

Real-time anomaly detection in electric motor operation noise

Van-Khanh Nguyen¹, Bao-Toan Thai¹, Vy-Khang Tran¹, Hai Pham^{1,2}, Chi-Ngon Nguyen¹

¹Faculty of Automation Engineering, College of Engineering, Can Tho University, Can Tho, Vietnam

²Aerospace Engineering and Aviation Discipline, School of Engineering, RMIT University, Melbourne, Australia

Article Info

Article history:

Received Feb 1, 2024

Revised Mar 4, 2024

Accepted Mar 9, 2024

Keywords:

Anomaly detection

Autoencoder

Modbus transmission control protocol

Programmable logic controller

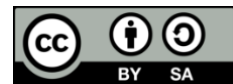
Real-time operating system

Spectrogram

ABSTRACT

Anomaly detection plays a very important role in many fields to identify abnormalities occurring in the system earlier. This study proposes a new abnormality detection solution for 3-phase electric motors based on their working noise. Normal and abnormal operating noise data sets for an electric motor were acquired in the laboratory. These datasets are converted into the corresponding two-dimensional gray spectrogram image sets. The normal set is used to train the autoencoder (AE) model to find the abnormality evaluation threshold. This threshold is validated again with anomalous data sets. The trained AE is then quantized to be installed on a system consisting of two duo-core microcontroller units (MCUs) for real-time testing. Free real-time operating system (FreeRTOS), a real-time operating system, is used to schedule tasks on the system. Experimental results show that the designed anomaly detector can accurately detect over 99% of abnormal events. The system can communicate with a supervisory control and data acquisition (SCADA) application running on the S7-1200 programmable logic controller (PLC) platform using the Modbus transmission control protocol (TCP) protocol. The SCADA application can continuously record evaluated results from the system and adjust abnormal thresholds for the system directly on the human-machine interface (HMI) screen.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Van-Khanh Nguyen

Faculty of Automation Engineering, College of Engineering, Can Tho University

Campus II, 3/2 street, Ninh Kieu district, Can Tho city, Vietnam

Email: vankhanh@ctu.edu.vn

1. INTRODUCTION

Anomaly detection (AD) is an important task in many fields, as it can detect potential problems so that they can be resolved in a timely manner. In the industrial sector, AD has been applied to electric motors based on analysis of working temperature [1], electric current [2], electric current and vibration [3]; heating plant [4]; rotating machinery [5]; industrial furnace [6]; industrial products [7]; elevator hydraulic power units [8]; water tank's level [9]; wind turbine [10]; and cyber-attacks against industrial control systems (ICS) [11]. In addition, it is also applied to detect abnormalities on satellite telemetry data [12]; transportation systems [13]; human health is based on electrocardiogram (ECG) [14]–[19]; and heart rate [20].

Many algorithms have been applied to detect anomalies, which can be classified into two main groups. One is based on statistical analysis to find the outer line of the normal signal so that abnormal events can be detected [2], [3]. While the other one is the application of artificial intelligence solutions such as autoencoder (AE) models [1], [8], [14], deep AE [9], [11], long short-term memory (LSTM)-AE [6], [8], [12], adversarial AE [17], convolution AE [5], [10], [21], transformer block [15], a lightweight neural network (ANNet) [16], inductive transformer (ITran) [7]. Furthermore, the AD model incorporates the data mining rule algorithm [18] or alternatively, a synthesis of principal component analysis (PCA) and machine

learning methodologies [19]. It can be seen that the application of AE models and their variations in AD is the most common. AE is trained unsupervised using the normal data set, so applying it to real-life applications is very feasible, especially when integrating into devices on internet of thing (IoT) systems.

Currently, most AD algorithms only stop at evaluating feasibility and do not focus on installation, therefore they can run on microcontroller units (MCUs) to increase the possibility of practical deployment. Some studies have installed AD functions on MCUs [16], [18] and have only applied them to ECG but not for the AE model. Therefore, researching the practical implementation of AD with AE on MCU will contribute to opening up more opportunities to deploy AD in practice, especially in IoT applications. Although to implement AE on MCUs, it is necessary to integrate on-device training (ODT) capabilities, on which tiny machine learning (TinyML) has been widely used. Indeed, TinyML can be applied to implement the AE model and pre-train it with normal data first before integrating into the system. This solution requires acquiring data and retraining the AE model when the monitoring object changes. However, AE is easy to perform and very feasible for monitored objects with normal conditions that do not fluctuate over time.

This study designs a system to detect working noise abnormalities in 3-phase electric motors using AE operating on MCUs that can be integrated into industrial applications. AD was designed and trained on a computer platform before being quantized to operate on an MCU under the Tensorflow lite framework. The training data is the noise in the normal working state of a 3-phase electric motor. The quantized AE model will be implemented on the proposed hardware system, including two dual-core MCUs to execute tasks such as acquiring and pre-processing noise; applying the AE to evaluate noise; and performing the Modbus transmission control protocol (TCP) protocol to send the evaluation results to the programmable logic controller (PLC) or supervisory control and data acquisition (SCADA) application via Modbus TCP protocol. As a result, the outcomes of this study can be deployed to integrate AD into industrial systems as well as the IoT.

2. MATERIALS AND METHODS

2.1. Materials

This study is based on the operating noise data of three-phase motors acquired by [22]. However, to increase the real-time running efficiency of the system, this dataset is not directly used to train and test the AE. They are treated as an actual noise source from a working electric motor and are played from a wireless speaker. The system developed in this study will acquire normal and anomaly data sets. The time-series data will then be converted into two-dimension (2-D) gray-scale spectrogram image right on the system to be used for AE training and testing. The experimental setup is presented in Figure 1. The center of the experiment is the AD box, which detects anomalies in the motor's working noise. The acquired temporal data and their spectrograms are saved on a secure digital (SD) card for AE training and testing.

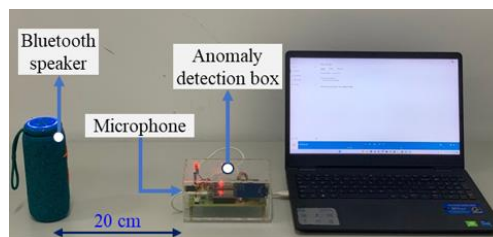


Figure 1. Data acquisition setup

Besides the data obtained from the experimental setup, an additional anomalous data set was also created by adding gaussian noise to the normal data. This dataset is used to evaluate trained AE because the anomaly dataset is not available in practice. All data will be converted to 2-D spectrogram images based on fast fourier transform conversion and the Mel filter; the conversion algorithm has been presented [22]. Figure 2 illustrates the samples in these datasets, which include normal and abnormal datasets. Figure 2(a) shows normal noise; for abnormal data sets such as normal noise added the Gaussian noise, phase loss, phase shift, and bearing beaking, corresponding to Figures 2(b) to 2(e). Each spectrogram corresponds to a one-second data segment in the time domain. Spectrogram imaging is performed right on the MCU to ensure that the developed system will operate in real-time most accurately. A summary of data sets after acquisition and processing is presented in Table 1.

