



CNN-powered malfunction detection in industrial machine sounds with fair data splitting and dynamic training control

Ameer B. A. Alaasam^{a,c}, Zena H. Khalil^b, Dhiah Al-Shammary^b, Ayman Ibaida^{d,*}, Khandakar Ahmed^d

^a School of Electronic Engineering and Computer Science, South Ural State University, Chelyabinsk, 454080, Russia

^b College of Computer Science and Information Technology, University of Al-Qadisiyah, Al Diwaniyah, 58001, Iraq

^c Directorate for Education in Al Qadisiyah under the Ministry of Education of Iraq, Al Diwaniyah, 58001, Iraq

^d Intelligent Technology Innovation Lab, Victoria University, Melbourne, 3011, Australia

ARTICLE INFO

Publishing editor: Xin Huang

Keywords:

Convolutional neural network
Early stopping
Imbalanced data classification
Industrial machine sound
Learning rate scheduling
Malfunction detection

ABSTRACT

In an industrial environment, sound is generally an important source of data that results from the movement and vibration of machines during operation. Each machine operation condition often generates a specific sound, for example, a familiar normal working sound, an unfamiliar sound, a breaking sound, or even the absence of sound, which may indicate a stop. This paper proposes a supervised convolutional neural network (CNN) model with early stopping and learning rate (LR) scheduling to achieve a more accurate and cost-effective trained deep learning (DL) model for machine sound classification. The proposed model is trained and evaluated using the MIMII dataset, which contains unbalanced real-world data and signal-to-noise ratios (SNRs). To ensure a fair (i.e., representative) data splitting strategy and robust coverage across all hierarchical structures of the MIMII dataset, this study adopts a hierarchical multi-stage stratified splitting protocol incorporating pre-split shuffling to prevent order bias. The synthetic minority oversampling technique (SMOTE) is used to balance the training set after feature extraction. Compared with state-of-the-art works implemented on the same dataset, the proposed model achieves optimal performance in terms of accuracy, area under the curve (AUC), F1-score, precision, recall, Matthews correlation coefficient (MCC), and Kappa without noise processing methods at an SNR of 6 dB, and maintains robust, competitive performance under severe noise conditions (0 dB and -6 dB).

* Corresponding author.

E-mail addresses: alaasamab@susu.ru (A.B.A. Alaasam), zena.khalil@qu.edu.iq (Z.H. Khalil), d.alshammary@qu.edu.iq (D. Al-Shammary), ayman.ibaida@vu.edu.au (A. Ibaida), khandakar.ahmed@vu.edu.au (K. Ahmed).

<https://doi.org/10.1016/j.jnlest.2026.100372>

Received 4 November 2025; Received in revised form 23 May 2026; Accepted 13 July 2026; Available online 31 August 2026.

1674-862X/© 2026 The Authors. Publishing Services provided by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1 Introduction

In industrial environments, sound is often an indicator of variation in the operational state of machinery. Machines create acoustic signatures by vibrating, rotating, and mechanically interacting, thereby indicating their health and performance. In addition, some machines may run normally with internal problems that may lead to future downtime or inefficiency. Therefore, sound could be critically important as it may provide information about the internal state, for example, physical states that are not visible or cannot be sensed or captured using traditional sensors. Analytically, such auditory signals can reveal possible malfunctions, mechanical wear, deviations from the standard operation mode, or a potential failure early enough to perform proactive maintenance and increase the reliability of the system.

Unlike image classification, where raw images of equal dimensions can be fed directly to convolutional neural network (CNN), using CNN for time-based signals, such as audio, classification presents practical challenges. For example, the length and other features of time-domain signals may vary, which may cause CNN performance degradation in terms of classification accuracy [1–3]. In addition, despite various studies that have successfully built deep learning (DL) models such as deep CNN [4] and deep recurrent neural network (RNN) [5] for audio processing tasks, DL models generally tend to face overfitting problems during training and fail to produce the same accuracy on unseen data [6]. Models such as CNN may require enhancement techniques to make them more robust under real-world conditions [7].

The overfitting problem is amplified when the data are noisy or imbalanced, where models learn too specific patterns instead of learning more general ones. In addition, training machine learning (ML) and DL models, in particular, complex neural networks (NNs) on large amounts and complex data such as audio, require considerable computational power and extended training times. These challenges significantly increase the costs in terms of energy consumption, infrastructure, and delays in the deployment of intelligent systems. Such problems become more serious considering the importance of real-time operational efficiency and decision-making in the context of smart industries, such as industrial Internet of things (IoT) systems. IoT comprises cyber-physical assets, networked objects, information technologies, and fog computing platforms to minimize processing time in industrial environments for an effective response to automation and process optimization [8].

However, while DL models have been successfully applied in many applications, challenges involve complex interactions, making the training of DL models a multi-objective optimization problem. Examples of conflicts and trade-offs are as follows: A long training time by increasing the epochs can lead to overfitting while reducing the number of epochs could lead to underfitting. Handling unbalanced data techniques, such as synthetic minority oversampling technique (SMOTE) (oversampling), increases data size, and hence training time; underfitting may also occur when simpler models are used to save training time. Several strategies have been proposed to address these challenges. For example, although high efficiency can be achieved by distributed training and parallelization on supercomputers and graphics processing unit (GPU) clusters, such infrastructure may not be accessible owing to economic or technical constraints, such as edge devices or local servers. In such cases, other strategies are applied to improve the efficiency of the model.

Some methods, such as early stopping, learning rate (LR) scheduling, and SMOTE, can effectively solve these problems [9,10]. Early stopping involves monitoring the training and stopping it when the performance starts to deteriorate to mitigate overfitting and reduce training time. There are traditional early stopping methods, such as stopping after the validation error increases or after no improvement over “ n ” epochs. Additionally, LR scheduling changes the LR during training to achieve a trade-off between speed and stability (e.g., by decreasing the LR as training proceeds, which can help the model arrive at a more desirable solution). On the other hand, SMOTE creates as many synthetic examples for the minority class as is required to handle class imbalance problems.

The other method is batch normalization (BN), which is the process of making NN faster and more stable by adding extra intermediate layers to the deep neural network (DNN) and normalizing activations in the intermediate layers. It normalizes the input features within a mini-batch to have unit standard deviation and zero mean, and then uses those normalized features to compute later features, scaling and shifting them by learned parameters. BN helps reduce internal covariate shift, improves the gradient flow, and allows higher LRs, making training faster and more stable [11].

Additionally, the two-dimensional max pooling (MaxPooling2D) method is a downsampling operation often used in CNNs to reduce the spatial dimensions of feature maps produced by convolution layers, helping

to retain only the important features as much as possible. In fact, MaxPooling2D discards non-maximum values and, as a result, can improve DL model performance [12].

Another useful regularization technique is dropout, which prevents overfitting by randomly deactivating a subset of neurons during the training process. This could push the network to learn more generalized features and potentially become more robust to unseen data.

