Classification of Vibration in Coal Mining Industry via Deep Neural Network

Jirasak Thothong[1], **Kanok Charoenchaiprakit**[2], **Wekin Piyarat**[1†], and **Kampol Woradit**[3], Non-members

ABSTRACT

Recently, the electricity from coal mining industry is still necessary because it serves as a primary electrical source in many areas in response to high demand for power. However, coal mining can harm the miners, environment, and villages near mining site due to ground vibration from blasts during the operation. Hence, every coal mine industry is required to report the ground vibration for safety purposes. Mostly, the ground vibration data comes from the vibration sensors deployed around the mining site, and the vibration data will be sent to the control room. Due to tons of the vibration data, operators have difficulty in classifying the blast vibrations from the records which causes time-consuming and possible human errors during the process. To solve these problems, this article proposes the Deep Neural Network (DNN) model for a blast vibration classification with 3 hidden layers. the ground vibration data used in training and validating the DNN are collected by the Mae Moh mine site in Thailand. As the result, the designed DNN meets the standard with the accuracy of 100%.

Keywords: Vibration Monitoring System, Deep Neural Network, Classification

1. INTRODUCTION

Coal mining industry can cause fatal injuries to people and the environment around the area. Researches on mine industry are continuously published in various areas such as safety monitoring systems [1-3] and vibration monitoring systems [4]. The ground vibration monitoring system is conventionally a fundamental tool that can prevent the effects of coal mining on humans and environment by observing the blast vibrations during the operation. Because all vibration data from the monitoring system are stored in a single computer in the control room, the operator spends a large amount of time deliberately classifying each of the blast vibrations to analyze the vibration pattern whether it is acceptable according to the standard.

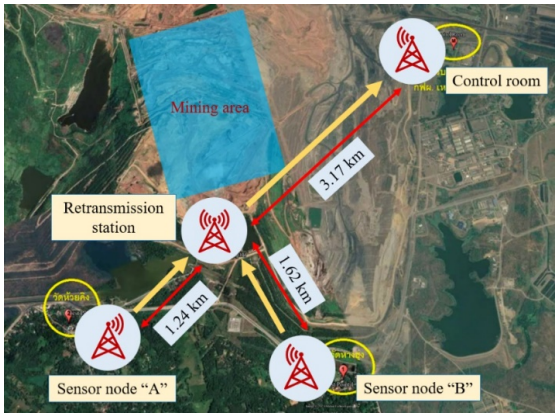


***Fig. 1**: Ground vibration monitoring system in the Mae Moh mine area.*

To overcome this problem, Mae Moh mining area in Thailand, was selected for this research. This mine used to be researched regarding air quality improvement [5], but it has never been studied for to analyze the ground vibrations. Hence, this study focuses on the classification of blast vibrations via the machine learning technique to support the safety standard. To measure the ground vibrations, this mine site has the wireless vibration monitoring systems installed around the mining area.

Fig. 1 shows one part of the ground vibration sensor network consisting of sensor nodes "A" and "B" deployed in the village near the Mea Moh mining area. The vibration data from the sensor nodes will be sent to the control room. Then, the operator will classify the blasting vibration data with naked eyes.

The vibration data from the sensor nodes are presented in 3 dimensions: transverse, vertical, and longitudinal dimension. Fig. 2 shows the blast vibration in the transverse dimension measured by the vibration sensor named "Minimate Plus". The vibration pattern in time domain has the curve that gradually approaches zero from negative values as shown in Fig.2(a). Therefore, it has a low frequency range approximately 0-30 Hz as shown in Fig. 2(b).

Manuscript received on March 9, 2024; revised on March 31, 2024; accepted on April 2, 2024. This paper was recommended by Associate Editor Kampol Woradit.

[1]The authors are with Department of Electrical Engineering, Srinakharinwirot University, Ongkharak Campus, Nakhon Nayok, Thailand.

[2]The author is with Department of Electrical Engineering, Chulachomklao Royal Military Academy, Nakhon Nayok, Thailand.

[3]The author is with Department of Computer Engineering, Chiang Mai University, Huay Kaew Road, Muang District, Chiang Mai, Thailand.

[†]Corresponding author: wekin@g.swu.ac.th

Digital Object Identifier: 10.37936/ecti-eec.2024222.253135

The vibration pattern that does not result from blast is shown in Fig. 3. The vibration in time domain, as shown in Fig.3(a), fluctuates in high amplitude at the beginning approximately 0-2 s, and then has low amplitude around zero. In frequency domain, the amplitude of the vibration gradually decreases to zero from 5-100 Hz as shown in Fig. 3(b).

Conventionally, manpower would be used to classify each vibration for safety purposes. As a result, this process relies on manpower and consumes much time. Nowadays, computing technology is rapidly developed.

Hence, DNN is widely used in classification applications. It can learn and classify data into multiple classes. Thus, this article proposes the application of the DNN to classify any ground vibration, which will serve as a useful source of knowledge for building a future automatic system. The rests of this paper consist of background and relayed works, design system, results and discussion, and conclusion.

2. BACKGROUND AND RELATED WORKS

In this article, bold letters are denoted as vectors and matrices. For example, a vector and matrix can be written as **a** or **A**. Regular letters are denoted as scalars. For example, a scalar can be written as s or A.

2.1 Vibration sensor

Each blast vibration at Mae Moh mine is measured using Minimate Plus produced by Instantel as shown in Fig. 4. This device has a transducer consisting of 3 three geophones to measure the ground vibration in 3 dimensions such as transverse, vertical, and longitudinal dimension [6].

Normally, Minimate Plus is set to send data to the computer in the control room when the ground vibration occurs. Users can utilize the recorded data via the software named "Blastware®". This software can export the raw data to .txt files which are easy to analyse with other programs such as MATLAB and Python IDE (Integrated Development Environment).

In the next subsection, it will explain the architecture and mechanism of the DNN as well as its calculation algorithms. Additionally, knowledge from this subsection will enable readers to understand the details of the designed system topic.

2.2 Deep Neural Networks

Neural Network (NN) is supervised learning which is one kind of the machine learning. Because supervision is required during training, NN needs the correct output data sets corresponding to the input data sets. It is utilized to classify data sets by training weights of the artificial neural branches connecting to the neural nodes. The neural nodes are arranged in columns, and one column of nodes is considered as one layer. Initially, NN necessity has input and output layer. The layer between the input and output layer is call "hidden layer". In addition, DNN is one type of the NN with more than one hidden layer [7].

Fig. 5 shows the DNN structure with L layers. The first layer is the input layer of the DNN $\mathbf{X}^1$, and the last layer is the output layer of the DNN $\mathbf{Y}^L$. The middle layers between the input and output layer of the DNN are the hidden layers. A hidden layer can be output layer of the previous layer and input layer of the next layer. Nodes in the layer are connected to those in next layer by neural branches, and each branch has its own weight parameter $\mathbf{W}$. Thus, the number of weights in the DNN equals to P or $L-1$.

To calculate the output of the DNN, the first data set is fed to the input layer and stored its values in the nodes at the 1^{st} layer. Then, the values of the input layer can be presented as an input matrix which can be expressed as

$$\mathbf{X}^l = \begin{bmatrix} x_1^l \\ \vdots \\ x_{m_l}^l \\ \vdots \\ x_{M_l}^l \end{bmatrix} \tag{1}$$

where $\mathbf{X}^l$ is input matrix at the l^{th} layer, $x_{m_l}^l$ is the m_l^{th} input node, and M_l is the number of nodes in the layer.

The next step is to calculate the weighted sum. The weight between the input layer $\mathbf{X}^l$ and the output layer $\mathbf{Y}^l + 1$ can be expressed as

$$\mathbf{W}^p = \begin{bmatrix} w_{1,1}^p & w_{1,2}^p & w_{1,3}^p & \cdots & w_{1,M_l}^p \\ w_{2,1}^p & w_{2,2}^p & w_{2,3}^p & \cdots & w_{2,M_l}^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{M_{l+1},1}^p & w_{M_{l+1},2}^p & w_{M_{l+1},3}^p & \cdots & w_{M_{l+1},M_l}^p \end{bmatrix} \tag{2}$$

where $\mathbf{W}^p$ is the p^{th} weight matrix, w^p is the weight member of the p^{th} weight matrix, M_{l+1} is the number of the output nodes, and M_l is the number of the input nodes.

The weighted sum is the sum of the multiplication of the weight matrix and input matrix with the bias values which can be expressed as

$$v_{m_{l+1}} = w_{m_{(l+1)},i} x_{m_i} + b_{ml+1} \tag{3}$$

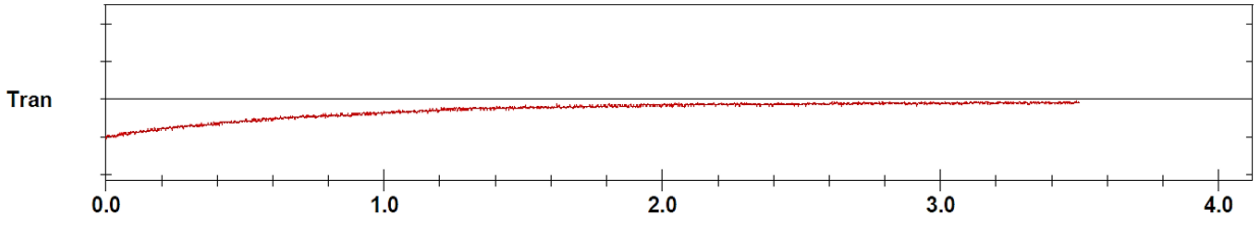
where v_{ml+1} is the m_{l+1}^{th} weighted sum, x_{m_i} is the m_i^{th} input value, and $b_{m_{l+1}}$ is the m_{l+1}^{th} bias value. Hence, the weighted sum matrix can be expressed as

$$\mathbf{V}^p = \mathbf{W}^p \mathbf{X}^l + \mathbf{B}^p \tag{4}$$

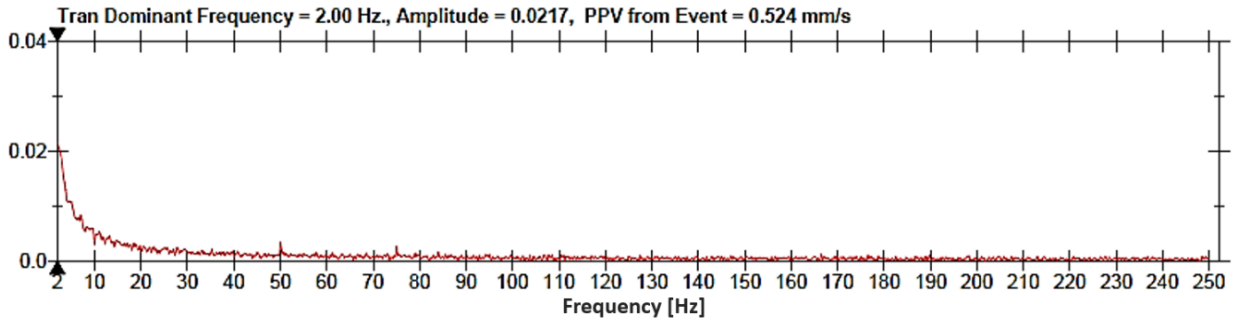
where $\mathbf{V}^p$ is the p^{th} weighted sum matrix and $\mathbf{B}^p$ is the p^{th} bias matrix. Therefore, the size of the weighted sum matrix equals to $M_{l+1} \times 1$.

Then, the weighted sum will be transformed to the designed output value via activation function $\varphi()$. Hence, the output matrix can be expressed as

$$\mathbf{Y}^{l+1} = \varphi(\mathbf{V}^p) \tag{5}$$

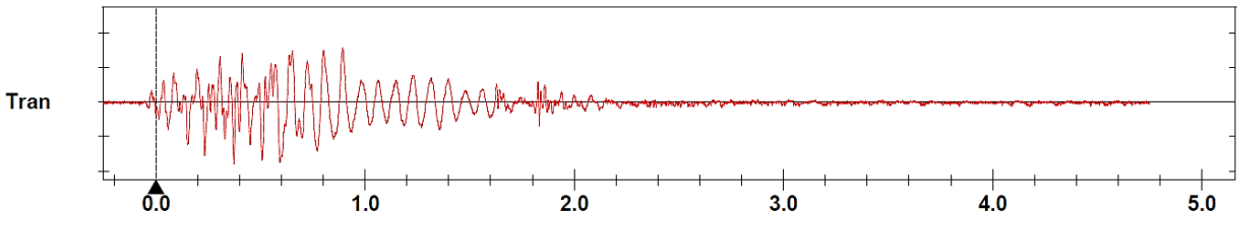


(a) Time domain

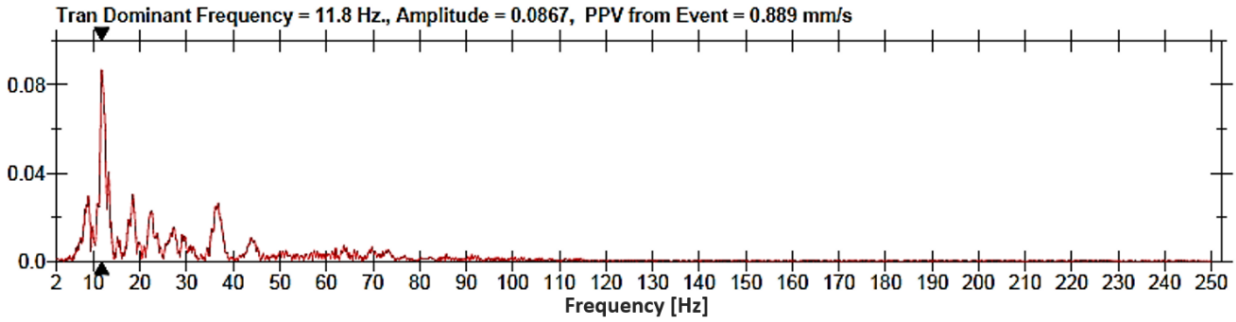


(b) Frequency domain

***Fig. 2**: Blast vibration report in the transverse dimension.*



(a) Time domain



(b) Frequency domain

***Fig. 3**: Non-blast vibration report in the transverse dimension.*

where Y^{l+1} is the $(l+1)^{th}$ output matrix and $\mathbf{V}p$ is the p^{th} weighted sum matrix. The examples of the activation function are Sigmoid function, Softmax function, and Ramp function (Rectified Linear Unit).

The Sigmoid function is one of the most popular activation functions, and its output has the value between zero and one. The output node from the Sigmoid function $\sigma()$ can be expressed as

$$y_{m_{l+1}} = \sigma\left(v_{m_{l+1}}\right) = \frac{1}{1+e^{-v_{m_{l+1}}}} \tag{6}$$

where $y_{m_{l+1}}$ is the m_{l+1}^{th} output node, $v_{m_{l+1}}$ is the m_{l+1}^{th} weighted sum.

In classification application, the last output layer of

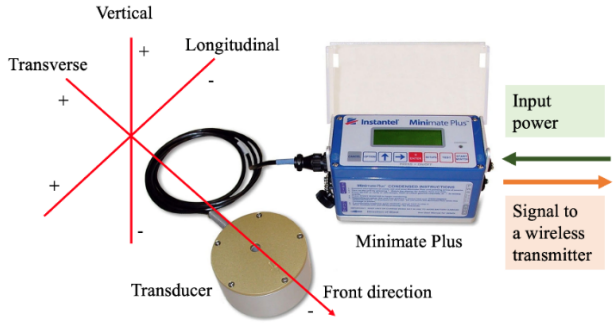


***Fig. 4**: Minimate Plus.*

the DNN is used to specify the predicted class. Then, the number of nodes in the last layer will be equal to the total number of classes, and their values should be the probability of being in a specific class. Therefore, the sum of the output nodes in the last layer will be equal to one which can be expressed as

$$\sum_{i=1}^{M_L} y_i^L = 1 \tag{7}$$

where y^L is the L^{th} output node which is in the last layer and M_L is the number of output nodes in the last layer. Unfortunately, the Sigmoid function can produce the output nodes with values from zeros to one, but cannot yield the sum of output nodes in unity. Thus, the suitable activation function for the last output layer is the Softmax function $\mathcal{S}()$ which can be expressed as

$$y_{m_L}^L = \mathcal{S}\left(v_{m_L}^P\right) = \frac{e^{v_{m_L}^P}}{\sum_{i=1}^{M_L} e^{v_i^P}} \tag{8}$$

where $y_{m_L}^L$ is the m_L^{th} output node at the last layer L and $v_{m_L}^P$ is the m_L^{th} weighted sum at the last weighted sum matrix P.

Following the property of the Softmax function, the sum of the Softmax function yields the sum of the output nodes which is equal to one and can be expressed as

$$\sum_{i=1}^{M_L} \mathcal{S}\left(v_i^P\right) = \sum_{i=1}^{M_L} y_i^L = 1. \tag{9}$$

In the past recent year, Softmax function was still be used in many researches [8–11]. Therefore, the complete algorithm for calculating output nodes of the DNN can be shown in Table 1.

2.3 Training Process

Before utilizing the designed DNN, the weight has to be trained until the DNN can predict input data in the correct class. Hence, to train the model, the trained data sets need the correct data to identify if the output of the DNN is the correct prediction or not. The correct data can be expressed as

***Table 1**: Pseudocode of the output calculation.*

Algorithm 1: Output calculation
Initialization
Prepare the input data: $\mathbf{X}^1 = \left[x_1^1, x_2^1, x_3^1, \ldots, x_{M_1}^1\right]$ where M_1 is the number of the input node at the first layer.
Set the number of layers: L
Set weight matrices: $\mathbf{W}^p$ with size $M_{l+1} \times M_l$ where $\mathbf{W}^p$ is the p^{th} weight matrix and l is the order of the layer.
Output calculation
for $p = 1 : P-1$
$l = p + 1$
$\mathbf{V}^p = \mathbf{W}^p \mathbf{X}^{l-1}$
$\mathbf{Y}^l = \sigma\left(\mathbf{V}^p\right)$
$\mathbf{X}^l = \mathbf{Y}^l$
end
$\mathbf{V}^P = \mathbf{W}^P \mathbf{X}^{L-1}$
$\mathbf{Y}^L = \mathcal{S}(\mathbf{V}^P)$

***Table 2**: Pseudocode of training algorithm.*

Algorithm 2: Training algorithm for one epoch
Initialization
Steps 1-3 are the same as Algorithm 1
Set the correct data: $\mathbf{C}$
Set the number of weight matrices: P
Set dropout ratio: $\frac{F}{M}$
Set learning rate: α
Output calculation
for $p = 1 : P-1$
$l = p + 1$
$\mathbf{V}^p = \mathbf{W}^p \mathbf{X}^{l-1}$
$\mathbf{Y}^l = \sigma\left(\mathbf{V}^p\right)$
$\mathbf{Y}^l = \mathcal{D}\left(\mathbf{Y}^l, \frac{F}{M}\right)$
$\mathbf{X}^l = \mathbf{Y}^l$
end
$\mathbf{V}^P = \mathbf{W}^P \mathbf{X}^{L-1}$
$\mathbf{Y}^L = \mathcal{S}(\mathbf{V}^P)$
Back propagation
Delta calculation
$\mathbf{E}^L = \mathbf{C} - \mathbf{Y}^L$
$\boldsymbol{\delta}^L = \mathbf{E}^L$
for $l = L : -1 : 2$
$p = l - 1$
$\mathbf{E}^{l-1} = \mathbf{W}^{p,T} \boldsymbol{\delta}^l$
$\boldsymbol{\delta}^{l-1} = \varphi'(\mathbf{V}^{p-1}) \mathbf{E}^{l-1}$
end
Weight adjustment
for $p = P : -1 : 1$
$l = p + 1$
$\Delta \mathbf{W}^p = \alpha \boldsymbol{\delta}^l \mathbf{X}^{l-1,T}$
$\mathbf{W}_{new}^p = \mathbf{W}_{old}^p + \Delta \mathbf{W}^p$
end

$$\mathbf{C} \in \text{a row in } \mathbf{I}_K \tag{10}$$

where $\mathbf{C}$ is the correct data, K is the number of classes, and $\mathbf{I}_K$ is the $K \times K$ identity matrix. For the classification of blast vibrations, data sets can be classified into 2 classes ($K = 2$). The class-1 data are blast vibrations, and the class-2 data are non-blast vibrations. Hence, the correct

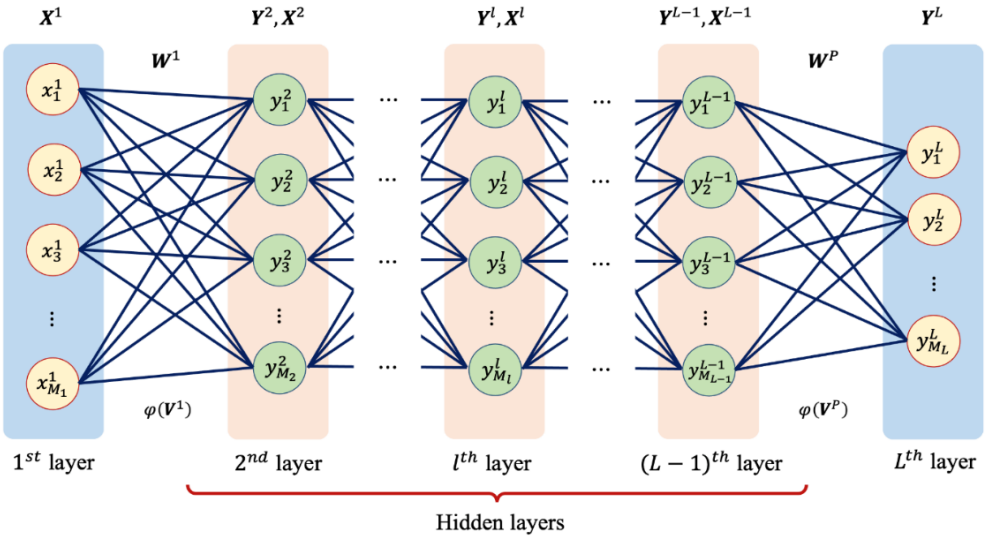


***Fig. 5**: Structure of the DNN.*

***Table 3**: Parameters of the training process.*

Parameter	Value
Number of Class 1 data sets Number of Class 2 data sets	20 20
Number of input nodes Number of output nodes	501 2
Number of hidden layers	3
Number of weight matrices	4
Size of weight matrix $\mathbf{W}^1$ Size of weight matrix $\mathbf{W}^2$, $\mathbf{W}^3$ Size of weight matrix $\mathbf{W}^4$	15 x 501 15 x 15 2 x 15
Activation function: Layer 2-4 Layer 5	 Sigmoid function Softmax function
Correct data	[1 0] and [0 1]
Dropout ratio	0.2
α	0.1, 0.01 & 0.001

data can be set as $\mathbf{C} \in \{[1\ 0], [0\ 1]\}$. Then, [1 0] is for class 1 and [0 1] is for class 2.

When the output of the DNN is computed, the prediction error of these trained data can be expressed as

$$\mathbf{E}^L = \mathbf{C} - \mathbf{Y}^L \tag{11}$$

where $\mathbf{E}^L$ is the error matrix at the last layer, $\mathbf{C}$ is the correct data matrix corresponding to input data, and $\mathbf{Y}^L$ is the output matrix at the last layer.

To reduces the error, the weights of the DNN have to be adjusted by the back-propagation algorithm. Fig. 6 shows the back-propagation diagram which consists of 2 processes: calculating the delta and updating the weight matrix via delta rule. As of the algorithm's name, the calculation begins at the last layer and ends at second layer. First, the error matrix for the last layer of the DNN is calculated by equation (11). Then, use the Delta rule to calculate delta matrix by

$$\boldsymbol{\delta}^l = \begin{cases} \mathbf{E}^L & , \text{ if } l = L \\ \varphi'\left(\mathbf{V}^p\right)\mathbf{E}^l & , \text{if } l < L \end{cases} \tag{12}$$

where $\boldsymbol{\delta}^l$ is the delta matrix at the l^{th} layer, $\varphi'\left(\mathbf{V}^p\right)$ is the derivative of the activation function of the p^{th} weighted sum matrix, and $p = l-1$. Then, the delta matrix is used to find the additional weight by

$$\Delta\mathbf{W}^p = \alpha\boldsymbol{\delta}^l\mathbf{X}^{l-1,T} \tag{13}$$

where $\Delta\mathbf{W}^p$ is an additional weight matrix for the p^{th} weight matrix, α is the learning rate, and $\mathbf{X}^{l-1,T}$ is the transpose of the input matrix at the $(l-1)^{th}$ layer. Then the updated weight matrix can be expressed as

$$\mathbf{W}^p_{new} = \mathbf{W}^p_{old} + \Delta\mathbf{W}^p \tag{14}$$

To update the previous weight matrix W^{p-1}, the error matrix of the $(l-1)^{th}$ layer is calculated by

$$\mathbf{E}^{l-1} = \mathbf{W}^{p,T}_{old}\boldsymbol{\delta}^l \tag{15}$$

where $\mathbf{E}^{l-1}$ is the error matrix of the $(l-1)^{th}$ layer, $\mathbf{W}^{p,T}_{old}$ is the transpose of the p^{th} old weight matrix, and $\boldsymbol{\delta}^l$ is the delta of the l^{th} layer.

Once all weight matrices in the DNN are updated, the next trained data set $\mathbf{X}$ and correct data $\mathbf{C}$ are used to train the model. This training process will repeat until it iterates through all the data sets. The entire iteration is called "one epoch". Only one epoch of training will not be sufficient to make an accurate prediction system. To improve the system, the weight matrices must be trained in many epochs. However, if the training takes too many epochs which is called "overtraining", the DNN will become too strict with the input data. Such a situation is called "overfitting" which will disregard a

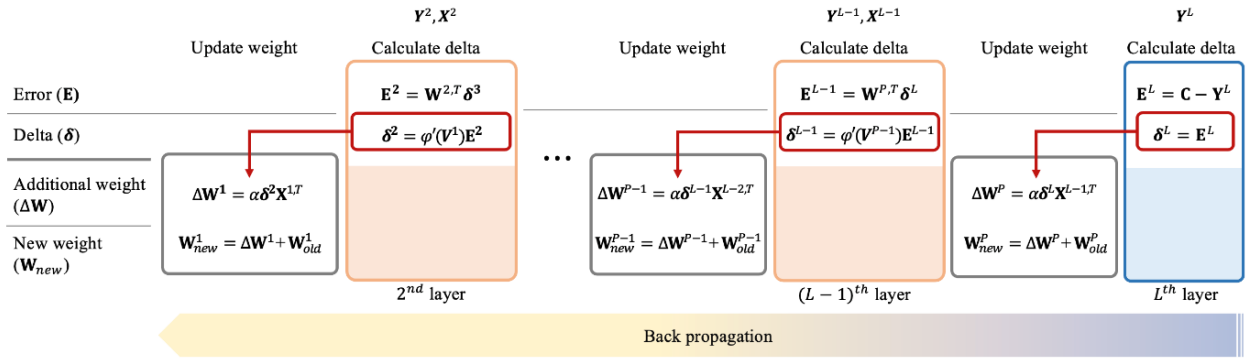


Fig. 6: Back-propagation diagram.

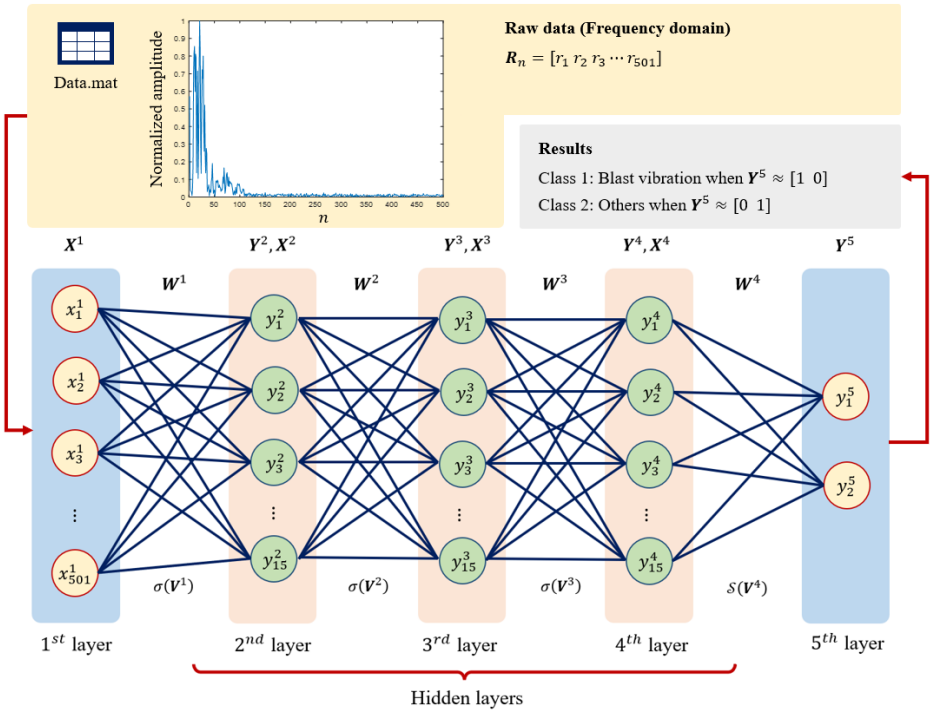


Fig. 7: Design deep neural network.

sense of flexibility for the prediction. Hence, the number of prediction errors will increase.

To overcome the overfitting problem without knowing the suitable number of epochs, a dropout technique is presented. Its function is to drop out some input and weight nodes in the DNN. Consequently, some nodes of the outputs from hidden layers will be randomly selected and set to 0. Thus, the dropout function can be expressed as

$$\left(\mathbf{Y}, \frac{F}{M}\right) = \begin{bmatrix} y_1 \\ \vdots \\ y_m \\ \vdots \\ y_M \end{bmatrix} \tag{16}$$

where y_m equals zero when $m \in \mathbf{O}$ and equals $y_m\left(\frac{1}{1-\frac{F}{M}}\right)$ when $m \notin \mathbf{O}$. $\frac{F}{M}$ is a dropout ratio which is the ratio of the number of the dropout value F to the total number of the output data M, $\mathbf{O}$ is the index matrix equal to $[o_1, o_2, \ldots, o_f, \ldots, o_F]$ which causes y_m equal to 0, and o_f is the random positive integer less than or equal to M.

The dropout ratio can be set from 0.2 - 0.5 depending on the application [12-15]. Hence, the complete training algorithm can be expressed in Table 2.

3. DESIGNED SYSTEM

The DNN used for this system has 3 hidden layers as shown in Fig. 7. The input data is the spectrum of ground vibrations recorded via Minimate Plus with the sample rate equal to 1024 bits/s. According to Nyquist–Shannon sampling theorem, the Nyquist frequency equals 1204

***Table 4**: Pseudocode of the weight training algorithm for the designed model.*

Algorithm 3: Weight training algorithm

Initialization

Load data matrix R from the file "data.mat"
Prepare test data sets with 80% of all data in $\mathbf{R}$:
$\mathbf{X} = [\,]$ and $\mathbf{D} = [\,]$
for $i = 1{:}\,2$ classes
  for $j = 1{:}\,20$ data
    $\mathbf{X} = [\mathbf{X} : \mathbf{R}((i-1)100 + j)]$
    $\mathbf{D} = [\mathbf{D} : \mathbf{C}((i-1)100 + j)]$
  end
end
Set weight matrices with the initial values between -1 to 1
$\mathbf{W}^1 = 2\text{rand}(15{,}501) - 1$
$\mathbf{W}^2 = 2\text{rand}(15{,}15) - 1$
$\mathbf{W}^3 = 2\text{rand}(15{,}15) - 1$
$\mathbf{W}^4 = 2\text{rand}(2{,}15) - 1$
Set dropout ratio: $\frac{F}{M} = 0.2$
Set learning rate: α

Weight training

for $i = 1$:epochs
  for $k = 1{:}\,40$ data sets
  Output calculation
    $\mathbf{X}^1 = \mathbf{R}^T(k,:)$ // Store k^{th} train
    // data set to $\mathbf{X}^1$
    $\mathbf{C} = \mathbf{D}^T(k,:)$ // Store k^{th}correct
    // data set to $\mathbf{C}$
    Calculate data from the 1st to 4th layer
    for $p = 1{:}\,3$
      $l = p + 1$
      $\mathbf{V}^p = \mathbf{W}^p\mathbf{X}^{l-1}$
      $\mathbf{Y}^l = \sigma(\mathbf{V}^p)$
      $\mathbf{Y}^l = \mathcal{D}\left(\mathbf{Y}^l, \frac{F}{M}\right)$
      $\mathbf{X}^l = \mathbf{Y}^l$
    end
    Calculate data for the last layer
    $\mathbf{V}^4 = \mathbf{W}^4\mathbf{X}^4$
    $\mathbf{Y}^5 = \mathcal{S}(\mathbf{V}^4)$
  Back propagation
    Delta calculation
    $\mathbf{E}^5 = \mathbf{C}^T(k,:) - \mathbf{Y}^5$
    $\boldsymbol{\delta}^5 = \mathbf{E}^5$
  for $l = 5{:}\,{-1}{:}\,2$
    $p = l - 1$
    $\mathbf{E}^{l-1} = \mathbf{W}^{p,T}\boldsymbol{\delta}^l$
    $\boldsymbol{\delta}^{l-1} = \varphi'(\mathbf{V}^{p-1})\mathbf{E}^{l-1}$
    end
    Weight adjustment
    for $p = 4{:}\,{-1}{:}\,1$
      $l = p + 1$
      $\Delta\mathbf{W}^p = \alpha\boldsymbol{\delta}^l\mathbf{X}^{l-1,T}$
      $\mathbf{W}^p_{new} = \mathbf{W}^p_{old} + \Delta\mathbf{W}^p$
    end
  end
end

***Table 5**: Pseudocode of the correlation classification.*

Algorithm 4: Correlation classification

Initialization

Steps 1-2 are the same as Algorithm 3
$\mathbf{X}' = \mathbf{X}(1,:)$ // Set the reference blast vibration data

Output calculation

for $k = 1{:}\,160$ data sets
  $\mathbf{X}'' = \mathbf{X}(k,:)$
  $\mathcal{C}_c = \frac{N\mathbf{X}'\mathbf{X}''^T - \sum\mathbf{X}'\sum\mathbf{X}''}{\sqrt{N\sum\mathbf{X}'^2 - (\sum\mathbf{X}')^2}\sqrt{N\sum\mathbf{X}''^2 - (\sum\mathbf{X}'')^2}}$
  if $\mathcal{C}_c > 0$
    $y_c = \mathcal{C}_c$
  else if $\mathcal{C}_c \le 0$
    $y_c = 0$
  end
end

Hz/2 = 512 Hz. Then, the Fast Fourier Transform (FFT) is used to convert the vibration data from time domain to frequency domain with 501 values. Consequently, the frequency interval between spectrum values is 512 Hz/500 intervals = 1.024 Hz which is an appropriate resolution for this application. To solve the uncertainty of amplitude of the vibration data, the spectrum needs to be normalized with its maximum value.

Then, the spectrum is fed into the Input layer which has the 501 nodes equal to the spectrum values. The output of the DNN has 2 nodes to classify only 2 classes: one is vibration from blast operation, and the others are vibrations from arbitrary sources. Hence, result of these trained data can either be [1 0] or [0 1], and the output layer has only 2 nodes. There are 3 hidden layers in the model. Each hidden layer has 15 nodes corresponding to 4 weight matrices with the sizes of 15x501, 15x15, 15x15, and 2x15, respectively. The activation function begins with the Sigmoid function, and the last layer ends with the Softmax function. All necessary parameters are shown in Table 3.

There are 200 data sets used in this study. Each data set is contained in the .txt file. These files are grouped into 2 categories. The first category is the blast vibration data consisting of 100 .txt files which are labelled with their file names from "Y1" to "Y100". The other is the non-blast vibration data also consisting of 100 .txt files which are labelled with their file names from "N1" to "N100". Then, MALAB is used to convert these data in .txt file format to the normalized-amplitude spectrum in the .mat file format via the FFT function and unity normalization. The .mat file contains the data matrix **R** and correct data matrix **C** which are arranged by the first 100 blast data sets with the significant order of data.

Next, 20% of all data sets (40 data sets) are used to train the model, and the rest (160 data sets) are used to validate the model. The training algorithm for the designed model is shown in Table 4.

Because the number in the initial weight matrix is arbitrary, the results of adjusting weight is not constant. Hence, the training and testing model are repeated 10 times with each of learning rates of 0.1, 0.01, and 0.001. After each training process, the model will be tested with 80% of the data sets.

After that, the performance of the DNN model will be compared with the other two classification techniques

such as correlation coefficient and coefficient of determination.

3.1 Correlation Coefficient

The correlation coefficient is a numerical indicator of the relation between two variables. The correlation coefficient technique can be used to reduce the data dimensions for detecting linear and non-linear correlation [16]. It also can be applied in power analysis [17] and energy-efficient MIMO (Multiple-Input Multiple-Output) applications [18].

This article uses the Pearson correlation coefficient to classify the vibration by setting the reference blast vibration data set as the matrix $\mathbf{X}' = [x'_1, x'_2, \ldots, x'_N]$ and the test vibration data set as the matrix $\mathbf{X}'' = [x''_1, x''_2, \ldots, x''_N]$. Then, the Pearson correlation coefficient C_c can be computed by

$$C_c = \frac{\sum_{n=1}^{N}(x'_n - \overline{x}')(x''_n - \overline{x}'')}{\sqrt{\sum_{n=1}^{N}\left(x'_n - \overline{x}'\right)^2}\sqrt{\sum_{n=1}^{N}\left(x''_n - \overline{x}''\right)^2}} \quad (17)$$

$$= \frac{N\mathbf{X}'\mathbf{X}''^T - \sum\mathbf{X}'\sum\mathbf{X}''}{\sqrt{N\sum\mathbf{X}'^2 - \left(\sum\mathbf{X}'\right)^2}\sqrt{N\sum\mathbf{X}''^2 - \left(\sum\mathbf{X}''\right)^2}}$$

where $\overline{x}'$ is the mean of the reference blast vibration data set and $\overline{x}''$ is the mean of the test vibration data set.

The Pearson correlation coefficient has the value between - 1 to 1 where 1 represents the two data sets are perfectly correlated. In contrast, 0 means the two data sets are uncorrelated. Therefore, the value of the correlation of the test data set being the blast vibration. Then, the output of the correlation coefficient classification y_c can be set as the probability like the DNN and expressed as

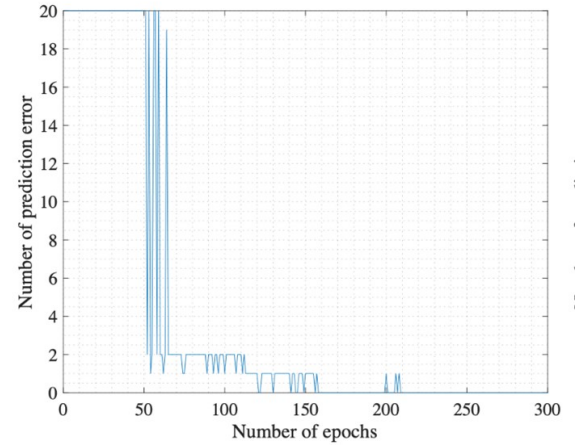
$$y_c = \begin{cases} y_c & , C_c > 0 \\ 0 & , C_c \le 0 \end{cases} . \quad (18)$$

Table 5 shows the pseudocode of correlation classification for this research. First is the initial process. The 200 vibration data sets are loaded from file "data.mat" to the matrix $\mathbf{R}$. Then, the 21^{st} - 100^{th} (blast vibration) and 121^{st} - 200^{th} (non-blast vibration) data sets are stored in the matrix $\mathbf{X}$. After that the 1st data set of the matrix $\mathbf{X}$ is set to the reference blast vibration data set and stored in the matrix $\mathbf{X}'$.

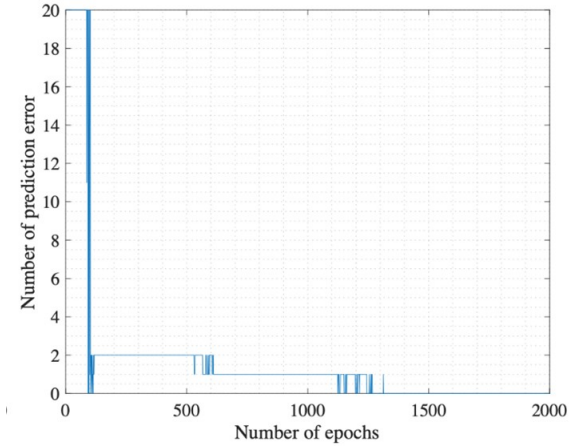
Second is the computational process. Each data set from matrix $\mathbf{X}$ is stored in the test data set matrix $\mathbf{X}''$, Next, Pearson correlation coefficient between matrix $\mathbf{X}'$ and $\mathbf{X}''$ is computed. Then, the probability of being the blast vibration y_c is only the positive value of the coefficient, otherwise is set to 0.

3.2 Coefficient of Determination

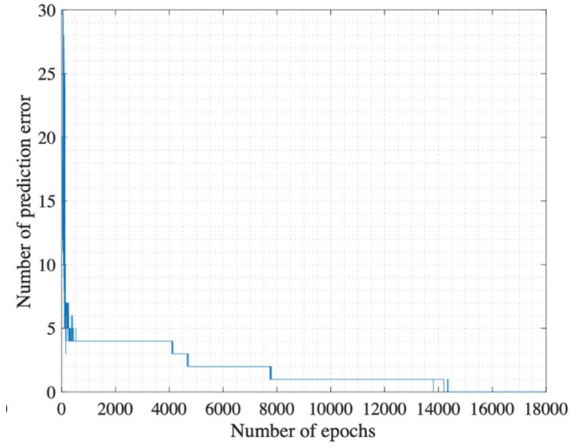
Coefficient of determination or R-squared is a numerical indicator of the similarity between two variables via proportion of variance which can be computed by



(a) α = 0.1



(b) α = 0.01



(c) α = 0.001

***Fig. 8**: Graph of the number of prediction error versus the number of epochs in different learning rates.*

$$\mathcal{R}^2 = 1 - \frac{\sum_{n=1}^{N}\left(x'_n - x''_n\right)^2}{\sum_{n=1}^{N}\left(x'_n - \overline{x}'\right)^2} \quad (19)$$

$$= 1 - \frac{\sum\left(\mathbf{X}' - \mathbf{X}''\right)^2}{\sum\left(\mathbf{X}' - \overline{\mathbf{X}}'\right)^2}. \quad (20)$$

***Table 6**: Pseudocode of the R-squared classification.*

Algorithm 5: R-squared classification
Initialization
Steps 1-3 are the same as Algorithm 4
4: $\bar{x}' = \text{mean}(\mathbf{X}')$
Output calculation
1: for $k = 1{:}160$ data sets
2: $\mathbf{X}'' = \mathbf{X}(k,:)$
3: $y_{R^2} = 1 - \frac{\sum(\mathbf{X}'-\mathbf{X}'')^2}{\sum(\mathbf{X}'-\bar{\mathbf{X}}')^2}$
4: end

The product of R-square has a value between 0-1 which can be implied as the probability of one variable being the other. R-squared can be used for analyzing the performance of the machine learning models [19], and also applied in the detection application for extracting data [20]. The concept of applying R-squared for vibration classification is similar to the correlation coefficient classification in the previous subsection.

To classify ground vibration into blast vibration, the probability of being blast vibration with R-squared is set as

y_{R^2} which is equal to the R-squared value. Then, Table 6 shows the pseudocode of the R-squared classification. The first initial process is similar to the correlation coefficient classification. The R-squared has one more step which is calculation of mean of the reference blast vibration data. In the computational process, the Pearson correlation coefficient formula is replaced by the R-squared formula, and "if" condition is eliminated.

4. RESULTS AND DISCUSSION

During the training process, the number of the prediction error decreases depending on the learning rate as shown in Fig. 8.

The graph clearly reveals that the size of the learning rate is directly proportional to the number of epochs. For the learning rate equal to 0.1, the number of the prediction error becomes zero after 220 epochs. For the learning rate equal to 0.01, the number of the prediction error is zero after 14400 epochs. Due to lager number of the training epochs, the training process takes longer time than that with smaller learning rates.

Table 7 shows the percentage of the class prediction in training and testing process. For training process, the models with learning rate 0.1 and 0.01 can correctly classify the vibration data. For learning rate 0.001, the model has 5% mistake from predicting class 2 as class 1.

For testing process, the prediction error occasionally occurs in all learning rates, but a larger learning rate can produce a larger error. The learning rate equal to 0.001 yields 7.5% maximum prediction error in class-2 data. Meanwhile, the learning rate equal to 0.1 yields only 5% maximum prediction error in class-2 data. However, all learning rates can produce weight matrices which have

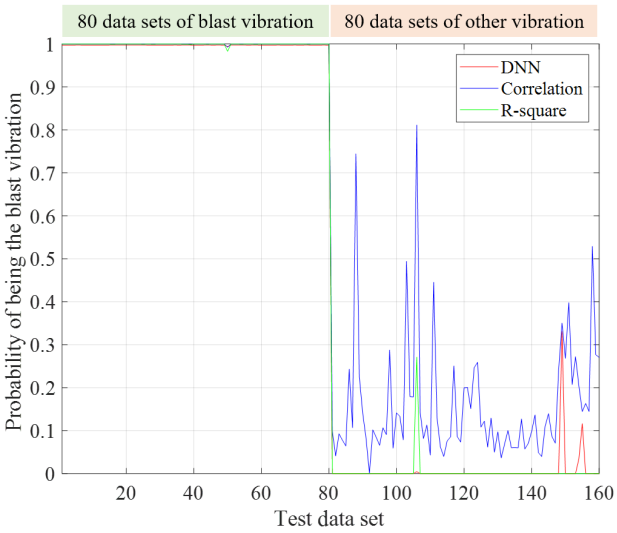


***Fig. 9**: Probability of being the blast vibration.*

0% prediction error. Thus, to reduce training time and yield smaller probable errors, the learning rate equal to 0.01 is appropriate.

Due to the Softmax function, every output of the DNN has the value between 0 to 1 which can be considered as probability measure. For the correlation coefficient, the minus value from -1 to 0 means unlikelihood between two signals. Besides, the positive value from 0 to 1 presents the likelihood which can also be considered as the probability.

To compare the prediction performance of two approaches, Fig. 9 shows the probability of being the blast vibration of the DNN and correlation technique. The DNN (red line) has better performance than correlation (blue line). If the threshold probability is set to 0.5, the correlation will make the wrong prediction for 3 data sets with the probabilities of 0.8113, 0.7442, and 0.5287. As the results, it is clear that the DNN is more a suitable technique than the correlation.

For calculation time, functions "tic" and "toc" are used. Table 8 shows the calculation time for testing 160 data sets in 5 times. R-squared uses the longest time around 0.17 s while DNN uses the shortest time around 0.0046 s. