offline	N/A	N/A	Higher than simpler methods due to iterative dictionary approach
149	Overlapping patch extraction (size 32×32; 2-sample shift) and 1-D vectorisation	Additive white random noise; realistic coherent noise	Unsupervised	Variational Autoencoder (VAE) + Squeeze-and-Excitation (SE) Network	Variational Autoencoder (VAE) + Squeeze-and-Excitation (SE) Network	N/A	Adam	ReLU, Sigmoid, Linear	SNR, Local Similarity Map, Imaging accuracy via Reverse Time Migration (RTM)	SNR improved from -32.45 dB to 0.54 dB;	Offline	N/A	CPU	N/A