Nevertheless, the fundamental phase for utilizing ML and DL methods is the pre-processing of audio signals. Feature extraction is a crucial method in audio preprocessing in which raw audio is converted into representations that carry important signal patterns. Each audio feature reflects different aspects of raw audio. ML and DL models may use more than one feature as input because different features encode complementary aspects of the audio. These features include the Log-Mel spectrogram (LM) and Mel-frequency spectral coefficients (MFCC), both of which rely on the short-time Fourier transform (STFT) and a Mel filter bank to approximate human auditory perception [13,14]. Spectral centroid (SC) [15] which is another audio feature reflects the center of the “gravity” of the spectrum of the audio frame. SC also relies on computing the discrete Fourier transform (DFT) of each audio frame to obtain the DFT coefficients. An additional audio feature is the spectral flatness (SF), which is used to evaluate the flatness characteristics of a noise-shaped spectrum. The SF factor results from comparing the geometric mean to the arithmetic mean of the power spectrum magnitudes obtained from STFT calculations [15,16]. Spectral roll-off (SR) is another audio feature [17] that usually represents the frequency below which a specified percentage (usually around 90%) of the magnitude distribution of the spectrum is concentrated. Spectral bandwidth (SB) [17] (also known as spectral spread) quantifies the spread of frequencies around the SC, representing the overall “width” or “spread” of a sound’s frequency content. Another interesting feature is the zero-crossing rate (ZCR), which represents the number of signal polarity changes [17].

In the smart industry, many researchers have employed ML and DL techniques, in conjunction with various preprocessing methods and audio features, to develop effective malfunction detection systems. For example, Ref. [18] proposed a supervised learning approach using MFCCs, Chroma, SC, SB, SR, and ZCR as features. The authors oversampled the abnormal samples and applied standard scaling. They split the dataset into training (49%), validation (21%), and testing (40%). Four supervised models were used: support vector machine (SVM), random forest (RF), Naive Bayes, and multilayer perceptron (MLP) with three hidden layers, Adam optimization, and binary cross-entropy for loss. The authors used separate models trained for each machine type rather than a unified model. For the evaluation, the F1-score and area under the curve (AUC) were used for benchmarking against a baseline autoencoder on Mel-spectrograms. The model achieved a 39.5% improvement in the AUC compared with the baseline.

Ref. [19] introduced a supervised learning method using a super-learner ensemble. Other datasets, such as ESC-50, were also used. The proposed method applied five base learners linked by a meta-learner: AdaBoost, Naive Bayes, K -nearest neighbors, SVM, and RF. Spectral feature extractions of MFCC, SC, SB, SR, and ZCR were applied along with SMOTE to reduce class imbalance. The dataset was split into 70% for training and 30% for testing using fold cross-validation ($K = 3$). SVM achieved the best testing performance for valves at SNR = 6 dB with an AUC of more than 0.9, whereas the worst performance occurred with the super learner on valves at SNR = -6 dB with an AUC of approximately 0.6. All four machine types were tested at various noise levels.

In addition, Ref. [20] proposed a supervised DL approach called weighted ensemble using EfficientNet-B0, B5, and B7 models to detect industrial machine malfunctions using the MIMII dataset. The authors evaluated only the pumps and valves because of their diagnostic complexity. Significant augmentation (e.g., pitch change, speed change, and noise addition) was used for data preprocessing to address data imbalance. Ninety percent of the data were used for training and 10% for testing. The training set was partitioned into 80:20 for training and validation. The authors evaluated the performance of the model using multiple metrics and achieved an accuracy of 98.211% with an AUC of 0.9993 for valves, an accuracy of 97.447%, and an AUC of 0.9981 for pumps. All chosen machine types (pumps and valves) were thoroughly tested. Although the pump and valve were the only components selected by the authors, it took nine days to train and approximately 12 min to test the process.

Additionally, Ref. [21] proposed a supervised fan fault detection method based on CNN-GRU models trained with the MIMII dataset with three noise levels (-6 dB, 0 dB, and 6 dB). The authors presented a

CNN-GRU fusion model, where the CNN provided spatial features from frequency domain features and the GRU took temporal features from the generated enhanced logarithmic filter bank (Log-Fbank) spectrograms. The data were divided into two balanced datasets: i) a noise-free sample and ii) a sample with proportionally equal amounts of noise at -6 dB, 0 dB, and 6 dB. Both datasets were split into training (70%) and testing (30%) datasets. Log-Fbank features were extracted and improved using histogram equalization to remove noise, and the input dimensions were normalized. On the test set of the mixed noise, the CNN-GRU model had the best accuracy (97.8%), which was better than the baseline CNN (92.8%) and visual geometry group 16-layer network (VGG16) (94.3%). In the low-noise sample, the authors observed signs of overfitting, as they achieved 95% training accuracy and 85% testing accuracy.

Ref. [22] evaluated three supervised learning approaches: binary, partially mixed, and completely mixed. In the binary approach, the number of models was the product of the machine types, machine number, and SNR levels, totaling 48 for the MIMII dataset. The partially mixed approach reduced this to three models corresponding to the SNR levels, each identifying both the equipment type and its status. In this case, the number of models equaled the number of SNR levels, while the number of classes corresponded to the product of the equipment types and their possible statuses. The fully mixed approach required only a single model with eight classifiers. The authors considered five algorithms: extreme tree classifier, linear discriminant analysis, gradient boosting (GB), light gradient boosting machine (LGBM), and SVM. In the preprocessing part, the extracted features included the MFCC, SC, bandwidth, roll-off, ZCR, and the SMOTE data balancing technique, as well as further Z-score normalization. Data splitting was performed at a ratio of 60:40 between the training and testing sets. The highest AUC-ROC was 0.93 for valves in the totally mixed model case, and the lowest value AUC-ROC was about 0.55 for valves in the binary case. First, all the machines (pump, fan, slide rail, and valve) were checked for all their parameters. ROC stands for receiver operating characteristic.

Therefore, this work seeks further improvements in the accuracy of fault detection in industrial machines, handling overfitting, and increasing the speed of deployment by reducing computational costs. This paper presents a new CNN-based DL model for the detection of malfunctions using the sounds of industrial machines. After feature extraction, SMOTE was used to balance the training set. Although recurrent architectures such as RNNs, LSTMs, and GRUs have been widely used for audio sequence modeling, this study adopts a CNN-based architecture because of its superior efficiency in learning localized time-frequency patterns from 2D acoustic feature representations. CNNs offer faster convergence, lower computational overhead, and improved robustness to localized noise distortions compared with sequential models, which are often more sensitive to noise accumulation across time steps. Moreover, CNNs integrate early stopping and dynamic LR scheduling, supporting the objective of reducing training costs and overfitting while maintaining high classification performance in industrial deployments, and the model was trained and tested on a well-known MIMII dataset [12], version “public 1.0”. MIMII is an imbalanced real-world dataset for industrial machine fault detection that features stationary and nonstationary records from four types of machines: valves, pumps, fans, and slide rails. Each machine includes a sub-dataset for different SNRs: -6 dB, 0 dB, and 6 dB. Each sub-dataset with a specific SNR includes four individual product models, which refer to each product model as a “device”, each of which contains normal and abnormal sounds, including issues such as leakage, contamination, rail damage, and unbalanced rotation. The datasets also include background noise from real factories. This dataset was specifically collected for tasks such as anomaly detection, transfer learning, and noise robustness testing. The results of the experiment showed that without specific methods to handle noise, the proposed model achieved optimal performance when the SNR was 6 dB and high performance in other SNRs compared with the state-of-the-art work implemented on the same dataset. In addition, a fair data-splitting approach was adopted to maintain the same imbalance ratio among the training, validation, and testing datasets, and a fair representation of each device in each phase.