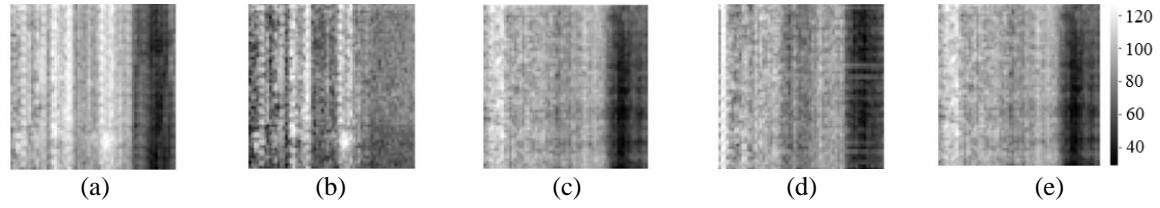


Figure 2. Samples spectrogram of electric motor's operating noise: (a) normal noise; anomalous noise including (b) normal noise added Gaussian noise, operating noise in the cases of (c) phase loss, (d) phase shift, and (e) bearing beaking

Table 1. Datasets

Data set name	Number of images	Split datasets		Test	
		Train (PC)	Validate (PC)	PC (image)	MCU (Time series)
<i>nDataset</i>	4,200	2,688	672	840	1,000s
<i>aDataset1</i>	4,200	-	-	4,200	1,000s
<i>aDataset2</i>	12,600	-	-	12,600	1,000s

2.2. System overview for real-time autoencoder-based anomaly detection

An overview of the implementation of the real-time AD system on the MCU is illustrated in Figure 3. This process includes two main stages. The first stage is data acquisition and training. During this stage, the normal data will be acquired and stored to the SD memory card. Normally, anomalous data will not be acquired at this stage because it rarely exists. In this study, an anomalous dataset purposely generated in the laboratory [23] is used to evaluate the system. These data are then converted into 2-D gray-scale spectrogram image datasets. They will then be transferred to the computer to train and test the AE set to statistically find the abnormal assessment threshold. The trained AE model will then be quantized to match operation on the MCU under the Tensorflow lite platform. After quantization, the trained AE will be converted into a 2-D array for integration into the MCU project. The second stage is to integrate AE into the MCU, program some additional algorithms such as acquiring new data, creating spectral images, calculating mean squared error (MSE) between the input image and the reconstructed image after encoding by the AE, and evaluating whether the new data is normal or anomalous based on the quantized threshold. On embedded systems, Modbus TCP is also implemented to communicate with industrial systems to increase the practical applicability of the research.

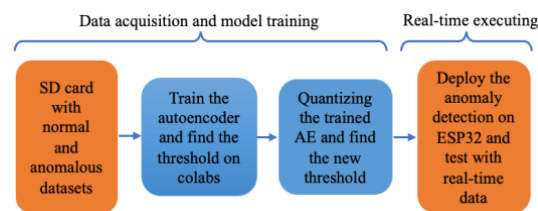


Figure 3. Process of performing AD in the MCU

2.3. Implementation of real-time autoencoder-based anomaly detection

Spectrograms of noise when the engine is operating normally will be used to train an AE model. The MSE between the encoded image and the input image will be calculated to find the abnormality threshold. The spectrum image used is a 64×64 grayscale image. To reduce the AE size, a standard 64-input AE is trained to encode each row of the image and repeated 64 times to encode the entire image.

The structure of AE is shown in Figure 4. Suppose the input image's row vector is x , the hidden layer is h , the reconstructed vector is \hat{x} , and the dataset has n vectors. Input vector $x_i^t \in R^n$, $x_i^t = [x_1^t, x_2^t, \dots, x_n^t]$ with $t = 1, 2, \dots, n$ is mapped into hidden layer $h \in R^m$, $h = [h_1, h_2, \dots, h_m]$, $m < n$. The output of the p^{th} neural of the hidden layer is calculated in (1):

$$h_p = af(\sum_{j=1}^n w_{pj}^E x_j^t + b_p^E) \quad (1)$$

Where af is the activation function, w_{pj}^E is the weight between input j^{th} and the p^{th} neural of hidden layer, and b_p^E is the bias of the p^{th} neural of the hidden layer. The hidden layer vector maps into the recovery vector $\hat{x}_t \in R^n$, $\hat{x}_t = [\hat{x}_1^t, \hat{x}_2^t, \dots, \hat{x}_n^t]$ the same size as the input. The output of the i^{th} neural of the output layer is calculated by (2):

$$\hat{x}_i^t = af(\sum_{j=1}^n w_{pi}^D h_p + b_p^D) \quad (2)$$

In this study, the AE will be trained using spectrogram gray spectrum images of normal noise, loss function using MSE, Adam optimizer, and a learning rate of 0.001. The AE is trained using spectrograms of the normal data set acquired and completely transformed using the MCU. Normal data sets are usually the easiest to acquire on most operating systems. After training, the MSE reconstruction between the input and output of this data set will be calculated using (3) to determine the evaluation threshold.

$$MSE = \frac{1}{N+M} \sum_{i=1}^N \sum_{j=1}^M (x_{ej}^i - x_{dj}^i)^2 \quad (3)$$

where x_e and x_d are the input and reconstruction, respectively, and i and j are the row and column numbers of the spectrogram, respectively.

The evaluation threshold is determined based on the gamma distribution function of the MSE of the normal dataset that was used to train the AE. The gamma distribution is the two-parameter distribution for which the density function is given in (4):

$$f(x) = \begin{cases} \frac{x^{\alpha-1} e^{-\frac{x}{\theta}}}{\Gamma(\alpha) \theta^\alpha}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\alpha > 0$ and $\theta > 0$ are shape and scale parameters, respectively. The function $\Gamma(\alpha)$ is defined as (5):

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx \quad (5)$$

The gamma distribution was estimated using the maximum likelihood estimation (MLE) method. This method estimates the two parameters of the gamma distribution by approximating the solution of the likelihood equation. The likelihood equation for the gamma distribution as in (6):

$$L(\alpha, \theta) = n[(\alpha - 1) \overline{\ln x} - \frac{\bar{x}}{\theta} - k \ln \theta - \ln \Gamma(\alpha)] \quad (6)$$

Where n is the number of data points (MSE between input and reconstruction) and x is the number of data points, donate $X = [x_1, x_2, \dots, x_n]$. The maximum value of $L(\alpha, \theta)$ occurs at and according to [24] the value of α is approximately determined by (7). In this study, the gamma probability distribution is fitted using the *gamma.fit* function in the scipy library [25] using the MLE method. After determining the gamma probability distribution, events with a probability of occurrence less than or equal to a given threshold will be considered abnormal.

$$\hat{\alpha} \approx \frac{0.5}{\log \bar{x} - \log x} \quad (7)$$

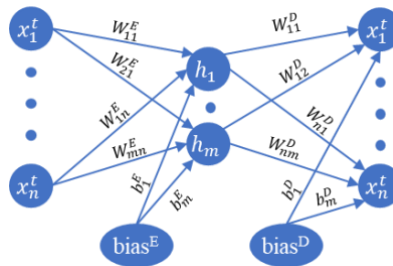


Figure 4. AE model structure

The training process and determining the MSE threshold are performed on Google Colab. To be able to perform AE on the MCU, the trained AE will be quantized to reduce model size, and the data will also be represented as a signed 8-bit integer to be suitable for MCUs with slow computing speed [26]. The original spectrogram image data of type uint8 is also converted to type int8 to conform to the quantized model. The quantized AE model is then used to further calculate the MSE between input and reconstruction to find a new evaluation threshold. Finally, this AE model will be converted to a 2-D constant array for integration into the embedded system for real-time testing with new noise data. The evaluation threshold will then be verified with two abnormal data sets, *aDataset1* and *aDataset2*. However, for practical applications, the *aDataset1* should be more suitable because it is difficult to obtain abnormal data of working systems.

The flowchart of firmware on MCUs is illustrated in Figure 5. The system runs on a two dual-core ESP32 MCUs platform proposed [22] to utilize the maximum computing capacity. Two MCUs communicate via the universal asynchronous receiver/transmitter (UART) port to transmit the spectrogram image. The flowchart delineates the comprehensive algorithm employed by the AD system. However, the detail communication protocol utilized by the two MCUs are not included in this representation. On the system, MCU1 has two main roles: buffering data noise, reading data, and creating spectrogram images for MCU2 to evaluate. The noise is acquired by the inter-IC sound (I2S) module to reduces the load on the MCU [27]. The sampling rate is 16 kHz. MCU2 also plays two main roles: executing the AE to calculate the MSE between input and reconstruction to determine the normal or anomalous based on the evaluated threshold, and running the Modbus TCP engine to communicate with industrial applications. MCUs' firmware is established based on the free real-time operating system (FreeRTOS) real-time operating system platform [28], [29].

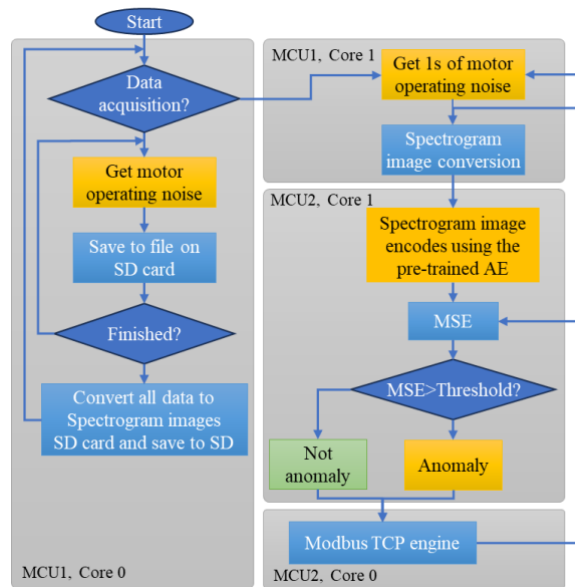


Figure 5. Flowchart of MCUs' program

To reduce the size of the AE model, the study uses AE to encode and calculate the MSE of each row of a 2-D spectrogram image using (3). Therefore, to completely encode a spectral image, the process is repeated a row of times, then averaging the MSE of the row to get the MSE of whole image. Details of the algorithm are presented in Algorithm 1.

Algorithm 1: Calculating MSE between input and reconstruction of a spectrogram image

```

Let img_buffer ← be the array containing input data
Let model_input_buffer, mse ← be arrays for storing intermediate calculations
Let output ← be the object holding the model output
Let interpreter ← be the TensorFlow Lite interpreter object
mse_total ← 0;
For row ← 0 to 63:
  For col ← 0 to 63:
    model_input_buffer[col] = int8_t(uint8_t(img_buffer[row*64+col]) - 128);
  End for

```

```

TfLiteStatus invoke_status = interpreter->Invoke();
If invoke_status != kTfLiteOk:
    error_reporter = Report("Invoke failed");
    return;
End If
mse[row] ← 0;
For col ← 0 to 63:
    model_input_buffer[col] = int8_t(uint8_t(img_buffer[row*64+col])-128);
    tmp_mse=pow(model_input_buffer[col]-output.data.int8[col], 2);
    mse[row] = mse[row] + tmp_mse;
End for
mse[row] = mse[row] / 64;
mse_total += mse[row];
End for
mse_total / = 64;

```

2.4. Modbus TCP protocol

Modbus TCP is an open de facto standard byte-oriented industrial communication protocol for data interchange between embedded systems, devices, and industrial applications [30]. Such a lightweight protocol for polling industrial equipment that behaves as servers could be inexpensively implemented and beneficial to devices functioning as clients. The server responds to the client's inquiries with a frame of bytes that either contains sensor measurement data or verifies that commands were carried out.

In this study, the server (MCU) responds to a request from the S7-1200 client in the same format as the read registers using the function code (FC) “internal and physical output registers” (FC=0x03), or executes the request write internal registers using the “write multiple registers” FC (FC=0x10). The server will be installed on the AD device to provide the MSE value of the current data sample to the client, a PLC S7-1200. With pre-determined threshold settings, the client can provide information about the current status of the monitored object right on the HMI interface. Updating the MSE value in the data register is also controlled by the client through the value of the data register. The data registers used are summarized in Table 2. The registers can be used to extend the functionality of the anomaly detector in the future. The Modbus TCP protocol on the MCU uses an open source library [31], available for free under the GNU GPL license. This library has been integrated into the system and tested in communication with the S7-1200 before officially being used for research. On the S7-1200, Modbus TCP uses built-in command blocks.

Table 2. Details of the addresses of Modbus TCP registers used

Register	Address	Description
Mb.MbData[1]	4001	MSE value bytes.
Mb.MbData[3]	4003	Enable MSE updates
Mb.MbData[4]	4004	Set the anomaly threshold

2.5. Performance evaluation

In (8)-(11) are used to evaluate the effectiveness of the implemented AD techniques in terms of accuracy (*ACC*), recall (*R*), precision (*P*), and score (*F1*) metrics. Their definitions are as follows: *ACC* is defined as the proportion of true positives and true negatives to all samples; *P* is defined as the proportion of the true positives to samples that are detected to be positives; *R* is defined as the proportion of the true positives to samples that are actually positives; *F1* score is defined as a combination of precision and recall.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (8)$$

$$P = \frac{TP}{TP+FP} \times 100 \quad (9)$$

$$R = \frac{TP}{(TP+FN)} \times 100 \quad (10)$$

$$F1 = \frac{(2 \times P \times R)}{(P+R)} \times 100 \quad (11)$$

Where TP is the number of cases where the spectrogram image of anomalous noise is evaluated accurately as anomaly; TN is the number of cases where the spectrogram image of normal noise is evaluated accurately as normal; FP is the number of cases where the spectrogram image of normal noise is evaluated wrongly as

abnormal, and FN is the number of cases where the abnormal spectrum image is judged wrongly to be normal.

3. RESULTS AND DISCUSSIONS

3.1. Autoencoder training

For the AD problem, the AE set will be trained with the normal dataset *nDataset*. In this study, 80% of the normal spectral image set, or 3360 images, were used for training. This set is further divided into two parts, with 80% for the train dataset and 20% for the validate dataset during the training process. Figure 6 presents the results of AE training on colab. After only two epochs, the AE's MSE has converged. The trained AE is used to calculate the MSE between the input image and the reconstruction image of the normal spectral image set to determine the anomalous threshold. This threshold is determined based on the gamma distribution of the MSE of the normal spectral image set.

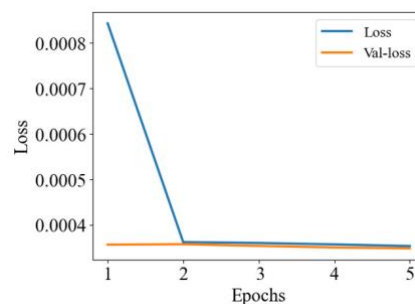


Figure 6. The AE training process

To evaluate the ability to detect anomalies with a defined MSE threshold, two anomaly datasets, *nDataset1* and *nDataset2* described in Table 2, have been applied. However, in reality, data set *nDataset1* is more suitable because the objects' normal data is easier to acquire. The histogram distribution of the two cases is presented in Figure 7, for which Figures 7(a) and 7(b) present the MSE distribution for data sets *nDataset1* and *nDataset2*, respectively. In this study, spectral images with a probability of occurrence less than 0.01% will be considered abnormal. From the gamma distribution, the threshold is determined to be 0.0006. The MSE distribution results of the two abnormal sets show that the trained AE set has been trained very effectively, and the MSE of the abnormal data is clearly larger than that of the normal.

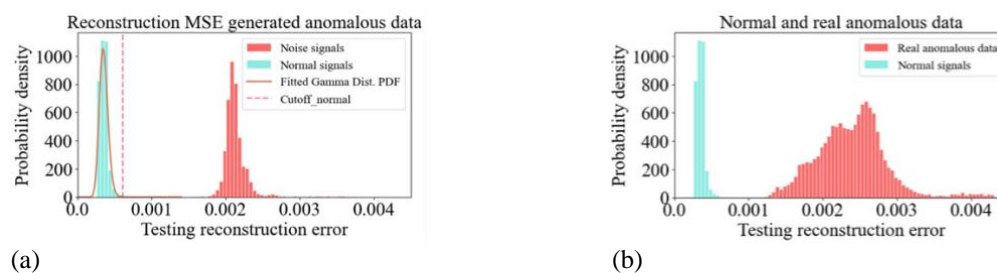


Figure 7. The MSE distribution of normal (a) *aDataset1* and (b) *aDataset2*

To be able to use trained AE on the MCU, it has to be quantized. Quantization is the process of converting the model to an 8-bit representation to be suitable for MCUs with limited resources. In this study, anomaly evaluation is performed on both original and quantized models to evaluate and compare the accuracy of the quantized model with the original one. Figure 8 presents the MSE distribution for the quantized data sets *aDataset1* and *aDataset2* which are also presented in Figures 8(a) and 8(b), respectively. Spectral images with a probability of occurrence less than 0.025% will be considered abnormal; the threshold is 99 derived from the gamma distribution.

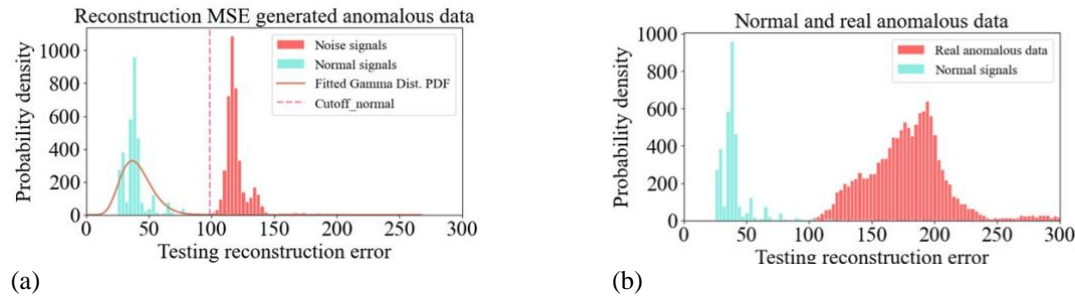


Figure 8. The MSE distribution is normal after being quantized and (a) *aDataset1* and (b) *aDataset2*

With the determined threshold, the confusion matrix of quantized AE is plotted in Figure 9, evaluated with the same testing set as the original AE, including *nDataset*, *aDataset1*, and *aDataset2*, corresponding to Figures 9(a) and 9(b). With the defined thresholds, *ACC*, *P*, *R*, and *F1* values are also calculated with trained AE before and after quantization. The results are presented in Table 3. The results show that the AE is very effective in encoding spectrograms of normal data. The system is capable of identifying the anomalous spectrogram with 100% accuracy. With trained AE's post-quantization model, the anomaly recognition accuracy decreased by a very small amount, still above 99.6%. The unsuccessful AD rate is very low.

Thus, an anomaly detector in the motor working noise based on the AE model has been successfully designed. With the evaluated threshold, the system can detect anomalies with over 99.6% accuracy for both pre- and post-quantization AE models. This quantized model will be converted to the constant array for further integration and real-time evaluation on the ESP32 MCU.

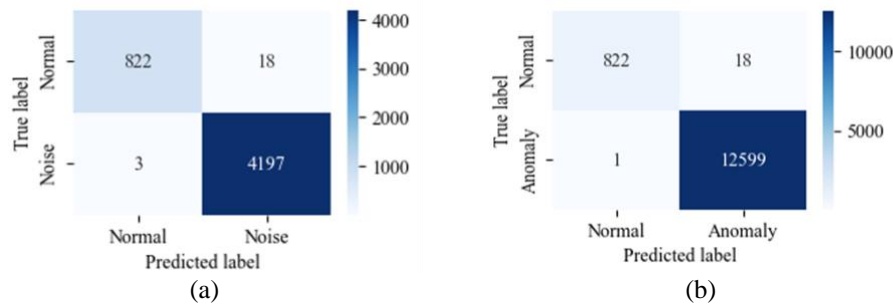


Figure 9. Confusion matrix (a) *aDataset* and *nDataset1* and (b) *aDataset* and *nDataset2*

Table 3. *ACC*, *P*, *R*, and *F1* values of the trained AE before and after quantizing

Trained AE	Normal dataset	Anomalous dataset	<i>R</i>	<i>P</i>	<i>F1</i>	<i>ACC</i>
Before quantizing	<i>nDataset</i>	<i>aDataset1</i>	100.0	99.95	99.98	99.96
	<i>nDataset</i>	<i>aDataset2</i>	100.0	99.95	99.99	99.99
After quantizing	<i>nDataset</i>	<i>aDataset1</i>	99.93	99.57	99.75	99.58
	<i>nDataset</i>	<i>aDataset2</i>	99.99	99.86	99.92	99.86

3.2. Real-time testing autoencoder-based anomaly detection on embedded system

The experimental setup for practical evaluation of the anomaly detector is illustrated in Figure 10. In which Figure 10(a) shows the real-time experimental setup, Figure 10(b) shows inside the designed AD box. The recording microphone on the abnormal detection box is placed opposite a speaker, which simulates the noise source from the electric motor. A laptop is used to play the recorded noise to evaluate the system. All types of noise will be played to evaluate the anomaly detector. Each type of noise will be played for 1,000 seconds, corresponding to an anomaly rating of 1,000 times. PLC S7-1200 and HMI screen are used to evaluate the ability to communicate with industrial systems via the Modbus TCP protocol. The hardware of the AD box is based on the proposed platform of the study [22]. The main components are two dual-core ESP32 MCUs; one plays the role of capturing the motor's working noise and converting it into a spectral image; while the other receives the spectral image to calculate the MSE between the input spectral image and

its reconstruction through an AE. The evaluation result is displayed on two LEDs for users' observation. The memory card is used to store the acquired and processed normal data for training when AE needs to be retrained. The difference in this research is that the W5500 Ethernet module is utilized to support the local area network (LAN) connection. With Modbus TCP, the AD system can communicate directly with industrial systems easily. A SCADA application is intended to illustrate the system's ability to connect to industrial equipment. The AD box operates directly with the alternating current (AC) power through an AD/DC Hi-Link device that creates a stable 5 V voltage for the system.

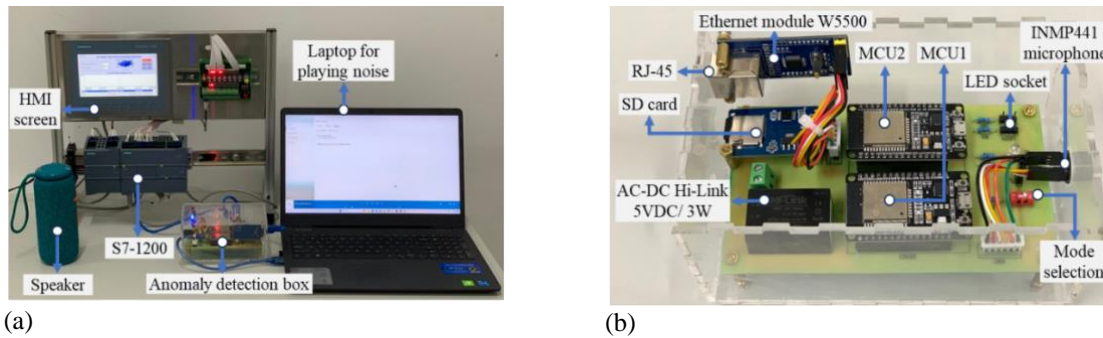


Figure 10. Real-time testing setup: (a) whole system and (b) inside the designed AD box

The hardware system has increased processing speed by pairing two multi-core MCUs. Indeed, on each MCU, the freeRTOS operating system is scheduled so that two main tasks will run in parallel. On MCU1, core 0 is responsible for capturing the noise audio segment every second from the I2S module stream; core 1 will wait for the segment to be converted into a spectrogram to send to MCU2, while the other core is recording the next audio segment. Similarly, on MCU2, core 0 waits for the spectrum image from MCU1 to calculate MSE based on trained AE to evaluate the current noise, and core 1 runs the Modbus TCP engine to exchange information with the SCADA. Tasks exchange messages with each other through a simple global variable mechanism. The timing diagram of the tasks on the system is illustrated in Figure 11. Parallelization of tasks has enabled the system to continuously evaluate new data and print results every second. Table 4 depicts that the total time from the start of recording until printing the results is 1,299 seconds. Results will be printed continuously every second.

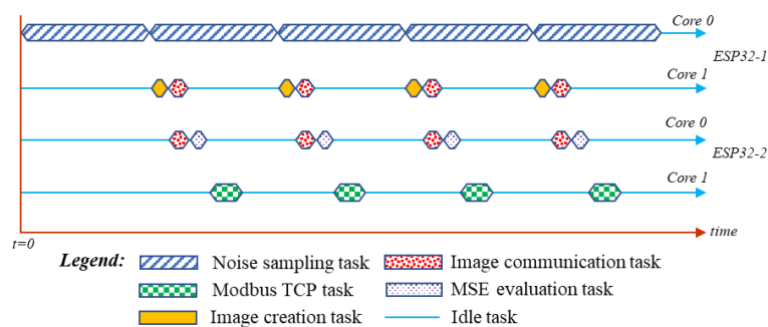


Figure 11. Time diagram for executing tasks on the system

Table 4. Time to perform tasks on the system

Task	Execute time (ms)	MCU/Core
Noise sampling	1000	1/0
Spectrogram creation	86	1/1
Spectrogram transmission	157	1/1
Spectrogram receipt	155	2/1
MSE evaluation	56	2/1
Modbus TCP	NA	2/0

The real-time evaluation results are presented in Figure 12. Figure 12(a) and Figure 12(b) present the probability distribution of the MSE of the two experiments. Compared with the experimental results with the post-quantization model on the computer, the MSE distribution of normal data and abnormal data has changed. However, with the post-quantization anomaly threshold, the system can still accurately detect abnormal working noise sequences. Figure 12(c) and Figure 12(d) present the confusion matrices of the two experiments. The results show that the autoencoder-based anomaly detection (AAD) system running on the ESP32 can accurately identify 100% of abnormal sound sequences, only confusing a normal sample into an abnormal one. Thus, when operating with real-time simulated noise, the designed AAD system can accurately detect over 99.9% of abnormalities occurring on electric motor based on its working noise shown in Table 5.

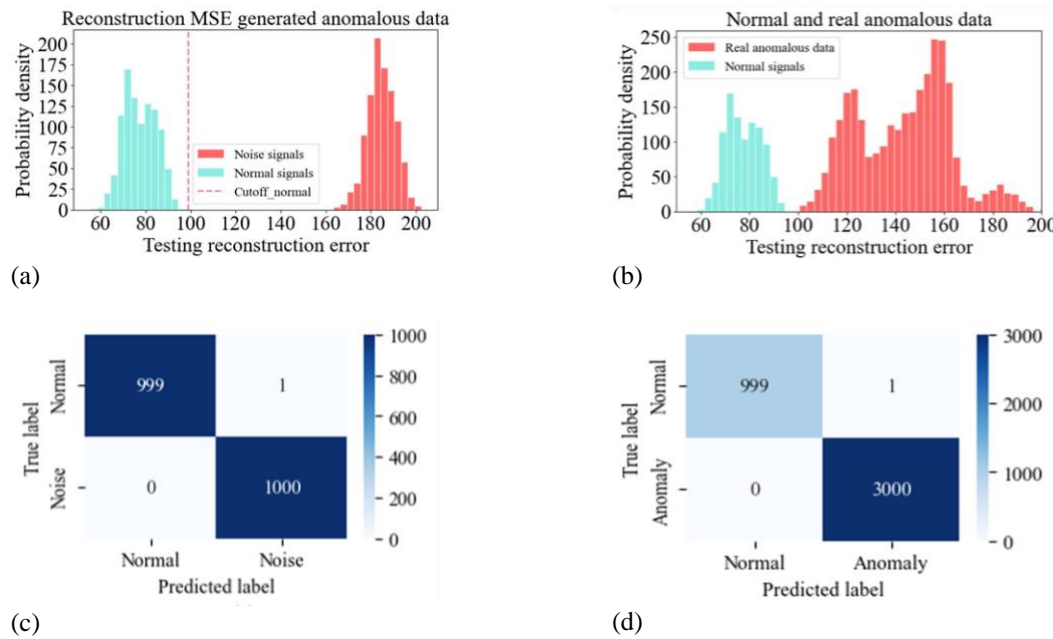


Figure 12. Real-time testing results: (a) normal and generated anomalous data, (b) normal and real anomalous data, (c) confusion matrix of generated anomalous data, and (d) confusion matrix of real anomalous data

Table 5. ACC, P, R, and F1 values of the real-time test

Normal noise source	Anomalous noise source	<i>R</i>	<i>P</i>	<i>F1</i>	<i>ACC</i>
nDataset	aDataset1	100.0	99.97	99.98	99.96
nDataset	aDataset2	100.0	99.90	99.95	99.95

3.3. Autoencoder-based anomaly detection testing with Siemens S7-1200 controller

The real-time communication between the designed AAD system and PLC S7-1200 is presented in Figure 10(a). A SCADA application designed for the S7-1200 PLC mainly communicates with the AD box. This application can monitor the motor status (normal (blue symbol), abnormal (red symbol), and idle (gray symbol)), set the abnormal threshold, display the current MSE value, and show 50 most recent MSE historical samples. The testing scenario is a series of normal and abnormal data being emitted to test the ability to communicate with the industrial system, including 10 seconds of normal data segments alternated by 3 seconds of anomaly data segments (i.e., phase loss, phase shift, and bearing breakage data). So, the total duration of the testing noise is 49 seconds.

The experimental result captured from the HMI screen is illustrated in Figure 13. According to the diagram in Figure 11, every second, the evaluation result will be sent by the AD system to S7-1200 to update information on the HMI screen. Thus, after 49 seconds, there will be 49 MSE results displayed on the trend chart. With the adjusted threshold set to 99, the system successfully detected all abnormal events as in the testing scenario. Figure 13 depicts the system status when identifying the 49th noise segment: the MSE is 83.7, and the motor is in a normal state (blue symbol).

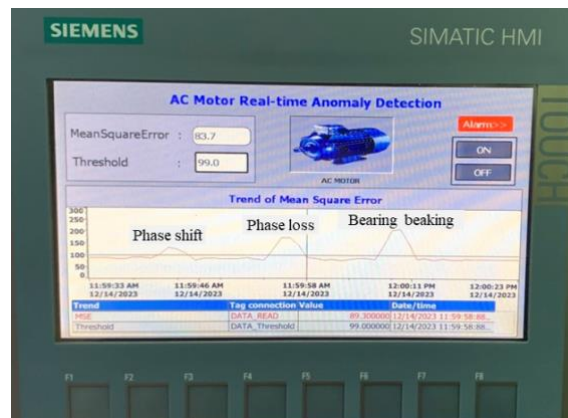


Figure 13. Real-time testing result on the HMI screen

3.4. Discussion

The research has successfully applied the AE to detect abnormalities in the working noise of 3-phase electric motors. The designed anomaly detector has been quantized and runs in real-time on a multi-microcontroller system. On each microcontroller, freeRTOS has also been applied to parallelize system tasks. The system has also integrated the ability to sample audio signals using the built-in I2S module of ESP32 MCU to improve the system's working efficiency. In addition, the open-source Modbus TCP engine has also been integrated to support connectivity with industrial systems. Experimental results show that the system has been designed successfully, achieving high accuracy with the evaluated threshold and communicating well with the SCADA application.

This application can be applied to real electric motors, but it is necessary to deploy data acquisition mode to retrain the system before running actual monitoring. In addition, because it relies on sound, it is likely to be affected by unwanted surrounding sound events, so further evaluation is needed under real conditions. The method in this study can also be applied to other types of timing signals, such as voltage, current, vibration, point temperature, and thermal imaging. Furthermore, the monitoring object of this research does not only use in electric motors but can also be applied to many other objects.

Although it promises quite good applicability, the actual implementation is still not very convenient. Indeed, systems based on TinyML or AI networks are trained on the personal computer before being deployed on embedded devices. This leads to a difficulty: when applying to a new object, you have to re-acquire data, retrain, and redeploy the AE on the MCU. These procedures are quite time-consuming, leading to difficulties in applying and upgrading the system later. An effective unsupervised training mechanism for the AE model should be researched and integrated in the future. This will make system deployment simpler and more flexible because the normal data of the potential objects is not difficult to acquire.

Furthermore, the proposed system has a lot of application expansion in the future. Indeed, besides integrating into industrial systems based on the Modbus TCP protocol, the system is also capable of supporting internet communication via the LAN or WiFi module of the ESP32. This enables the system to be easily integrated into the IoT system, expanding the scope of monitoring to many objects. From there, it further increases the practical significance of the research.

4. CONCLUSION

Abnormalities occurring on 3-phase electric motors have been detected very effectively based on spectrogram images of the working noise. Research has suggested an interesting approach to system design: data is acquired and processed directly by the device instead of using specialized equipment. This approach has proven effectiveness when deploying real-time systems. The classic AE suite has been used to encode the spectral images of the noise when the motor is operating normally. After training, the spectrogram images of abnormal cases will create large errors between the input and output of the AE, helping the system detect the occurrence of abnormal events. The study successfully trained AE and demonstrated that the quantized AE model had very little decrease in accuracy. The anomaly threshold is effectively determined automatically based on the gamma distribution of the MSE. The designed system has identified abnormalities in electric motors with an accuracy of over 99.6% in both test cases with real-time collected and tested data. This is quite an impressive result, and it promises the ability to practically deploy the system. In addition, an




embedded Modbus TCP engine has also been successfully integrated so that the system can connect to industrial applications, increasing the applicability of the system. Furthermore, the computing resources of MCUs are still very large, so they have the ability to expand to support additional connections with IoT platforms. From there, the system can be fully applied to develop distributed anomaly monitoring systems.

REFERENCES




- [1] E. H. Demircioğlu and E. Yılmaz, "A method based on an autoencoder for anomaly detection in DC motor body temperature," *Applied Sciences*, vol. 13, no. 15, pp. 1–11, Jul. 2023, doi: 10.3390/app13158701.
- [2] T. Watanabe, I. Kono, and H. Onozuka, "Anomaly detection methods in turning based on motor data analysis," *Procedia Manufacturing*, vol. 48, pp. 882–893, 2020, doi: 10.1016/j.promfg.2020.05.126.
- [3] L. Magadán, F. J. Suárez, J. C. Granda, and D. F. García, "Real-time monitoring of electric motors for detection of operating anomalies and predictive maintenance," in *Science and Technologies for Smart Cities*, Springer International Publishing, 2020, pp. 301–311, doi: 10.1007/978-3-030-51005-3_25.
- [4] R. Wang, H. Qiu, X. Cheng, and X. Liu, "Anomaly detection with a container-based stream processing framework for industrial internet of things," *Journal of Industrial Information Integration*, vol. 35, pp. 1–17, Oct. 2023, doi: 10.1016/j.jii.2023.100507.
- [5] W. Li, Z. Shang, J. Zhang, M. Gao, and S. Qian, "A novel unsupervised anomaly detection method for rotating machinery based on memory augmented temporal convolutional autoencoder," *Engineering Applications of Artificial Intelligence*, vol. 123, Aug. 2023, doi: 10.1016/j.engappai.2023.106312.
- [6] M. Pota, G. De Pietro, and M. Esposito, "Real-time anomaly detection on time series of industrial furnaces: A comparison of autoencoder architectures," *Engineering Applications of Artificial Intelligence*, vol. 124, pp. 1–20, Sep. 2023, doi: 10.1016/j.engappai.2023.106597.
- [7] X. Cai, R. Xiao, Z. Zeng, P. Gong, and Y. Ni, "ITran: a novel transformer-based approach for industrial anomaly detection and localization," *Engineering Applications of Artificial Intelligence*, vol. 125, Oct. 2023, doi: 10.1016/j.engappai.2023.106677.
- [8] T. Tziolas, K. Papageorgiou, T. Theodosiou, E. Papageorgiou, T. Mastos, and A. Papadopoulos, "Autoencoders for anomaly detection in an industrial multivariate time series dataset," in *Engineering Proceedings*, Jun. 2022, pp. 1–11, doi: 10.3390/engproc2022018023.
- [9] I. T. Nicholas, J. R. Park, K. Jung, J. S. Lee, and D.-K. Kang, "Anomaly detection of water level using deep autoencoder," *Sensors*, vol. 21, no. 19, pp. 1–18, Oct. 2021, doi: 10.3390/s21196679.
- [10] J. Liu, G. Yang, X. Li, Q. Wang, Y. He, and X. Yang, "Wind turbine anomaly detection based on SCADA: a deep autoencoder enhanced by fault instances," *ISA Transactions*, vol. 139, pp. 586–605, Aug. 2023, doi: 10.1016/j.isatra.2023.03.045.
- [11] L. Faramondi, F. Flammini, S. Guarino, and R. Setola, "A hybrid behavior- and Bayesian network-based framework for cyber-physical anomaly detection," *Computers and Electrical Engineering*, vol. 112, Dec. 2023, doi: 10.1016/j.compeleceng.2023.108988.
- [12] Z. Xu, Z. Cheng, Q. Tang, and B. Guo, "An encoder-decoder generative adversarial network-based anomaly detection approach for satellite telemetry data," *Acta Astronautica*, vol. 213, pp. 547–558, Dec. 2023, doi: 10.1016/j.actaastro.2023.09.032.
- [13] J. Ni, W. Chen, J. Tong, H. Wang, and L. Wu, "High-speed anomaly traffic detection based on staged frequency domain features," *Journal of Information Security and Applications*, vol. 77, Sep. 2023, doi: 10.1016/j.jisa.2023.103575.
- [14] A. Jiang *et al.*, "Multi-scale cross-restoration framework for electrocardiogram anomaly detection," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2023*, Springer: Switzerland, 2023, pp. 87–97, doi: 10.1007/978-3-031-43907-0_9.
- [15] A. Alam and A. Artoli, "Unsupervised transformer-based anomaly detection in ECG signals," *Algorithms*, vol. 16, no. 3, pp. 1–15, Mar. 2023, doi: 10.3390/a16030152.
- [16] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, and D. John, "ANNet: A lightweight neural network for ECG anomaly detection in IoT edge sensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 1, pp. 24–35, Feb. 2022, doi: 10.1109/tbcas.2021.3137646.
- [17] L. Shan *et al.*, "Abnormal ECG detection based on an adversarial autoencoder," *Frontiers in Physiology*, vol. 13, pp. 1–14, Sep. 2022, doi: 10.3389/fphys.2022.961724.
- [18] G. Sivapalan, K. K. Nundy, A. James, B. Cardiff, and D. John, "Interpretable rule mining for real-time ECG anomaly detection in IoT edge sensors," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13095–13108, Aug. 2023, doi: 10.1109/jiot.2023.3260722.
- [19] H. Zhou and C. Kan, "Tensor-based ECG anomaly detection toward cardiac monitoring in the internet of health things," *Sensors*, vol. 21, no. 12, pp. 1–17, Jun. 2021, doi: 10.3390/s21124173.
- [20] K. Fujiwara *et al.*, "Heat illness detection with heart rate variability analysis and anomaly detection algorithm," *Biomedical Signal Processing and Control*, vol. 87, pp. 1–11, Jan. 2024, doi: 10.1016/j.bspc.2023.105520.
- [21] P. Xu, H. Gan, H. Fu, and Z. Zhang, "STEAMCODER: Spatial and temporal adaptive dynamic convolution autoencoder for anomaly detection," *Knowledge-Based Systems*, vol. 279, Nov. 2023, doi: 10.1016/j.knosys.2023.110929.
- [22] V. K. Nguyen, V. K. Tran, H. Pham, V. M. Nguyen, H. D. Nguyen, and C. N. Nguyen, "A multi-microcontroller-based hardware for deploying Tiny machine learning model," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5727–5736, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5727-5736.
- [23] V. K. Nguyen, V. K. Tran, M. K. Nguyen, V. T. E. Thach, T. L. H. Pham, and C. N. Nguyen, "Realtime non-invasive fault diagnosis of three-phase induction motor," *Journal of Technical Education Science*, vol. 17, no. 3, pp. 1–11, Oct. 2022, doi: 10.54644/jte.72B.2022.1231.
- [24] T. P. Minka, "Estimating a gamma distribution," *Microsoft Research*, pp. 1–3, 2002.
- [25] "SciPy v1.12.0 manual," *SciPy documentation*, Accessed: Dec. 24, 2023. [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gamma.html>
- [26] "Post-training quantization," *TensorFlow*, Accessed: Dec. 24, 2023. [Online]. Available: https://www.tensorflow.org/lite/performance/post_training_quantization
- [27] ESPRESSIF, "Inter-IC sound (I2S)," *ESP-IDF Programming Guide*, 2024. Accessed: June 28, 2024 [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/i2s.html#introduction>
- [28] N. He and H.-W. Huang, "Use of freeRTOS in teaching real-time embedded systems design course," *Computers in Education Journal*, vol. 5, no. 4, pp. 18–25, 2014, doi: 10.18260/1-2--23240.
- [29] R. Barry, *Mastering the freertosTM real time kernel: A hands-on tutorial guide*, Greater Manchester, United Kingdom: Real Time Engineers Ltd., 2016.
- [30] S. Jaloudi, "Communication protocols of an industrial internet of things environment: a comparative study," *Future Internet*, vol. 11, no. 3, pp. 1–18, Mar. 2019, doi: 10.3390/fi11030066.
- [31] "Modbus library," *My Arduino Projects*, Accessed: Dec. 24, 2023. [Online]. Available: <https://myarduino.projects.com/modbus.html>

BIOGRAPHIES OF AUTHORS






Van-Khanh Nguyen    received his master's degree from Ho Chi Minh University of Technology, Vietnam, in 2014 and his Doctor of Engineering degree from Tokyo University of Marine Science and Technology, Japan, in 2020. Since 2007, he has been a lecturer at the Faculty of Automation Technology, College of Engineering, Can Tho University. Currently, he is the head of PLC Technology and the Industrial IoT Lab. His research interests concentrate on embedded systems and AIoT- and IoT-based applications in environmental and agricultural control, electrocardiogram (ECG) real-time classification, and anomaly detection. He can be contacted at email: vankhanh@ctu.edu.vn.






Bao-Toan Thai    is a B.S. degree student in Automation and Control Engineering of the Faculty of Automation Technology, College of Engineering, Can Tho University, Vietnam. He can be contacted at email: thaibaotoan.ag@gmail.com.






Vy-Khang Tran    is a MSc degree student in Automation and Control Engineering of the Faculty of Automation Technology, College of Engineering, Can Tho University, Vietnam. His research interests focus on embedded systems and AIoT- and IoT-based applications in environmental and agricultural control. He can be contacted at email: tranvykhang1906@gmail.com.



Hai Pham    received his master's degree from University of South Australia (UniSA) in 2010. Since 2012, he has been a lecturer at Faculty of Automation Technology, College of Engineering, Can Tho University. His research interests focus on Bistatic LIDAR system for gas measurement in environmental and agricultural applications. He can be contacted at email: ptlhai@ctu.edu.vn.



Chi-Ngon Nguyen    received B.S. and M.S. degrees in Electronic Engineering from Can Tho University and the National University, Ho Chi Minh City University of Technology, Vietnam, in 1996 and 2001, respectively. The degree of Ph.D. in Control Engineering was awarded by the University of Rostock, Germany, in 2007. Since 1996, he has worked at the Can Tho University. He is an associate professor in automation at Faculty of Automation Technology, and former dean of the College of Engineering at the Can Tho University. Currently, he is a Vice Chairman of the Board of Trustee of Can Tho University. His research interests are intelligent control, medical control, pattern recognition, classifications, speech recognition, computer vision and agricultural automation. He can be contacted at email: ncngon@ctu.edu.vn.