

# Development of Fault Detector with Acoustic Emission Discrimination for Mechanical Motors

Joy Iong-Zong Chen\* and Wen-Chueh Lo

Dep. of Electrical Engineering, Da-Yeh University, No. 168, University Rd., Dacun, Changhua 515006, Taiwan

(Received June 12, 2023; accepted September 28, 2023)

**Keywords:** acoustic emission (AE), AIoT, edge computing (EC), feature extraction, tiny machine learning (TinyML)

The autonomous fault diagnosis of mechanical systems is crucial to addressing smart manufacturing product issues. In this article, we propose intelligent diagnosis and prediction technologies based on acoustic emission (AE) for mechanical motors. The integration of practical technologies, such as acoustic analysis, artificial intelligence (AI), edge computing (EC), electromagnetics, communication, and other theory-based subjects, is convenient for achieving flexible changes made in response to the edge operation trend. The proposed model, developed using acoustic information links with machine learning (ML) platforms to collect acoustic information via feature extraction (FE), is novel in that it can detect system health and prevent system failures. It can inspire innovative design concepts once the above model is combined with the EC migration module. In addition, in this paper, we discuss the embedded system in smart manufacturing applications, including AE, to establish an ML framework that is trained using audio emission data. The valuable results from the proposed algorithm experiments show that the audio judgment accuracy rate can be above 90%. At the current stage, the metric accuracy and precision of mechanical motor discrimination can reach 93.5% and 0.97, respectively. In this paper, we present an analytical method for performing motor axis misalignment judgment based on tiny machine learning (TinyML) techniques, which will enable the IoT field to move toward smart energy savings.

## 1. Introduction

Industrial development is gradually shifting towards intelligent autonomy in line with the Industry 4.0 development trend. A well-known technique for obtaining information about the critical conditions of structural systems is acoustic emission (AE), which is developed from the acoustic data obtained using a data-driven machine learning (ML) system. The AE system could be a tectonic plate with a characteristically large length or an electronic device with a length of millimeters. Its various sources include force input or other mechanical excitations that produce AE events that would merit necessary research. Using the gathered AE data can identify several

---

\*Corresponding author: e-mail: [jchen@mail.dyu.edu.tw](mailto:jchen@mail.dyu.edu.tw)  
<https://doi.org/10.18494/SAM4545>

global parameters, and it is known that the evolution of these parameters under certain circumstances defines when and how the structural damage process evolves to the point of collapse. With better artificial intelligence (AI) development, machine intelligence autonomy is not far away. Saucedo-Dorantes *et al.* decomposed the vibration signal in multiple IMFs and identified the vibration modes under different fault conditions.<sup>(1)</sup> Generally, the damage process is difficult to prove. The AE signals used in a bidirectional, unique link with pure information, collected from an autonomous system, can be utilized to identify the damage process. In addition, fault detection devices for wind turbines have been designed,<sup>(2,3)</sup> and in other works, accelerometers (acceleration) and GPS tracking wireless systems have been employed.<sup>(4)</sup> Adding sound leakage judgment technology makes it possible to accurately monitor the device repair and maintenance process in a dynamic location. Using a radio-frequency-transmitting chip on a circuit board,<sup>(5)</sup> information transmission can be achieved to create an interface for exchanging messages and accomplish remote monitoring. The acoustic application discussed in this article is not limited to industrial motors but is also applied to households. For example, the signal that comes from the square current signal used to compare the differences can be employed to make the model more accurate.<sup>(5)</sup> A study on acoustic systems revealed that speech pathology can also adopt abnormal sound detection using autocorrelation as a feature to detect and classify pathological samples.<sup>(6)</sup> These features were investigated analytically in different frequency bands to assess their contribution to the detection and classification process for each band. For example, the above research data were used in combination with an array of microphones and signal characteristics for monitoring.<sup>(7)</sup> Furthermore, in order to complete signal collection and application, a context-gated convolution was adopted to build an end-to-end learning framework, enabling convolutional layers that dynamically capture representative local patterns and compose local features of interest under the global context.<sup>(8)</sup>

Edge computing (EC) has been widely adopted in industries such as industrial manufacturing, agricultural production, medical applications, and environmental monitoring. Some of the main advantages of using EC in IoT applications are summarized here: the overall system delay is significantly reduced, the data leakage risk during data transmission can be avoided, data management resiliency and reliability are improved, reliability is better, the latency, bandwidth consumption, and energy requirements are significantly reduced, and most of the recent AI and ML applications are supported.<sup>(9,10)</sup> Generally, building a model through ML and applying it to the machine automation process is nothing more than saving manpower and reducing erroneous actions caused by human error. Therefore, a set of training sites suitable for automated production lines with self-diagnosis capabilities and even error correction functions is worth establishing for ML applications.<sup>(11)</sup> This is an interesting issue addressed by Ren *et al.*, who embedded frameworks trained by tiny ML (TinyML) into the proposed system, which could receive the computational results from the EC components.<sup>(12)</sup> Incidentally, Zhan *et al.* presented a resource scheduling model for EC using the Markov decision process and deep reinforcement learning algorithm.<sup>(13)</sup> Collaborative learning procedures and deep learning models have also been used to address resource scheduling issues in mobile edge computing.<sup>(14)</sup>

As previously discussed, many upcoming concepts will be used to implement a smart industrial robot in this study. Accordingly, a context-gated convolution is adopted to build an

end-to-end learning framework that enables convolutional layers to dynamically capture representative local patterns and compose local features of interest under the global context.<sup>(15)</sup>

IoT is currently a thriving technology trend that is being discussed around the world. There has been increasing growth in modern low-latency computing applications, including (i) augmented reality (AR) applications, (ii) real-time traffic control systems that require low-latency responses to avoid potential collisions,<sup>(16)</sup> and (iii) real-time smart grid management systems that aggregate data from distributed geosensors and control the grid in real time.<sup>(17)</sup> Moreover, Ma *et al.* demonstrated a switching multitier control (SMC) architecture that can be optimized and controlled by local and edge controllers over a multi-tier network.<sup>(18)</sup> A case study of an industrial robot is used to prove that the SMC significantly outperforms both a local and an edge controller in a wireless network. In addition, Kianoush *et al.* focused on a heterogeneous ecosystem design for IoT radio-based systems that combine and transform multiple RF sources/detectors into different virtual sensors.<sup>(19)</sup> They also applied long-short term memory (LSTM) and convolutional neural network (CNN) tools to capture photo frames and perform the position/activity classification at the cloud backend.

Edge computing complements the backend computing provided by the cloud to fill up the critical latency gaps between the endpoints and the cloud.<sup>(20)</sup> To achieve efficient processing at the edge, smart gateways and micro-data centers are two key methods proposed in the literature.<sup>(21)</sup> The primary benefit of edge computing comes from its ability to offer low-latency computing resources on the fly for applications that have strict latency requirements. All the above-mentioned solutions assume the resource sharing and latency benefits of the edge computing model. To the best of our knowledge, the work presented in this paper is the first research effort on the fault detection of motors and is focused on fault diagnosis and dynamic edge computing.

Several traditional data extraction methods are used for classification that could provide sufficient information for further data analysis classification steps. The dimension reduction method plays an inevitable role in analyzing and visualizing high-dimensional multisource data. Fanaee and Thoresen adopted quality metrics to compare the performance for multi-ohmic integrative analysis datasets from the multiview category to evaluate benchmark method performance.<sup>(22)</sup> In Ref. (23), maximal independent classification information and minimal redundancy (MICIMR), a hybrid altered FS algorithm based on information theory, was proposed. It outperforms other FS algorithms. In addition, classical ML methods were applied to provide expert support in terms of time and effort by making sense of features.<sup>(24)</sup> However, images may be affected by different lighting conditions during image acquisition, which causes worse performance results.<sup>(25)</sup> Fira and Goras utilized nonlinear approaches to reduce dimensionality for ECG signal detection.<sup>(26)</sup> Also, the nonlinear dimensionality reduction approaches with different degrees of effectiveness were investigated using electrocardiography (ECG) signals as to multiple classifier performance.<sup>(27)</sup> Furthermore, Alshorman *et al.* and Liu *et al.* conducted comprehensive discussions to establish the current state-of-the-art techniques for fault detection in mechanical motors using the AE method.<sup>(28,29)</sup> The above-mentioned research groups reviewed condition monitoring (CM) and fault diagnosis (FD) studies in terms of sound and AE for four types of fault bearing, rotor, stator, and compound.

The motivation and objective outcomes of the contribution of our proposed algorithms are summarized below.

1. Develop a framework employing ML with AE datasets gathered, trained, and validated with six situations in the current research.
2. Develop different methods for the collection of datasets, which are preprocessed by feature extraction. The results confirmed that dataset preprocessing promotes an accurate rate of performance for the training results.
3. Apply the EC combined with the proposed ML model to a practical scenario that validates the experimental results.
4. The proposed fault detection system would be implemented by manufacturers if the “noise constraint” could be avoided.

The remainder of this paper is organized as follows. Section 2 is dedicated to the theoretical methodologies deployed in the development. In Sect. 3, we present the development stages of the proposed algorithms and applied methods for the implemented systems. The results and discussion are provided in Sect. 4. The conclusions are presented in Sect. 5.

## 2. Preliminaries of the Proposed Methodologies

In this section, all applied methodologies are described in detail with four schemes: the proposed application to establish the neural network (NN) model framework, the dimensionality reduction scheme, the EC scheme, and the feature extraction scheme.

### 2.1 Proposed scheme with CNN framework

This research is based on the supervised neural network framework technology and uses the CNN ML model to collect the AE characteristics generated by rotating shaft drive devices. The system then trains and builds models for different AE characteristics.<sup>(30)</sup> The proposed mechanism is based on the CNN ML structure and is shown in Fig. 1.<sup>(31)</sup> The basic idea behind the CNN module is the convolution process, which can be regarded as correlated. The above-mentioned process is useful to help the feature correlation adopted later in this work. The CNN model used in the developed mechanism initially converts the input into subclasses based on the

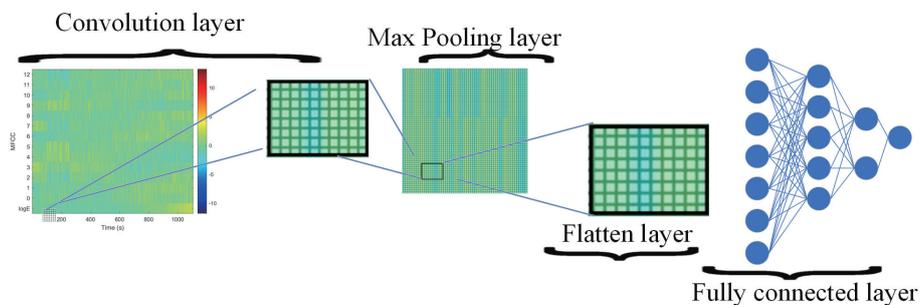


Fig. 1. (Color online) Proposed scheme with CNN framework.

requirements and is first provided to the convolution layer. The convolution layer and maximum pooling layers are used in the proposed architecture to extract useful AE signals from the resource requirements.<sup>(28)</sup> The CNN data processing capability is much better and can maintain neighborhood relationships compared with traditional neural network models. The automatic training process makes the network adopt various data features while representing those features. Let us carry out a simple review of the CNN model. Consider the weights of the one-dimensional kernel as  $\{k_1, k_2, \dots, k_n\}$ , where  $n$  represents the length of the kernel. Accordingly, the convolution process is mathematically derived as

$$O_p = f\left\{\sum_{u=1}^n k_u * x_{p-u+1}\right\}, \quad (1)$$

where the input sample is represented as  $\{k_1, k_2, \dots, k_n\}$  and the information generated at time  $p$  is represented as  $O_p$ .  $f\{\}$  represents the activation function. In the developed model, the rectified linear unit (ReLU) is the activation function, which is normally expressed as

$$ReLU\{l\} = \begin{cases} l, & l > 0 \\ 0, & l \leq 0 \end{cases} \quad (2)$$

Max pooling is employed in the proposed architecture after the convolution layer to reduce the feature dimensions. It is mathematically expressed as

$$Mp_{v,w} = MAX(h_{v, n(w-1)+r}), \quad (3)$$

where  $MAX(\cdot)$  is a function that extracts the maximum value. The filters are presented as  $v$ , then  $w$  and  $n$  represent the maximum pooled band and the pooling shift allowed between the regions, respectively. After this simple convolution operation is completed, the result is passed through the round-robin layer, and then the flatten layer operation is performed to convert the characteristic sound leak data into vector data that can be calculated. In general, the max pooling layer down-samples the convolution layer output to reduce variability. Its operator yields the maximum value. The next step after the pooling function is batch normalization in which the features are normalized to improve the training performance. Batch normalization features are expressed mathematically as

$$\mu = \frac{\sum_{n=1}^{N_{batch}} w_n}{n_{batch}}, \quad (4)$$

$$\Sigma^2 = \sum_{n=1}^N (w_n - \mu) / n_{batch}, \quad (5)$$

$$\hat{k} = k_n - \mu / (\chi + \zeta), \quad (6)$$

$$O_p = \gamma \hat{k} + \delta, \quad (7)$$

where the batch size is represented as  $n_{batch}$  and the input data is represented as  $k_n$ .  $\mu$  represents the mean and  $\chi$  represents the batch variance. The normalized data are represented as  $\hat{k}$ , and to avoid zero gradients, a constant  $\zeta$  is included in the normalized data. The vector learning parameters are represented as  $\gamma$  and  $\beta$ . The output feature is represented as  $O_p$ . Finally, the obtained characteristic signal is input through the fully connected layer. To complete the CNN ML signal model used throughout the text, each mixed feature model is established. This work will be explained later.

## 2.2 Feature extraction schemes

FE terminology is an important issue addressed in NN deep training and testing. The literature reviews of ML and deep learning are discussed in this section. FE techniques are used to find the best subset of the original set of features to improve the model performance. If a feature can be used to predict the class or is related to the class, it is valuable. Otherwise, it is useless. In addition to irrelevant features, empirical data from the feature selection literature suggest that duplicate information should also be removed.<sup>(32,33)</sup> By selecting the appropriate linear methods, it is possible to solve gene expression datasets that are not linearly separated and thus perform better classification. Key components of the preprocessed data can be represented in the compact form of feature vectors using the dimensionality reduction approach; this is known as feature extraction.

Canonical correlation analysis (CCA) is one of the statistical methods used to determine how the derived features are related to each other. On the other hand, CCA is also a multivariate statistical method that is used when there are two sets of data that may have some underlying correlation. It finds a pair of linear combinations for two sets, such that the correlation between two canonical variables is maximized. CCA extends ordinary correlation to two sets of variables and is widely used in statistical and information mining. For example, CCA finds pairs of directions  $w_g$  and  $w_s$  that maximize the correlation between the projections  $g = w_g^T$  and  $s = w_s^T$  for two random vectors  $g \in R^p$  and  $s \in R^q$ . These representations are also known as canonical variates. In most cases, the directions can be determined by finding the maxima of the function given by Bach and Jordan:<sup>(32)</sup>

$$\rho = \frac{E[gs]}{\sqrt{E[g^2]E[s^2]}} = \frac{E[w_g^T g s^T w_s]}{\sqrt{E[w_g^T g g^T w_g]E[w_s^T s s^T w_s]}}, \quad (8)$$

where  $\rho$  represents the canonical correlation and  $E$  indicates the expectation.  $c_{gg} \in R^{p \times p}$  and  $c_{ss} \in R^{q \times q}$  indicate within covariance matrix sets of  $g$  and  $s$ , respectively, and  $c_{gs} \in R^{p \times q}$  represents that  $g$  and  $s$  are between the covariance matrix set. As the equivalent eigenvalues, the highest results  $\rho(w^j)$  are considered canonical correlations.

### 2.3 Proposed scenario design and structure (edge computing)

A developed EC scheme embedded within a framework trained by TinyML techniques and many different sensors used to collect surveillance data are described in this subsection. Because the existing traditionally designed, fault detection operating mode for inline working schemes cannot meet the ML computational data or audio requirements, it is difficult to adapt the developed system to remote environments. In particular, it cannot be applied to a scenario where the computing is for the data driven at the *edge* (local but not through cloud operation). Because of the existing critical shortcomings, in the current work, ML, edge computing, and inline operation are intelligently integrated with a model trained using TinyML techniques. Eventually, different parts of the ML algorithm deployed in cloud and edge computing ends are set up in and assessed using two typical scenarios over the IoT network for the mentioned detection algorithm.

## 3. Development Stages of Proposed Algorithms

The first step in establishing the proposed fault detection model based on the CNN framework and technique of FE is the selection of parameters that determine the structure. Once the parameter identification is complete, the second step is the elimination of similar parameters. After that, there are three stages for the proposed fault detection model, as shown in Fig. 2. When the driving rotating element (such as a motor) is in a faulty operation, the sound leakage phenomenon caused by the fault is produced. In order to diagnose the cause of the fault, the proposed fault detection system uses the development process shown in Fig. 3 as its operation for the entire process. First, the possible data from the original sound leakage are obtained and then preprocessed. These processes include data cleaning, data feature selection, and dataset reconstruction. Random data sampling is then performed during this process. The system will judge the usable (positive) and unavailable (negative) data in the dataset. It then randomly processes the leakage data obtained in accordance with the model trained beforehand using different AE data to judge the fault. Eventually, the prediction can be prediagnosed on the basis of the possible factors that cause the driving rotating element failure. As might be expected, in the fault diagnosis and prediction results using the drive component failure caused by the leakage, the classifier based on the pretrained dataset can be used to classify the predicted fault types caused by the AE. The recursive method is then used to select the tone type determined by the best classifier. The identification results are finally obtained using the proposed supervised CNN platform shown in Fig. 1.

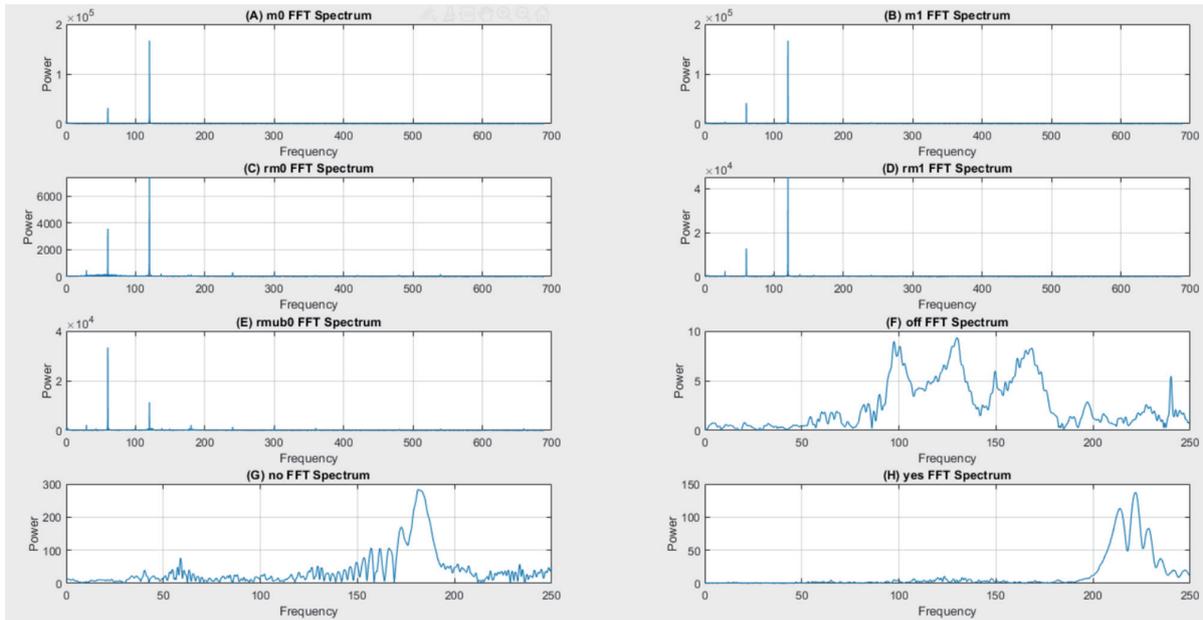


Fig. 2. (Color online) FFT spectrums of different types of AE (*m0*, *rm0*, and *mub0*) and sound audio signals (*Yes*, *No*, and *off*).

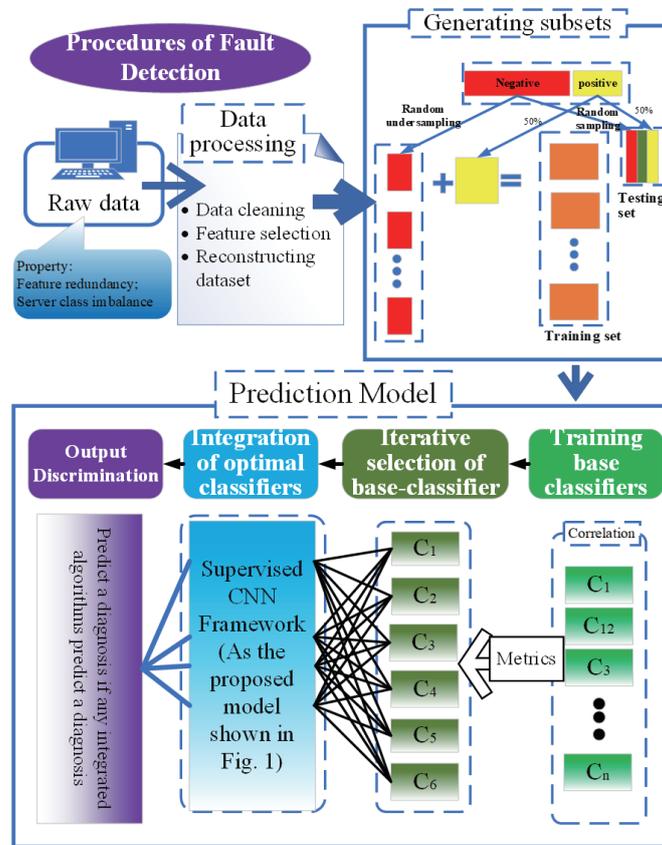


Fig. 3. (Color online) Proposed prediction model for rotating device fault detection.

### 3.1 Gaussian approximation for single distributed layer of the utilized NN mode

Figure 2 shows FFT spectrums for individual signals, which consist of five types of AE ( $m0$ ,  $m1$ ,  $rm0$ ,  $rm1$ , and  $mub0$ ) and three types of sound audio signal (*Yes*, *No*, and *off*). In Fig. 2, the graphs from (A) to (H) respectively represent the signal frequency spectrums recorded by the motor at a certain sampling time after FFT conversion. Among them, the  $x$ -axis represents *frequency* and the  $y$ -axis sampling denotes *signal power*. For example, Fig. 2(A) shows the FFT conversion results collected when the motor rotated forward ( $m0$ ) during a certain sampling time. From the information in Fig. 2, it is observed that multiple similar features are presented. The direct classification of these signals consumes much time and reduces accuracy. Hence, feature selection is required before classifying the signals. Moreover, the feature selection algorithm selects relevant features, removes redundant features, and elucidates dependences among these features. Figure 2 shows the FE output. The features are optimized using a suitable optimization algorithm [e.g., Gaussian mixture model (GMM)] to improve the classifier accuracy.

The basic CNN framework is applied and the correlation between the gathered types of AE will be described. Generally, the feature terms are usually assumed to exhibit the Gaussian distribution for ML system application. This indicates that the Gaussian approximation results are still reasonably accurate even for small tag number values  $j < 20$  when the tag number is equal to or greater than 10. Accordingly, the variance of each feature, conditioned on  $w_i$  in the  $i^{\text{th}}$  weight for the NN system, is given by<sup>(2)</sup>

$$\text{var}(F_{w_i}^{(j)})_{FE} = \sum_{i=2}^W E \left[ \left( \sum_{j=0}^{J-1} F_{w_i}^{(j)} \right)^2 \right], \quad (9)$$

where  $F_{w_i}^{(j)}$ ,  $j = 0, 1, \dots, J-1$ , denotes the value of the possibly extracted  $j^{\text{th}}$  feature from all  $J$  features and  $E \left[ \left( \sum_{j=0}^{J-1} F_{w_i}^{(j)} \right)^2 \right]$  represents the second moment of correlated features within a single distributed layer of the utilized NN mode. Since the separation between different features is deeply related to the adopted AE samples, the sample's intensity fluctuation due to features cannot be considered independent. Let  $\lambda_{nm}$  be the feature correlation coefficient between the  $F_{w_i}^{(n)}$  and  $F_{w_i}^{(m)}$  features, defined as

$$\lambda_{nm} = \frac{\{E[F_{w_i}^{(n)} \cdot F_{w_i}^{(m)}] - E[F_{w_i}^{(n)}]E[F_{w_i}^{(m)}]\}}{\{(E[(F_{w_i}^{(n)})^2] - E^2[F_{w_i}^{(n)}])(E[(F_{w_i}^{(m)})^2] - E^2[F_{w_i}^{(m)}])\}^{1/2}}, \quad (10)$$

where  $n, m = 0, 1, \dots, J-1$ , and the variable  $F_{w_i}^{(j)}$  is defined in Eq. (10). The normalized covariance matrix  $C_x$  has, as its element, the set of  $\lambda_{nm}$ ,  $n, m = 0, 1, \dots, J-1$ . The correlation coefficient of  $F_{w_i}^{(n)}$ ,  $F_{w_i}^{(m)}$  is  $\lambda_{nm}$ , which is defined as the equation above. Assume that the AE sampled frames are to be well controlled so that all features in a single-cell case have the same average second moment at the deployed scenario.

### 3.2 Raw data and dataset processing

An ensemble framework for predicting rotating equipment failures uses fixed sampling with a predetermined constant value chosen prior to the model training phase, along with an adjustable replacement strategy. The raw data were obtained from a closed space, as shown in Fig. 4(a), using normal and broken devices. The latter devices are separated into the cases of *broken in dry oil* and *broken in half-bearing*, and are shown in Figs. 4(b) and 4(c), respectively.

Four types of AE are gathered in the datasets. Data cleaning, feature selection, and dataset reconstruction are performed during data processing. Fixed sampling data are adopted with replacement to generate multiple data subsets. We chose CNN to train base classifiers and retained five base classifiers that meet our requirements using iterative selection. The final prediction results are obtained through four different ensemble algorithms and output discrimination. An ensemble-based framework is shown in Fig. 4, where the complete workflow includes the AE-collecting box and two types of gear.

The dataset employed consists of real cases investigated in the laboratory. The confirmed detection rotating device cases account only for a small part, so the problem of uneven numbers of positive and negative samples occurs in the collected data. Specifically, the collected datasets are available and can be found on the website.<sup>(34)</sup> If the unbalanced dataset (assuming the positive:negative ratio of the dataset is 9:1) is trained directly without any processing, the model can easily achieve an accuracy of more than 95% by predicting all the outputs as positive, but such a model has no practical application value. The coding tools are adopted separately in the training and EC platforms. Typically, Tensorflow<sup>®</sup> with Python<sup>®</sup> syntax is used in the former platform, and the latter platform is developed with the C<sup>++</sup> tool.

Technically, six training models, *Train-1* to *Train-6*, are provided in the investigation, as shown in Table 1. The AE signals include *forward*, *reverse*, and *unbalanced* rotations for the three parts of rotating motors. In addition, the signals add another type of sound. The AE signals for *M0* and *M1* are for forward rotation, and those for *Rm0* and *Rm1* are for reversion rotation. The AE signals for *Rmub01* and *dry* are for unbalanced rotation. Furthermore, the words *Yes*, *No*, *off*, *aa*, *ba*, *silence*, and *unknown* belong to the *Audio Sounds* dataset. Therefore, a total of 11

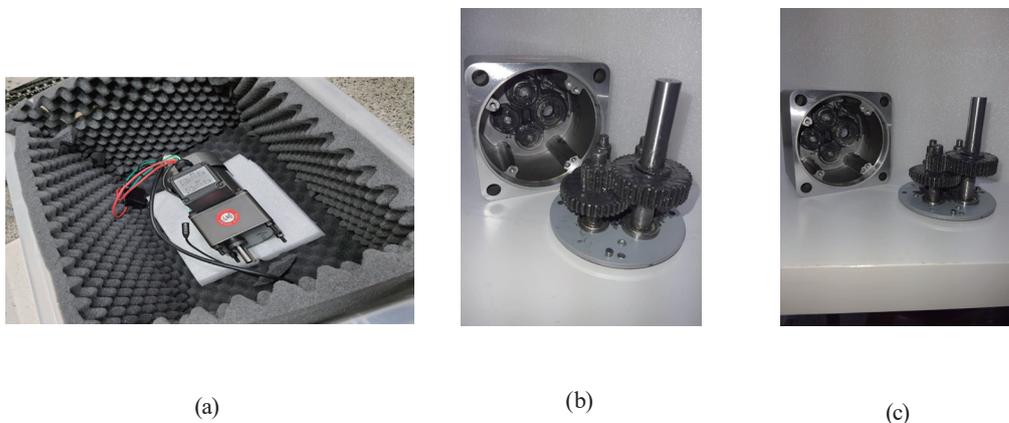


Fig. 4. (Color online) (a) AE-collecting box, (b) *Broken in dry oil* gear, and (c) *Broken in half-bearing* gear.

Table 1  
Collected datasets for different labels and six training models.

Dataset (Wanted %)	Labels							
	Forward rotation		Reverse rotation		Unbalanced rotation		Normal	
	<i>M0</i>	<i>M1</i>	<i>Rm0</i>	<i>Rm1</i>	<i>Rm ub01</i>	<i>dry</i>	<i>aa (F)</i>	<i>ba (R)</i>
Amount/giga byte	6020/0	6111/1.09	7047/1.18	7122/1.19	7135/1.19	4240/2.1	10976/1.83	11754/1.96
Frame(s)	1	1	1	1	1	1	1	1
Train-1	6020/0	6111/0	7047/0	7122/0	7135/0	4240/43	10976/0	11754/0
Train-2	6020/0	6111/0	7047/0	7122/0	7135/0	4240/45	10976/0	11754/0
Train-3	6020/0	6111/0	7047/0	7122/0	7135/0	4240/76	10976/58.5	11754/0
Train-4	6020/0	6111/0	7047/0	7122/0	7135/0	4240/66	10976/0	11754/59.5
Train-5	6020/0	6111/0	7047/0	7122/0	7135/0	4240/50	10976/0	11754/0
Train-6	6020/0	6111/0	7047/0	7122/0	7135/0	4240/64	10976/0	11754/0

Dataset (Wanted %)	Audio sounds					Accuracy	Loss
	<i>Yes</i>	<i>No</i>	<i>off</i>	<i>silence</i>	<i>unknown</i>		
Amount/giga byte	4044/0.121	3940/0.115	3745/0.112	#	#	NA	NA
Frame(s)	1	1	1				
Train-1	4044/2.5	3940/2.3	3745/2.3	#/10	#/10	92.3	7.7
Train-2	4044/2.5	3940/2.5	3745/0	#/10	#/10	95.2	4.8
Train-3	4044/4	3940/0	3745/0	#/10	#/10	97.9	2.1
Train-4	4044/0	3940/0	3745/0	#/10	#/10	96.6	4.4
Train-5	4044/0	3940/0	3745/0	#/10	#/10	99.3	0.7
Train-6	4044/0	3940/0	3745/0	#/10	#/10	99.4	0.6

types of collected audio signal are shown in Table 1 with the exclusion of *silence* and *unknown*. Specifically, *dry* is the AE signal coming from an engine that was initially destroyed by the removal of all the grease covering it. Not only the number of records for each training audio signal but also the desired *key detected word* (Wanted %) is included in Table 1.

The reasons for deploying six cases trained in the experiment are listed below.

- (A) The editorial size limitation is the main reason; in fact, there are many more models that have been trained already.
- (B) The established model accuracy rate exceeds 90%, which suitably proves the accuracy of the proposed algorithms.
- (C) Many of the situations are difficult to develop under constrained conditions, such as waste of time, cost expense, and the scale of the collected dataset, for example.

Moreover, in Fig. 5, the images generated from the MFCC transform must be recognized. Normally, the MFCC filter bank is a set of filter banks with a nonlinear distribution. The distribution is dense in the low-frequency domain and sparse in the high-frequency domain. This distribution is used to better meet the auditory characteristics of the human ear (or machine ear). The spectrogram results after FFT transformation are shown in Fig. 2. After taking the logarithm, the convolution signal is converted into an additive signal. This is the reason for the FFT and logarithm. Figure 5 shows the results of using the mel filter bank. In each subfigure, the *x*- and *y*-axes are indicated as *time* and *MFCC (frequency)*, respectively. The middle frequency components are canceled in that MFCC subfigure.

A sample experiment adopts the collected datasets shown in Table 1. The sample experiment has the training dataset with the values 6020, 6111, 7047, 7122, 7135, 4240, 10976, and 11754 for

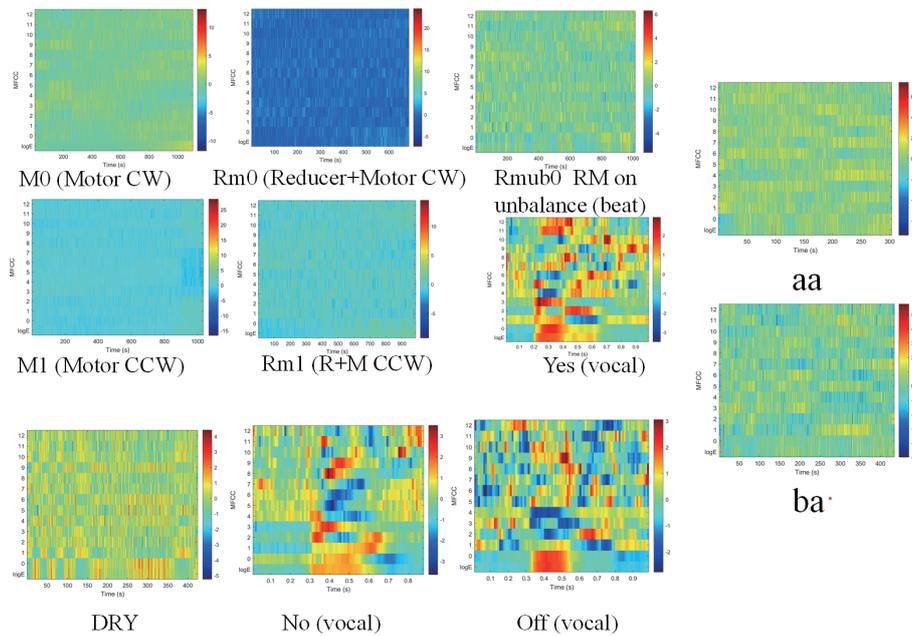


Fig. 5. (Color online) MFCC spectrums for different types of AE for rotating devices and training words.

types *M0*, *M1*, *Rm0*, *Rm1*, *Rmub01*, *dry*, *aa* (*F*), and *ba* (*R*), respectively. The words *Yes*, *No*, and *off* are assigned the values 4044, 3940, and 3745, respectively. The types *silence* and *unknown* are ignored. Moreover, the ratio for each type, *Wanted %*, is determined to be 43, 2.5, and 2.5% for *Yes*, *No*, and *off*, respectively, and 25% for both *silence* and *unknown*. The confusion matrix is going to be generated and discussed later, in which the conditions of datasets for training models are the same as those shown in Table 1.

### 3.3 Scenario design and structure (edge computing)

The main advantage of the EC operation is that it can build an adaptive model using a large amount of data through ML training on the slave device; through the serverless operation structure, the overall work can be performed during the communication flow between the master and the slave. The flow mentioned previously can avoid congestion, further reduce the memory capacity during operation, and allow the EC to respond instantly to various industrial schemes. In general, the memory capacity is a critical issue in processing the EC situation. It is commonly known to carry out data training while continuously adjusting the model weights and biases until it produces useful predictions. In the proposed method, a checkpoint is marked every 100 execution steps during the calculation process. That is, if the training fails in the middle, it can be restarted from the most recent checkpoint without losing progress. In addition, the training activity also uses Tensorflow Lite by converting the Tensorflow server into a Tensorflow Lite image file. Finally, as long as *toco* provided on Tensorflow light is used for inference, the model can be converted into the Tensorflow light format, with only one sample

being processed at a time. The complete model generated after conversion will be written to a file called *tiny\_conv.tflite*. Generally, the memory size of each output model occupies only about 48.208 kB.

The proposed EC system is shown in Fig. 6. Two main parts, *Rotating device* and *Loading screw support*, constitute the full research environment. *Mode switch*, *Buffering capacitor*, *Rotating device (Motor)*, *Gear reducer*, and *Gear of motor* are fully constructed in the first part. The second part consists of *Microphone* and *BLE-33 module*, which are responsible for dataset collection and AE detection, respectively. These two parts can each operate autonomously and are able to cooperate with each via the connector. In the BLE-33 module, there is a DSP system with an arm-4mf chip to complete the audio signal complex digital conversion calculation, which facilitates the judgment of the audio signal emitted from a specific motor. In fact, numerous different MCU modular series can be applied as EC device options. The choice of MCU module depends on the arithmetic speed and storage (RAM) size, for example, the Stm<sup>®</sup>, Raspberry<sup>®</sup>, and AMD<sup>®</sup> series.

#### 4. Results and Discussion

The relevant equipment of the experimental EC system described above was provided by Sesame Motor Company Ltd.<sup>@</sup>, and its specifications are listed in Table 2. The available rotating devices include a motor [5RK60A(GN)-CFT], a decelerator (5GN120KE with the gear reduction ratio 1:80), and SRJ-30C12 and SRJ-20C5, as shown in Table 2. A combined robot arm with a single axis (KP02602KN-300-P+C0IS01) was also included. The real experimental platform of the experimental EC system is shown in Fig. 7.

In this research, the six types of AE signal generated by a combination of motors are used to test the characteristics of different operations. They correspond to the descriptions in Table 1, which includes the operations of *M0* (normal forward), *MI* (normal reverse), *Rm0* (normal forward with a deceleration decelerator), *Rml* (normal reverse with a decelerator), *Rmub0*

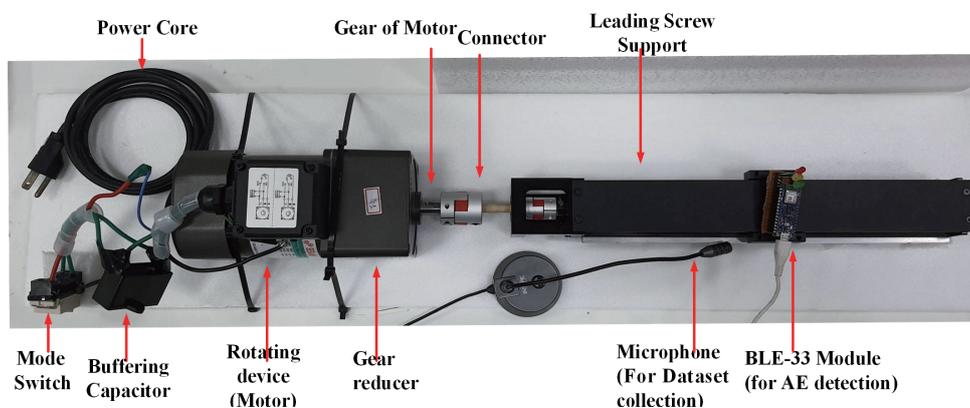


Fig. 6. (Color online) Proposed EC system.

Table 2  
Available rotating experimental devices.

Device name	Oriental Motor	Decelerator	Motor coupling	Single coupling
Type number	5RK60A(GN)-CFT	5GN120KE	SRJ-30C12	SRJ-20C5
Specifications	<ul style="list-style-type: none"> <li>• Single-phase power</li> <li>• Rating o/p 60W</li> <li>• Two rotation directions</li> </ul>	<ul style="list-style-type: none"> <li>• Gear reduction ratio of 1:80</li> </ul>	<ul style="list-style-type: none"> <li>• Flexible coupling</li> </ul>	<ul style="list-style-type: none"> <li>• Flexible coupling</li> </ul>

```

"/tmp/speech_commands_train\tiny_conv.ckpt-18000
"
I0412 18:42:04.989595 23956 train.py:292] Saving
to
"/tmp/speech_commands_train\tiny_conv.ckpt-18000
"
INFO:tensorflow:set_size=2574
I0412 18:42:05.571610 23956 train.py:296]
set_size=2574
WARNING:tensorflow:Confusion Matrix:
[[ 207   4   2   2]
 [   2 185   5  23]
 [   0   0 1725   0]
 [   5  10   1 403]]
W0412 18:42:56.858938 23956 train.py:315]
Confusion Matrix:
[[ 207   4   2   2]
 [   2 185   5  23]
 [   0   0 1725   0]
 [   5  10   1 403]]
WARNING:tensorflow:Final test accuracy = 97.9%
(N=2574)
W0412 18:42:56.858938 23956 train.py:317] Final

```

Fig. 7. Confusion matrix from training with dataset AE of *dry*.

(normal forward with a decelerator with vibration), and *Dry* (normal forward without oil). Using the measurements of the above-mentioned several characteristic forms of motor operations with the pre-training models to complete the experiment, the normal motor forward operation is considered unbalanced when the motor or shaft is without oil, which causes the forward and reverse rotations of the motor to be unclearly discriminated. Therefore, to test the motor's running characteristics with a high degree of discrimination, we adopted key feature extraction conditions to build these models. The training parameters and the presentation of different key features are also considered. It is difficult to improve the training performance when the raw data is not preprocessed. The proposed preprocessing algorithm first records the collected acoustics with a small audio slice, which is less than 1 s. The image spectrum is abstracted after the MFCC transform (as shown in Fig. 5) depending on the spectrum bandwidth. These methods were implemented using the feature extraction method discussed above.

Different training models were used to test the various motor operating modes. The training results described in the previous sections will be presented before the measurement results. The AE label training procedures with *dry* will be held as an example. There are three types of AE used in the training model. The training matrix consists of  $4 \times 4$  elements. This matrix is illustrated in Fig. 7, which shows the items consisting of the confusion matrix, the training time, the AE *dry* training label, and the final test accuracy (97.9%). In the latter case, four label types

are included. The other training model shown in Fig. 8 contains the AE *unbalanced* training label plus five other labels and achieves an accuracy rate of 92.3%. It is easy to see that a higher accuracy can be obtained with fewer labels. That is, *dry* has four label types and is less accurate than *unbalanced* with six label types. Therefore, the more training labels assigned to the training model, the lower the accuracy rate will be.

Moreover, in this study, the outcomes are validated using many merits for evaluating the performance of the established training models. The accuracy (*Acc*) is computed as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% , \quad (11)$$

where *FN* is *False Negative*, *FP* is *False Positive*, *TN* is *True Negative*, and *TP* is *True Positive*. In addition, the precision calculation (*Pre*) is depicted as

$$Pre = \frac{TP}{TP + FP} . \quad (12)$$

Accordingly, the accuracy and precision results calculated for each outcome are listed in Table 3. The results calculated for different training models, Train-1, Train-2, and Train-3, using Eqs. (11) and (12) are also shown in Table 3. The numbers of *TP*, *FP*, *FN*, and *TN* results are obtained from the Train-1 model tests, as shown in Fig. 9. Out of 200 total test results for Train-1, only 17 are displayed. Finally, the calculated metric results listed in Table 3 confirm the validity of performance for the proposed schemes with different training cases.

```
I0411 15:10:55.296143 15416 train.py:292] Saving to "/tmp/speech_commands_train/tiny_conv.ckpt-18000"
INFO:tensorflow:set_size=4427
I0411 15:10:55.841095 15416 train.py:296] set_size=4427
WARNING:tensorflow:Confusion Matrix:
[[ 717  18  2  1  0  0]
 [ 7 598  5 21 50 57]
 [ 1  0 1724  0  0  0]
 [ 6 14  0 386  7  6]
 [ 4 46  2  6 335  9]
 [ 3 60  2 10  6 324]]
W0411 15:12:22.287995 15416 train.py:315] Confusion Matrix:
[[ 717  18  2  1  0  0]
 [ 7 598  5 21 50 57]
 [ 1  0 1724  0  0  0]
 [ 6 14  0 386  7  6]
 [ 4 46  2  6 335  9]
 [ 3 60  2 10  6 324]]
WARNING:tensorflow:Final test accuracy = 92.3% (N=4427)
W0411 15:12:22.287995 15416 train.py:317] Final test accuracy = 92.3% (N=4427)
```

Fig. 8. (Color online) Confusion matrix from training with dataset AE of *unbalanced*.

Table 3  
Accuracy and precision for different training models.

	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>TN</i>	Accuracy (%)	Precision
Train-1	97	3	10	90	93.5	0.97
Train-2	90	10	15	85	87.5	0.90
Train-3	5	95	50	50	27.5	0.50

⋮

13:03:00.163 -> Heard dry0 (201) @399808ms <sup>⚡</sup>	}	TP
13:03:02.391 -> Heard dry0 (203) @404224ms <sup>⚡</sup>		
13:03:08.813 -> Heard dry0 (202) @417088ms <sup>⚡</sup>		
13:03:18.455 -> Heard dry0 (201) @436416ms <sup>⚡</sup>		
13:03:21.711 -> Heard dry0 (214) @442848ms <sup>⚡</sup>		
13:03:25.713 -> Heard dry0 (213) @450912ms <sup>⚡</sup>		
13:03:36.965 -> Heard dry0 (211) @473440ms <sup>⚡</sup>		
13:03:45.849 -> Heard dry0 (202) @491136ms <sup>⚡</sup>		
13:03:49.851 -> Heard dry0 (208) @499200ms <sup>⚡</sup>		
13:03:51.858 -> Heard dry0 (206) @503200ms <sup>⚡</sup>		
13:03:52.464 -> Heard unknown (201) @504416ms <sup>⚡</sup>	←	TN
13:03:53.301 -> Heard dry0 (201) @506048ms <sup>⚡</sup>	}	TP
13:03:59.314 -> Heard dry0 (202) @518112ms <sup>⚡</sup>		
13:04:00.108 -> Heard dry0 (213) @519712ms <sup>⚡</sup>		
13:04:19.243 -> Heard dry0 (204) @557952ms <sup>⚡</sup>		
13:04:25.692 -> Heard dry0 (205) @570816ms <sup>⚡</sup>		
13:04:28.107 -> Heard dry0 (203) @575648ms <sup>⚡</sup>		
⋮		

Fig. 9. Results obtained from Train-1 testing.

## 5. Limitations

This work focused on AE detection from different rotating motor types set up in a smart factory environment. The current working method records the sound from a single motor in a simple environment. It is known that rotating equipment is usually set up in a large manufacturing area. For experimental purposes, only one independent motor was built in our laboratory. For example, Fig. 5 shows the training dataset collection box, the complete environment platform, and the MFCC spectrums for the rotations of different types of motor. Specifically, Fig. 5 shows results for a microphone installed to collect the raw training datasets. A normal rotating motor is placed in a closed box to avoid ambient noise. This means that the raw audio signals are collected without being disturbed by any noise. In addition, an AE deployment (abnormal audio signal) detection environment is assumed to be free of interference. The motor is placed in a closed environment for audio collection. The audio collected in this study is almost pure (almost no ambient noise), but it also causes numerous test errors. Because the environment is very clean, the test becomes a major obstacle in this study. To prove that our proposed method can be used as a practical method, adding noise to simulate a real factory situation is necessary. A practical environment should be considered first when setting up a test environment. Under the current trend of intelligent automation, our method provides a solution to an important problem in the automation process, that is, machine equipment malfunction diagnosis.

In this research, the quality control (QC) stage of the production line, where frequency response analysis methods are used to analyze the audio from mechanical equipment, is examined. Such analysis methods include audio collection, training, modeling, extraction, and equipment operating test identification. The production quality from the automated production line is then estimated. At the final stage, acoustic data are converted and analyzed in real time. The application of AIoT in AI technology can solve many problems arising in factory automation activities.

## 6. Conclusions and Further Challenges

We proposed a fault diagnosis method for mechanical devices using AE models. Some valuable results were acquired experimentally. The proposed algorithm achieved an accuracy rate of more than 90%. The metric discrimination accuracy and precision for a specific mechanical motor can reach about 93.5% and 0.97, respectively. The control and operation of traditional production lines will require engineers who have a good understanding of the manufacturing process to make preliminary judgments based on their subjective experience to determine whether the actual procedures and corresponding control settings are normal. However, manually controlled production lines, which rely on experience, have more cost considerations, resulting in the transition from a manually controlled production line to an automated production line with many uncertainties. Therefore, the digital transformation concept is effective in building an intelligent production inspection auxiliary system based on knowledge from various fields. The above facts motivate the adoption of the AI methodology to develop the best application program through the analysis of big data from the production line. There still exist many open challenges that should be further investigated.

1. In many traditional factories, machinery operation still relies on the experience and judgment of plant engineers. Changing the behavior of conventional operations presents many challenges.
2. Dramatization and edge computing can be used to avoid possible errors and risks in advance.
3. Safe operation, personnel maintenance, more accurate prevention, machine maintenance, and factory operations can be realized through intelligent automation.
4. Many suitable analysis outcomes can promote excellent production rates. That is, the goal of improving control performance and reducing production cost can be easily achieved.

## References

- 1 J. J. Saucedo-Dorantes, M. Delgado-Prieto, R. A. Osornio-Rios, and R. D. J. Romero-Troncoso: *IEEE Trans. Industry App.* **53** (2017) 3. <https://doi.org/10.1109/tia.2016.2637307>
- 2 X. Pan, Z. Liu, R. Xu, J. Luo, Y. Shen, J. Qiu, L. Qi, and L. Chen: *J. Sound Vib.* **537** (2022) 117209. <https://doi.org/10.1016/j.jsv.2022.117209>
- 3 J. Huang, Z. Yang, J. Yu, L. Xiong, and Y. Xu: *IEEE Trans. Sus. Energy* **13** (2022) 31. <https://doi.org/10.1109/TSTE.2021.3102611>
- 4 V. R. Patil and S. S. Pardeshi: *Mater. Today Proc.* **72** (2023) 1975. <https://doi.org/10.1016/j.matpr.2022.11.215>
- 5 K. Rajalashmi, M. Sathiyaseelan, S. Subarathan, C. Karthi, and S. Vignesh: *Mater. Today Proc.* **66** (2022) 1284. <https://doi.org/10.1016/j.matpr.2022.05.127>

- 6 R. Jigyasu, V. Shrivastava, and S. Singh: Mater. Today Proc. **43** (2020) 355. <https://doi.org/10.1016/j.matpr.2020.11.677>
- 7 A. Al-Nasheri, G. Muhammad, M. Alsulaiman, Z. Ali, K. H. Malki, T. A. Mesallam, and M. F. Ibrahim: IEEE Access **6** (2017) 6961. <https://doi.org/10.1109/ACCESS.2017.2696056>
- 8 Institutional Repository of NYCU: <http://hdl.handle.net/11536/41088> (accessed April 2023).
- 9 Zhu, X. Liang, Y. Sun, and H. Wang: Ind. Robot. **49** (2022) 2. <https://doi.org/10.1108/IR-06-2021-0127>
- 10 F. Xue, Q. Hai, T. Dong, Z. Cui, and Y. Gong: Infor. Sci. **608** (2022) 362. <https://doi.org/10.1016/j.ins.2022.06.078>
- 11 A. Y. C. Nee and S. K. Ong: IFAC Proc. **46** (IEEE, 2013) 15.
- 12 H. Ren, D. Anicic, and T. Runkler: TinyOL: TinyML with Online-Learning on Microcontrollers, (O'REILLY W. S., USA, 2021).
- 13 W. Zhan, C. Luo, J. Wang, C. Wang, G. Min, H. Duan, and Q. Zhu: IEEE Internet of Things J. **7** (2020) 6 5449. <https://doi.org/10.1109/JIOT.2020.2978830>
- 14 M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang: IEEE Trans. Veh. Tech. **68** (2019) 7691. <https://doi.org/10.1109/TVT.2019.2922668>
- 15 X. Zhu, X. Liang, H. Y. Sun, and X. Wang: Industrial Robot **49** (2022) 301. <https://doi.org/10.1108/IR-06-2021-0127>
- 16 Cornell University: <https://doi.org/10.48550/arXiv.1810.06819> (accessed March 2023).
- 17 P. Siano: Renewable and Sustainable Energy Rev. **30** (2014) 461. <https://doi.org/10.1016/j.rser.2013.10.022>
- 18 Y. Ma, C. Lu, B. Sinopoli, and S. Zeng: IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **39** (2020) 3506. <https://doi.org/10.1109/TCAD.2020.3012648>
- 19 S. Kianoush, S. Savazzi, M. Beschi, S. Sigg, and V. Rampa: IEEE Internet of Thing J. **8** (2021) 1154. <https://doi.org/10.1109/JIOT.2020.3011809>
- 20 Cornell University: <https://doi.org/10.48550/arXiv.1502.01815> (accessed February 2023).
- 21 M. Aazam and E.-N. Huh: Proc. 2015 IEEE 29th Int. Conf. Advanced Information Networking and Apps (AINA) (IEEE, 2015) 687–695.
- 22 T. H. Fanaee and M. Thoresen: Inf. Sci. **493** (2019) 105. <https://doi.org/10.1016/j.ins.2019.04.041>
- 23 L. Zhang and X. Chen: IEEE Access **9** (2021) 13845. <https://doi.org/10.1109/ACCESS.2021.3049815>
- 24 B. Ari, A. Ari, A. Sengür, and S. A. Tuncer: Proc. 2019 1st Int. Informatics and Software Eng. Conf. (UBMYK) (IEEE, 2019) 1–5.
- 25 Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang: Sensors **20** (2020) 3535. <https://doi.org/10.3390/s20123535>
- 26 M. Fira and L. Goras: Inter. J. Adv. Comput. Sci. App. **19** (2019) 9 326. <https://doi.org/10.3390/bios11050161>
- 27 M. Fira, H.-N. Costin, and L. Goras: Biosensors **11** (2021) 1. <https://doi.org/10.3390/bios11050161>
- 28 O. AlShorman, F. Alkhatni, M. Masadeh, M. Irfan, A. Glowacz, F. Althobiani, J. Kozik, and W. Glowacz: Adv. Mech. Eng. **13** (2021) 1. <https://doi.org/10.1177/1687814021996915>
- 29 Z. Liu, X. Wang, and L. Zhang: IEEE Trans. Instru. and Meas. **69** (2020) 6630. <https://doi.org/10.1109/TIM.2020.2969062>
- 30 A. Kumar, S. A. Kumar, V. Dutt, A. K. Dubey, and V. García-Díaz: Biomed. Signal Proc. Control **76** (2022) 103638. <https://doi.org/10.1016/j.bspc.2022.103638>
- 31 S. S. Joseph and A. Dennisan: Comput. Methods Biomech. Biomed. Eng.: Imaging Visualization **11** (2022) 3796. <https://doi.org/10.1080/21681163.2022.2113436>
- 32 F. R. Bach and M. I. Jordan: [https://www.researchgate.net/publication/228351694\\_A\\_probabilistic\\_interpretation\\_of\\_canonical\\_correlation\\_analysis](https://www.researchgate.net/publication/228351694_A_probabilistic_interpretation_of_canonical_correlation_analysis) (accessed May 2023).
- 33 R. Kohavi and G. H. John: Artif. Intelli. **97** (1997) 1. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)
- 34 <https://drive.google.com/drive/folders/1YsdgGDkLkXmCiYX6SFqKwYxuOhW3C4y> (accessed May 2023).

## About the Authors



**Joy Iong-Zong Chen** received his B.S. degree in electronics engineering from National Taiwan Technical University, Taipei, Taiwan, and M.Sc. degree in electrical engineering from Da-Yeh University, Chung-Hua, Taiwan, in 1985 and 1995, respectively, and Ph.D. degree in electrical engineering from National Defense University, Tao-Yuan, Taiwan, in 2001. He is currently a full professor at the Department of Communication Engineering, Da-Yeh University in Changhua, Taiwan. Prior to joining Da-Yeh University, he worked at the Control Data Company (Taiwan) as a technical manager from Sep. 1985 to Sep. 1996. His research interests include wireless communications, ML techniques, robot systems, and wireless sensor networks.

([jchen@mail.dyu.edu.tw](mailto:jchen@mail.dyu.edu.tw))



**Wen-Chueh Lo** is currently a technical staff member of the Taiwan Battery Manager Company. He obtained his M.S. degree from the Department of Electrical Engineering of Chieh-Kuo University of Technical Institute and is now pursuing a Ph.D. degree. His research interests include AI, machine learning, and battery power management. ([lowctwan@gmail.com](mailto:lowctwan@gmail.com))