2 Proposed approach

2.1 Fair per device/state data splitting

In this work, to preserve fair (i.e., representative) samples from each device and its states in training, validation, and testing for more reliable performance estimates, this work implements a hierarchical

adaptation of a multi-stage stratified splitting protocol where “hierarchical” refers to nested organization of data (SNR → machine type → device ID → state/label), and “multi-stage” reflects that splitting is carried out independently within each device/state group before submitting the final training, validation, and test sets. Shuffling was implemented before splitting to prevent order bias. For instance, considering a pump device operating at 6 dB SNR with 3749 normal sound files, this work performs splitting for each device separately, based on 70:15:15. The final sum of the results was 2623 training samples (approximately 69.96%), 560 validation samples (approximately 14.93%), and 566 test samples (approximately 15.10%). In such cases, the results are close to the ideal 70:15:15, because when splitting, rounding the integer for training and then for validation may leave a bit more samples in the testing set.

Table 1 provides an example of pump-splitting results.

Table 1

Data splitting example on pump.

SNR (dB)	ID	State	Total	Train	Validate	Test
6	ID 00	Normal	1006	704	150	152
		Abnormal	143	100	21	22
	ID 02	Normal	1005	703	150	152
		Abnormal	111	77	16	18
	ID 04	Normal	702	491	105	106
		Abnormal	100	70	15	15
	ID 06	Normal	1036	725	155	156
		Abnormal	102	71	15	16
	Sum normal		3749	2623	560	566
	Sum abnormal		456	318	67	71

To evaluate the adopted data-splitting protocol, we compare it with a stratified baseline method with similar settings, including split ratio and hierarchical structure, but the baseline uses the class label (state) without enforcing per-device in machine type. To assess fairness, the maximum deviation measures were considered because they expose the largest discrepancy produced during the splitting process. This is particularly important in industrial settings, where strict guarantees are required to ensure that each split is as much as possible faithful and representative miniature of the original. Therefore, to quantify the effect on device representativeness, the change in device composition within each class before and after splitting was calculated as follows:

i) For each class $y \in \{\text{normal, abnormal}\}$ and device d , the device fraction in the full set $p_{\text{all}}(d|y)$ and in each split $p_s(d|y)$ for $s \in \{\text{train, val, test}\}$ was calculated. For example, in the (6 dB, pump) subset in the adopted splitting protocol, as shown in Table 1, $p_{\text{all}}(\text{id_00}|\text{normal}) = 1006/3749 \approx 0.268$; and in the training split $p_{\text{train}}(\text{id_00}|\text{normal}) = 704/2623 \approx 0.268$ for other splits and states.

ii) Then report the maximum deviation as $\Delta(d, y) = \max_s |p_s(d|y) - p_{\text{all}}(d|y)|$. A lower Δ indicates that the device coverage is preserved more faithfully across the splits.

The results showed that the adopted protocol yielded smaller deviations in device representativeness in the selected (6 dB, pump) subset (normal: $\max \Delta = 0.000722$; abnormal: $\max \Delta = 0.010100$), while the stratified baseline showed larger device-level drift (normal: $\max \Delta = 0.040849$; abnormal: $\max \Delta = 0.102576$).

2.2 General workflow

The proposed workflow based on the MIMII sound dataset is shown in Fig. 1. The original dataset includes three levels of SNR (−6 dB, 0 dB, and 6 dB) for different types of machines (pump, fan, valve, and slide rail). This study considered all the data for each machine type under all SNRs and the presence of normal and abnormal samples. For each machine, there is a model trained on the data for that machine at all SNR levels. Four models were used in the results. The training and validation sets were first subjected to feature extraction and normalization, and then SMOTE was applied to balance the training set only. The trained and validated models were evaluated using a preprocessed testing set. The workflow was applied to all machine types.

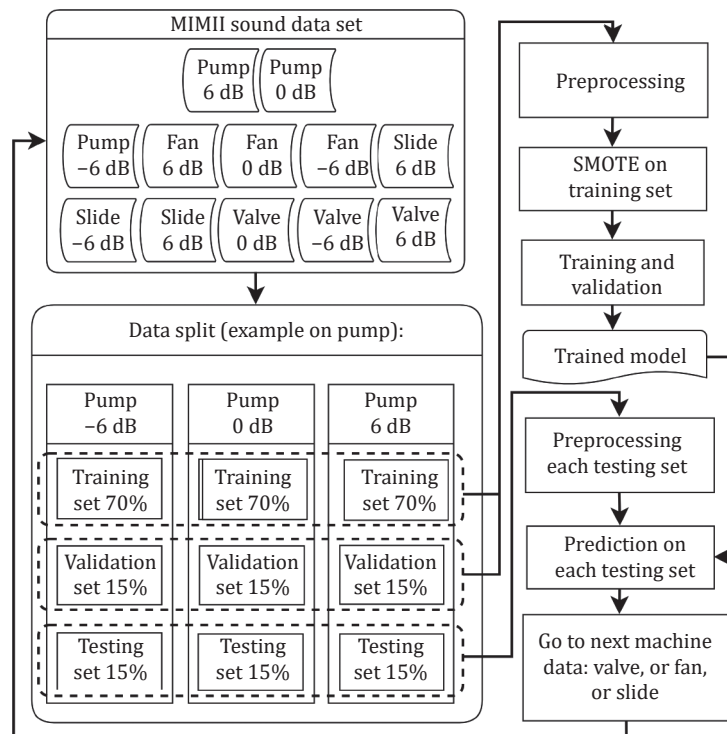


Fig. 1. General workflow.

2.3 Preprocessing

Each waveform audio file format (WAV) file transformed into a NumPy array in Python represented a signal in the time domain. The feature extraction steps were explained where the multichannel audio was averaged to mono as the input and then extracts a broad range of features.

The extracted features were concatenated into a unified feature matrix, which was reshaped for DL architectures. All the features were extracted using the Librosa library (Version 0.10.2) in Python at a sampling rate of 16 kHz. We used an FFT size (n_{FFT}) of 2048, with 128 Mel filters and 40 MFCCs. LMs were computed using `power_to_db (ref = np.max)`, SR used `roll_percent = 0.85` (the percentage threshold used to compute the spectral roll-off frequency), Hann window, `win_length = 2048` (the number of audio samples included in each analysis window during signal processing), and `hop_length = 512` (the number of samples between consecutive analysis windows in audio processing, `win_length/4`). Other details are presented in Fig. 2. The training and validation feature sets were then standardized using `StandardScaler` from the Scikit-learn library (Version 1.5.2) in Python, ensuring a mean of zero and standard deviation of 1 for uniform scaling.

Subsequently, SMOTE was imported from the `imblearn.over_sampling` library and configured with (`random_state = 42`, `sampling_strategy = 'auto'`, and `k_neighbors = 5`) and was applied only to the training sets to balance the data before model training. Note that when setting strategy to 'auto', the algorithm identifies all classes except the majority class and oversamples them until they contain the same number of samples as the majority class. SMOTE operates in the feature space using K -nearest neighbors, where for each minority sample in the training set, a neighbor is selected and a new sample is created by linear interpolation between them. Table 2 shows an example of class distribution before and after SMOTE.

Note that the Librosa library (Version 0.10.2) preserves the core mathematical concepts of audio features (e.g., SC) while adjusting output formats and implementation details, such as outputting SC in Hz rather than bin indices as detailed in its documentation [37], enhancing computational efficiency and DL compatibility.

Require: Input audio signal x (NumPy array), sampling rate f_s (default 16 kHz),
number of MFCC coefficients n_{MFCC} (40 used), FFT window size n_{FFT} (2048
used), number of Mel bands n_{Mel} (128 used)
Ensure: Feature matrix F
1: **if** x is multi-channel **then**
2: $x \leftarrow \text{mean}(x)$
3: **end if**
4: $\text{MFCC} \leftarrow \text{mfcc}(x, f_s, n_{\text{MFCC}}, n_{\text{FFT}})$
5: $\text{MelSpec} \leftarrow \text{melspectrogram}(x, f_s, n_{\text{FFT}}, n_{\text{Mel}})$
6: $\text{LM} \leftarrow \text{power_to_dB}(\text{MelSpec}, \text{ref} = \max(\text{MelSpec}))$
7: $\text{SC} \leftarrow \text{spectral_centroid}(x, f_s, n_{\text{FFT}})$
8: $\text{SR} \leftarrow \text{spectral_rolloff}(x, f_s, n_{\text{FFT}}, 0.85)$
9: $\text{SF} \leftarrow \text{spectral_flatness}(x, n_{\text{FFT}})$
10: $\text{SB} \leftarrow \text{spectral_bandwidth}(x, f_s, n_{\text{FFT}})$
11: $\text{ZCR} \leftarrow \text{zero_crossing_rate}(x)$
12: $F \leftarrow \begin{bmatrix} \text{MFCC} \\ \text{LM} \\ \text{SC} \\ \text{SR} \\ \text{SF} \\ \text{SB} \\ \text{ZCR} \end{bmatrix}$ ▷ Vertical concatenation
13: $F \leftarrow \text{reshape}(F)$
14: **return** F

Fig. 2. Preprocessing.**Table 2**

Class distribution of training data before and after SMOTE.

Machine	Training dataset	Normal	Abnormal	Total samples	Synthetic samples generated
Valve	Before SMOTE	7746	1005	8751	/
	After SMOTE	7746	7746	15492	6741
Pump	Before SMOTE	7864	954	8818	/
	After SMOTE	7864	7864	15728	6910
Slide	Before SMOTE	6720	1863	8583	/
	After SMOTE	6720	6720	13440	4857
Fan	Before SMOTE	8553	3090	11643	/
	After SMOTE	8553	8553	17106	5463

2.4 Proposed CNN model architecture

The architecture of the proposed model is illustrated in Fig. 3, where the output shapes (none, 85, 155, 32) are (batch, height, width, and channels). The model follows a sequential architecture consisting of three convolutional blocks, where the number of filters ranges from 32 to 128. Each convolutional block contains further Conv2D layers using a 3×3 kernel and ReLU activations assisted by BN, which is used to stabilize and speed up training. Downsampling features were achieved by applying MaxPooling2D layers, and dropout was used to overcome overfitting, where the dropout rate begins at 0.3 and increases in the last block. The extracted features were passed through fully connected layers. The data were reshaped for dense processing by the flattened layer, followed by a dense layer of 256 units with ReLU activations, BN, and a dropout rate of 0.5. Such a configuration is used to ensure a balance between deep feature learning and reduce overfitting. Finally, the dense layer consists of a single unit and sigmoid activation, which outputs a probability score for the binary classification. The model was compiled with the Adam optimizer chosen for its capacity for adaptive LR and binary cross-entropy loss for binary classification purposes. The batch size was determined to be 32 when the compiled model was used for training. During training, metrics such as accuracy and AUC were used to evaluate performance.

2.5 Dynamic training criteria

The proposed training strategy combines EarlyStopping and ReduceLROnPlateau callbacks imported from Tensorflow. keras. callbacks to enhance the performance. The validation loss triggers the training termination with the restoration of the best weight when it shows no improvement over seven validation epochs (patience = 7). Smaller patience values lead to premature convergence, whereas larger values do not yield measurable performance gains. Using these methods results in both the prevention of overfitting and a reduction in computation costs. The ReduceLROnPlateau callback uses validation loss trends to automatically determine changes in LR. The LR decreases by 0.2 factor during three sequential epochs with no reduction in loss to improve weight updates. The threshold value for the minimum LR was 1×10^{-5} to avoid ineffective adjustments. These parameters were selected empirically through preliminary experiments to balance the convergence stability and training efficiency. Smaller patience values (≤ 3) led to premature termination before validation convergence, particularly at lower SNR levels, whereas larger values (> 10) did not yield measurable performance gains but increased training time. Similarly, reducing LR by a factor of 0.2 provided a stable trade-off between convergence speed and fine-grained optimization; more aggressive reductions slowed convergence, whereas smaller reductions had a negligible effect.

3 Experiments

3.1 Setup

The experiments were implemented using Python Version 3.10.12, on a single node equipped with a 12th Generation Core i7-12700H processor, 16 GB DDR5 RAM, and an NVIDIA GeForce RTX 3060 GPU with 6 GB memory. Training was implemented on a GPU using TensorFlow Version 2.17.0. The dataset was the MIMII dataset for malfunctioning detection in the sounds of industrial machines. All machine types (valves, pumps, fans, and slide rails) with all their devices and under all SNRs available at the time of writing this paper were considered. For each machine type, a separate model was trained on all SNR values, including all machine model IDs available in the dataset for this machine.

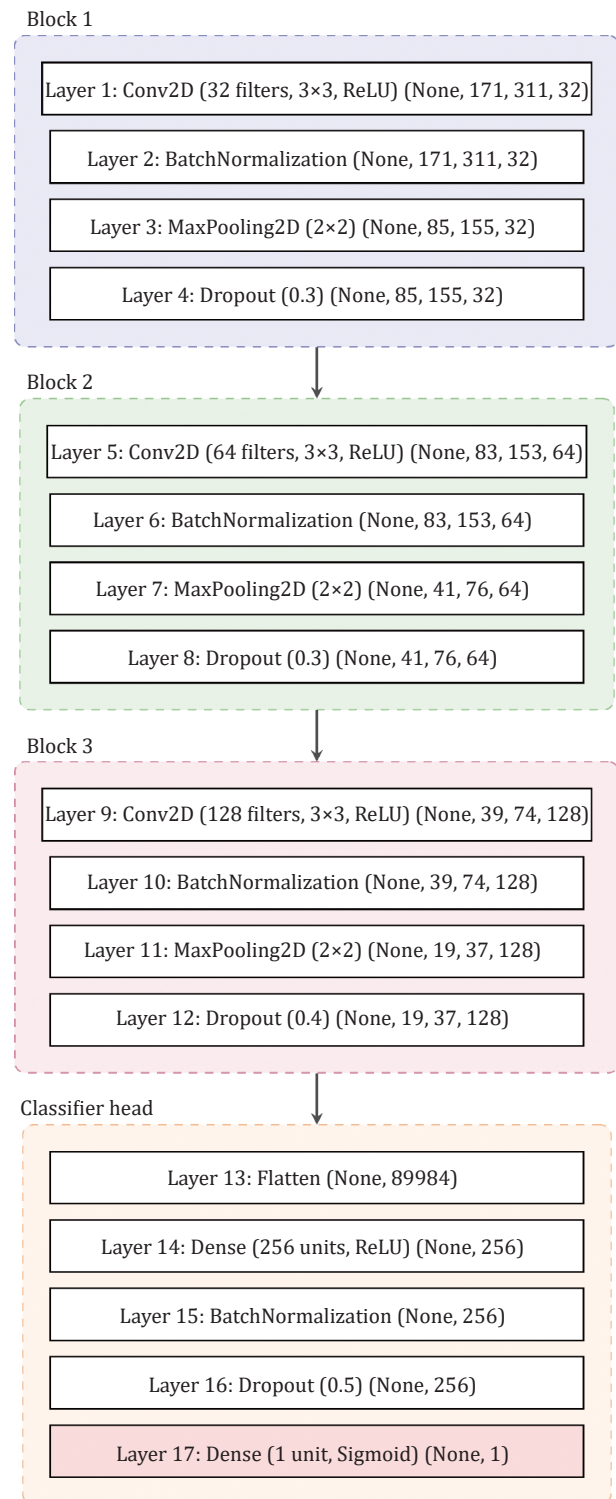


Fig. 3. Model architecture in the proposed approach.

3.2 Results

This study used accuracy, AUC, F1-score, precision, recall, MCC, and Kappa as base evaluation metrics. In addition, Fig. 4 shows the confusion matrices obtained in the present experiment with the dynamic criteria for all machines and all SNR levels in the dataset. Table 3 shows that the number of epochs used during training (best and all) varied owing to the callbacks used (EarlyStopping and ReduceLROnPlateau).

Table 3
Number of epochs in each run.

Machine	1st run		2nd run		3rd run	
	Best	All	Best	All	Best	All
Valve	20	27	12	19	6	13
Pump	11	18	19	26	10	17
Fan	19	26	15	22	10	17
Slide	11	18	19	26	12	19

While the initial setup was configured to run at 100 epochs, subsequent training runs required considerably fewer epochs, indicating better efficiency and lower computational costs. For example, the largest number of epochs the valve machine trained for in the 1st run was up to 27 epochs and 13 epochs in the 3rd run. Additionally, regarding the time cost, Table 4 reports the GPU training times with and without dynamic criteria for pump and fan as an example, where enabling the dynamic criteria reduced the training time to 10.94 min (18 epochs), saving 44.68 min (~80.3%), with a similar gain in time with fan saving of 45.8 min (~74.69%).

Table 4
GPU training time and executed epochs for pump and fan 1st run, comparing dynamic criteria training against a fixed epoch training without dynamic criteria under the same hardware and experimental setup (combined SNR: 6 dB, 0 dB, and -6 dB).

Machine	Dynamic criteria	Planned epochs	Executed epochs	Training time (min)	Time saved (min)	Cost reduction (%)
Pump	Yes	100	18	~10.94	~44.68	~80.30
	No	100	100	~55.62	/	/
Fan	Yes	100	26	~15.52	~45.80	~74.69
	No	100	100	~61.32	/	/

Table 5 shows the accuracy, AUC, F1-score, recall, precision, MCC, and Kappa resulting from this

		Predicted	
		Normal	Abnormal
True	Normal	TN	FP
	Abnormal	FN	TP

SNR	Machine	1st run		2nd run		3rd run	
6	Valve	557	0	557	0	557	0
		0	73	0	73	0	73
	Pump	566	0	566	0	566	0
		0	71	0	71	0	71
	Fan	615	0	615	0	615	0
		0	225	0	225	0	225
	Slide	484	0	484	0	484	0
		0	137	0	137	0	137

SNR	Machine	1st run		2nd run		3rd run	
-6	Valve	554	3	553	4	553	4
		0	73	0	73	0	73
	Pump	565	1	561	5	562	4
		4	67	4	67	6	65
	Fan	607	8	605	10	606	9
		20	205	17	208	21	204
	Slide	482	2	482	2	482	2
		3	134	3	134	3	134

SNR	Machine	1st run		2nd run		3rd run	
0	Valve	557	0	557	0	557	0
		1	72	1	72	1	72
	Pump	566	0	566	0	565	1
		3	68	3	68	5	66
	Fan	615	0	614	1	615	0
		1	224	1	224	2	223
	Slide	484	0	483	1	484	0
		0	137	0	137	0	137

Fig. 4. Confusion matrices obtained in the proposed approach experiment with dynamic criteria's and SMOTE on training sets.

study with dynamic criteria. Each value of the performance metrics was averaged over three training and testing runs, with independent initialization.

Table 5

Averages of accuracy, AUC, F1-score, precision, recall, MCC, and Kappa of the proposed model on the testing set after three training and testing runs with independent initialization.

Machine	SNR (dB)	Accuracy	AUC	F1-score	Precision	Recall	MCC	Kappa
Valve	6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	0	0.99841	0.99521	0.99310	1.00000	0.98630	0.99224	0.99221
	-6	0.99418	0.99992	0.97551	0.95221	1.00000	0.97259	0.97221
Pump	6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	0	0.99372	0.99728	0.97112	0.99502	0.94836	0.96796	0.9676
	-6	0.98744	0.99648	0.94322	0.95263	0.93427	0.9363	0.93617
Fan	6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	0	0.99802	0.99999	0.99629	0.99852	0.99408	0.99494	0.99493
	-6	0.96627	0.99376	0.93554	0.95811	0.91407	0.91321	0.91272
Slide	6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	0	0.99946	1.00000	0.99879	0.99758	1.00000	0.99845	0.99844
	-6	0.99195	0.99978	0.98168	0.98529	0.9781	0.97654	0.97653

The results in Table 5 show that with an SNR of 6 dB, the proposed model performs perfectly for all machines, as indicated by all metrics being uniformly equal to 1, demonstrating that the model works best with high-quality signals under the highest signal-quality condition in the MIMII dataset. At SNR levels of 0 dB and -6 dB, where noise interference is more hardened, there is a slight degradation in performance, particularly for pump and fan machines. Regarding the recall scores of the pump, note the least recall (0.94836 at 0 dB, 0.93427 at -6 dB) and MCC (0.96796 at 0 dB, 0.9363 at -6 dB).

It is worth noting that data leakage is explicitly prevented by the strict per-device and per-state data-splitting strategy described in Section 2. All splits were performed before pre-processing, and SMOTE was applied exclusively to the training set. The validation and testing sets remained untouched, and feature normalization parameters were derived only from the training data. In addition, perfect performance was observed only at 6-dB SNR, which is the highest-quality signal condition in the MIMII dataset. At lower SNRs (0 dB and -6 dB), the performance consistently degrades (e.g., Fan Recall = 0.91407 at -6 dB), which would not occur if the model was overfitting or memorizing the data. Overfitting is further prevented through early stopping, dynamic LR scheduling, dropout, and BN. The stability of the results across runs, together with the SNR-dependent performance trend, supports the validity of the reported results.

In general, at an SNR of -6 dB the fan appears to be the most challenging machine, despite being better than the pump at an SNR of 0 dB, suggesting that some patterns tend to be difficult to identify under different noise conditions. Nevertheless, considering the overall performance, it indicates the robustness of the proposed model at different SNR levels given that no special methods to treat noise were applied. In addition, the results strongly suggest that if noise is handled, the proposed model can achieve better predictions across all machines regardless of the complexity of their sound patterns.

However, for additional investigation, three additional experiments were provided for the most challenging machine, which was the fan (noting that “full model” will refers to the proposed model with SMOTE and all dynamic criteria in this work) and as follows:

i) The same model, but with SMOTE and without dynamic criteria: The training completes the 100 epochs and the main degradation is at -6 dB, where the abnormal (minority-class) recall falls to 0.88444 (it was 0.91407 on average in the full model), causing missing more minority samples and the misclassified samples increasing from ~28 (in the full model) to 34. At 0 dB, the degradation is small but still measurable, where the misclassified samples range from 1 (in the full model) to 2. At 6 dB, this effect was negligible.

ii) The same model, but without SMOTE and with dynamic criteria: At -6 dB, the model recall for abnormal (minority class) drops to 0.8933, meaning it misses more from the minority class than it did (with 0.91407 on average in the full model); therefore, the total number of misclassified samples also increase from 28 to 31. At 0 dB, the decline was still noticeable when the misclassified samples ranged from 1 to 3. At

6 dB, the effect is minimal because the performance remains near-perfect, but (without SMOTE) still sees a single misclassified sample instead of none in the full model.

iii) The same model but without SMOTE and without dynamic criteria: The training completes the 100 epochs and the drop at -6 dB becomes very pronounced, where the recall for the minority class decreased from 0.91407 on average in the full model to 0.85333, causing 40 misclassified samples in total, compared to only approximately 28 in the full model. At 0 dB, the degradation is minor but still observable, where the misclassified samples increase to 3. At 6 dB, the effect is negligible, whereas the performance remains perfect, with zero misclassified samples.

3.3 Comparisons of results

To maintain consistency and reliability, this study compares the results of the proposed method with those of the supervised methods from the literatures, where both are evaluated on the same dataset using a clearly explained data splitting approach. Comparisons are limited to officially published studies with well-defined metrics and experiments conducted on a full dataset or reasonable subsets. For each machine, all related data were thoroughly considered. Only studies presenting numerical metrics in tables or text were included, avoiding estimates from figures unless values were clearly discernible.

Table 6 shows a comparison of the results between the proposed model and those in Ref. [18], specifically for the best AUC results achieved by the authors when they implemented MLP with a full stack of features used, including MFCC. Improvements were made in all the metrics of the proposed model.

Table 6

Comparison of the results between the proposed method and those in Ref. [18] in terms of AUC.

Machine	SNR (dB)	MLP + full features in Ref. [18]	Proposed
Fan	6	0.997	1.00000
	0	0.977	0.99999
	-6	0.917	0.99376
Pump	6	0.983	1.00000
	0	0.966	0.99728
	-6	0.928	0.99648
Slide	6	0.994	1.00000
	0	0.985	1.00000
	-6	0.961	0.99978
Valve	6	0.929	1.00000
	0	0.842	0.99521
	-6	0.766	0.99992

Table 7 shows the average results of the proposed model compared with the results of the proposed weighted majority voting (WMV) approach in Ref. [20]. Considering all 14 values (for both pump and valve) of the comparison metrics, the proposed method achieved improvement over WMV in 10 of them.

Table 7

Comparison of the averaged results over all SNRs with results of WMV in Ref. [20].

Machine	Work	Accuracy	AUC	F1-score	Precision	Recall	MCC	Kappa
Pump	Proposed	0.99372	0.99792	0.97145	0.98255	0.96088	0.96809	0.96792
	WMV	0.97447	0.99810	0.97562	0.98198	0.96535	0.94893	0.94884
Valve	Proposed	0.99753	0.99838	0.98954	0.98407	0.99543	0.98828	0.98814
	WMV	0.98211	0.99930	0.98289	0.98221	0.98357	0.96414	0.96414

Furthermore, Ref. [21] achieved an accuracy of 97.8 in classifying fan sounds, while the proposed method achieved an average of 0.9881.

In addition, based on confusion matrices for LGBM and SVM in the partially mixed classification method proposed in Ref. [22], Table 8 shows that the proposed approach significantly outperforms all the results of the classification performance.

Table 8

Comparison of the classification performance with partially-mixed classification in Ref. [22].

SNR (dB)	Machine	State	Model	Value	Proposed
6	Valve	Normal	LGBM	0.83	1.00000
		Abnormal	SVM	0.68	1.00000
	Pump	Normal	SVM	0.99	1.00000
		Abnormal	SVM	0.93	1.00000
	Fan	Normal	LGBM or SVM	0.99	1.00000
		Abnormal	LGBM or SVM	0.99	1.00000
	Slide	Normal	SVM	0.96	1.00000
		Abnormal	SVM	0.95	1.00000
	Valve	Normal	LGBM	0.67	1.00000
		Abnormal	SVM	0.47	0.98630
0	Pump	Normal	LGBM	0.89	0.99941
		Abnormal	SVM	0.81	0.94836
	Fan	Normal	LGBM or SVM	0.90	0.99946
		Abnormal	SVM	0.96	0.99407
	Slide	Normal	SVM	0.80	0.99931
		Abnormal	SVM	0.88	1.00000
	Valve	Normal	LGBM	0.47	0.99342
		Abnormal	LGBM	0.31	1.00000
	Pump	Normal	LGBM	0.64	0.99411
		Abnormal	SVM	0.68	0.93427
-6	Fan	Normal	LGBM	0.66	0.98537
		Abnormal	SVM	0.92	0.91407
	Slide	Normal	LGBM	0.49	0.99587
		Abnormal	SVM	0.74	0.97810

4 Discussion

The stratified data splitting method is typically used to preserve class proportions across the training, validation, and testing sets, yet it may fail to reflect the higher-level structure inherent in multi-branch, multi-level hierarchical datasets, such as MIMII, unless additional constraints are explicitly enforced. Such a stratification split may not guarantee that the devices are fairly represented within each class. This is not a trivial theoretical concern in multi-device industrial data, where device-to-device variability often embeds recording, sensor, and environmental signatures that can lead a DL model to learn device-specific signatures rather than fault-related patterns. As a result, performance may appear stable while the underlying device composition drifts across the splits.

Therefore, the contribution of the adopted hierarchical multi-stage stratified splitting protocol is not the claim that the baseline stratification split method “does not work”, but that it is insufficient to satisfy a device-level fairness requirement. This requirement was applied through a multistage hierarchical stratified splitting protocol (SNR→machine type→device ID→state/label), which performs splitting within each (device, state) group before preprocessing. The effectiveness was confirmed using the maximum deviation in device composition within each class (see Section 2), where the baseline stratified splitting method exhibits substantial drift (normal: max $\Delta = 0.040849$; abnormal: max $\Delta = 0.102576$), whereas the adopted protocol reduces these deviations to near-zero levels (normal: 0.000722; abnormal: 0.010100), indicating that the issue lies not in stratification itself, but the level at which constraints are imposed.

With respect to SMOTE, the pre-SMOTE totals are approximately 30883 normal samples versus 6912 abnormal samples in the training set, that is, about a 4.47:1 ratio in favor of the normal class (about 81.7%) versus abnormal (about 18.3%). In addition, skipping dynamic training controls such as EarlyStopping and ReduceLROnPlateau, even when keeping SMOTE, would increase the cost by approximately 80%. Removing

SMOTE alone or by removing the dynamic criteria would cause a bias classifier, especially under higher noise conditions toward the majority class, and lead to a clear drop in the ability to classify the minority class (abnormal) classification, especially in high noise conditions, as shown by the experiments implemented in the results.

Additionally, the comparative results in Tables 6–8 indicate a meaningful advantage for the proposed model, given that the comparisons were restricted to officially published studies on the same dataset with clearly reported numerical metrics. Against Ref. [18] and Table 6, the improvement in AUC holds across all machine types and SNR levels. More importantly, the gap with Ref. [18] tends to increase as the signal quality deteriorates (0 dB and –6 dB), suggesting that the proposed model does not outshine on a better signal (6 dB) but remains competitive under more challenging noise conditions. This is particularly evident for the valve machine, where in Ref. [18] AUC increases (from 0.766, 0.842, to 0.929), whereas in the proposed model, it is more stable (from 0.99521, 0.99992, to 1.00000). In comparison with the WMV approach shown in Table 7 proposed in Ref. [20], considering all 14 values (for both pump and valve) of the comparison metrics, the proposed model achieved improvement over WMV in 10 of them. In addition, the fan result reported in Ref. [21] (97.8% accuracy) is comparable to the proposed method (0.9881 average accuracy). However, Ref. [21] showed signs of overfitting under low signal quality (95% training vs. 85% testing accuracy), whereas the proposed model maintained competitive performance under more severe noise conditions (0 dB and –6 dB). Finally, Table 8 provides a comparison with the partially mixed classification method in Ref. [22], where, over all machines, SNR levels, and crucially across states, the proposed model yields superior performance. The most significant improvements appeared for the abnormal class under 0 dB and –6 dB. Overall, these demonstrate more robust and consistent gains in malfunction detection, with benefits under noisier conditions, which are more realistic in industrial environments.

5 Conclusions

Audio analysis could be important for industrial operations when it serves as a data source that reveals information about a machine's mechanical states. Acoustic analysis methods help manufacturers discover operational abnormalities, including internal problems that cannot be detected by traditional sensing methods. The complex nature of sound signals requires both specific preprocessing and the recognition of particular acoustic features as part of their non-stationary characteristics. When dealing with unbalanced datasets, the task becomes more difficult because this situation may cause DL models to be biased toward predictions that align with the major classes. Accurate model performance requires selecting suitable acoustic features together with a proper data-splitting strategy. In a supervised CNN model, featuring an early stopping method and a dynamic LR procedure was proposed to achieve the improved model performance across all metrics, enhance generalization, and reduce training costs by automatically managing the number of epochs. The assessment of the model was implemented using the MIMII real-world dataset, which was imbalanced and contained multiple SNR. SMOTE was applied to balance the training data after feature extraction. The adopted data splitting strategy uses devices and their state as the unit to distribute samples between the training, validation, and testing phases to maintain fair representative data in each subset. The results showed the optimal model performance at an SNR of 6 dB for all machines with minimal performance degradation when the signal was corrupted by the SNR (0 dB and –6 dB). Within the MIMII dataset, pump demonstrated the lowest values of recall 0.94836 at 0 dB and 0.93427 at –6 dB, as well as MCC 0.96796 at 0 dB and 0.9363 at –6 dB. Interestingly, while the fan outperformed the pump at 0 dB, it became more difficult at –6 dB.

Future research may include testing the proposed model by using industrial machine datasets to establish its true generalization. Future analysis will also determine the performance of the proposed system when using multiple noise treatment methods and data streaming conditions.

CRedit authorship contribution statement

Ameer B. A. Alaasam: Writing—original draft, Visualization. **Zena H. Khalil:** Resources, Methodology. **Dhiah Al-Shammary:** Investigation, Formal analysis. **Ayman Ibaida:** Formal analysis, Data curation. **Khandakar Ahmed:** Data curation, Conceptualization.

Declaration of competing interest

The authors declare no conflicts of interest.

References

- [1] R.V. Sharan, H. Xiong, S. Berkovsky, Benchmarking audio signal representation techniques for classification with convolutional neural networks, *Sensors* 21 (10) (2021) 3434.
- [2] L. Hertel, H. Phan, A. Mertins, Comparing time and frequency domain for audio event recognition using deep learning, in: Proc. of the 2016 Intl. Joint Conf. on Neural Networks (IJCNN), Vancouver, Canada, 2016, pp. 3407–3411.
- [3] P. Golik, Z. Tüske, R. Schlüter, et al., Convolutional neural networks for acoustic modeling of raw time signal in LVCSR, in: Proc. of the INTERSPEECH 2015, 16th Annual Conf. of the Intl. Speech Communication Association, Dresden, Germany, 2015, pp. 26–30.
- [4] T. Sercu, C. Puhersch, B. Kingsbury, et al., Very deep multilingual convolutional neural networks for LVCSR, in: Proc. of the 2016 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016, pp. 4955–4959.
- [5] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proc. of the 2013 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Vancouver, Canada, 2013, pp. 6645–6649.
- [6] D.S. Park, W. Chan, Y. Zhang, et al., SpecAugment: a simple data augmentation method for automatic speech recognition, in: Proc. of the 20th Annual Conf. of the Intl. Speech Communication Association, Graz, Austria, 2019, pp. 2613–2617.
- [7] V. Pann, K.S. Kwon, B. Kim, et al., Robustness of CNN-based model assessment for pig vocalization classification across diverse acoustic environments, *Comput. Electron. Agric.* 240 (2026) 111181.
- [8] A.B.A. Alaasam, G.I. Radchenko, A.N. Tchernykh, Micro-workflows data stream processing model for industrial internet of things, *Supercomput. Front. Innov.* 8 (1) (2021) 82–98.
- [9] J. Konar, P. Khandelwal, R. Tripathi, Comparison of various learning rate scheduling techniques on convolutional neural network, in: Proc. of the 2020 IEEE Intl. Students' Conf. on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1–5.
- [10] A. Fernández, S. García, F. Herrera, et al., SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary, *J. Artif. Intell. Res.* (1) (2018) 863–905.
- [11] Y.-H. Chen, Z.-Q. Xie, J.-Y. Zhong, et al., SQKformer: spiking sparse QKformer with adaptive batch normalization for membrane potential, *Neurocomputing* 671 (2026) 132666.
- [12] X.-Y. Jin, Analysis of residual block in the ResNet for image classification, in: Proc. of the 1st Intl. Conf. on Data Analysis and Machine Learning, Kuala Lumpur, Malaysia, 2023, pp. 246–250.
- [13] W. Han, C.F. Chan, C.S. Choy, et al., An efficient MFCC extraction method in speech recognition, in: Proc. of the 2006 IEEE Intl. Symposium on Circuits and Systems, Kos, Greece, 2006, p. 4.
- [14] M. Maseri, M. Mamat, Malay language speech recognition for preschool children using hidden Markov model (HMM) system training, in: Proc. of the 5th ICCST 2018 on Computational Science and Technology, Kota Kinabalu, Malaysia, 2019, pp. 205–214.
- [15] N. Madhu, Note on measures for spectral flatness, *Electron. Lett.* 45 (23) (2009) 1195–1196.
- [16] J. Herre, E. Allamanche, O. Hellmuth, Robust matching of audio signals using spectral flatness features, in: Proc. of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575), New Platz, USA, 2001, pp. 127–130.
- [17] T. Giannakopoulos, A. Pikrakis, Audio features, in: T. Giannakopoulos, A. Pikrakis (Eds.), *Introduction to Audio Analysis*, Elsevier, Amsterdam, The Netherlands, 2014, pp. 59–103.
- [18] L. Gantert, M. Sammarco, M. Detyniecki, et al., A supervised approach for corrective maintenance using spectral features from industrial sounds, in: Proc. of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, USA, 2021, pp. 723–728.
- [19] L. Gantert, M. Sammarco, M. Detyniecki, et al., Super learner ensemble for sound classification using spectral features, in: Proc. of the 2022 IEEE Latin-American Conf. on Communications (LATINCOM), Rio de Janeiro, Brazil, 2022, pp. 1–6.
- [20] B.A. Tama, M. Vania, I. Kim, et al., An EfficientNet-based weighted ensemble model for industrial machine malfunction detection using acoustic signals, *IEEE Access* 10 (2022) 34625–34636.
- [21] F. Qiu, S. Chen, D. Jin, et al., Fan sound fault detection algorithm based on the fusion of convolutional neural network and gated recurrent unit network, in: Proc. of the Sixth Intl. Conf. on Advanced Electronic Materials, Computers, and Software Engineering (AEMCSE 2023), Shenyang, China, 2023, p. 127870P.
- [22] L. Gantert, T. Zeffiro, M. Sammarco, et al., Multiclass classification of faulty industrial machinery using sound samples, *Eng. Appl. Artif. Intell.* 136 (2024) 108943.



Ameer B.A. Alaasam received the Ph.D. degree in computer science from South Ural State University, Chelyabinsk, Russia in 2022. He is currently an Associate Professor with the Department of System Programming, School of Electronic Engineering and Computer Science, South Ural State University. He has been involved in teaching and scientific research in computer science since 2007. He has participated in multiple funded research projects in Russia focusing on fog computing, digital twins, and containerized cloud solutions. He developed registered platforms, including CLOUDFOGNET and a micro-workflow concept implemented in the Kepler Scientific Workflow platform. His research interests include stream processing, distributed and parallel computing, digital twins, and smart industry technologies.



Zena H. Khalil received the B.Sc. degree in computer science from the University of Al-Qadisiyah, Al Diwaniyah, Iraq, and the M.Sc. degree from Babylon University, Hillah, Iraq in 2005 and 2011, respectively. In 2021, she completed postgraduate studies from South Ural State University. Since 2005, she has been employed as a Lecturer with the University of Al-Qadisiyah. She also worked as a Research Engineer with South Ural State University from late 2018 to late 2020. Her research interests include signal and digital image processing, remote sensing, artificial intelligence (AI), machine learning, and biometrics.



Dhiah Al-Shammary obtained the Ph.D. degree in computer science from RMIT University, Melbourne, Australia in 2014. Currently he is a Senior Academic at the University of Al-Qadisiyah. He worked for several universities in both Iraq and Australia. He worked in some Silicon Valley based companies in Australia like Optcts and AgilePQ. He has published with his Australian-USA team a potential patent for designing new post-quantum data encryption with high performance. He has several publications at highly reputed venues. His special interests include data security and privacy, clustering techniques, classifications methods, optimization, and networking.



Ayman Ibaida received the Ph.D. degree in computer science and IT from RMIT University in 2014. He was a Lecturer with the Computer College, United Arab Emirates University, Al Ain, Dubai from 2006 to 2008. He was the Technical Lead at AgilePQ Australia Ltd. and a Cofounder of EyeCura Pty Ltd., Melbourne, Australia in 2017. He is currently a Lecturer with Victoria University, Melbourne, Australia. His research interests include AI and machine learning in biomedical applications, diagnoses, and patient health records security as well as cyber security in healthcare systems.



Khandakar Ahmed received the Ph.D. degree from RMIT University in 2015 and the M.Sc. degree in networking and e-business centered computing (NeBCC) under the joint consortia of the University of Reading, Reading, UK in 2009; Aristotle University of Thessaloniki, Thessaloniki, Greece, and Charles III University of Madrid (UC3M), Madrid, Spain in 2011. He started the career as a Full-Stack Developer and later switched to the academy by joining his alma mater as a Lecturer in 2007, when he completed his undergraduate program. He taught at several universities in Europe and Australia and worked as a Post-Doctoral Research Fellow at Royal Melbourne Institute of Technology (RMIT) from 2015 to 2016, before joining Victoria University in 2017. He is currently a Senior Lecturer with the College of Engineering and Science, Victoria University. He has extensive industry engagement as a Chief Investigator in multiple research projects related to the Internet of things, smart cities, machine learning, cybersecurity, and biomedical informatics.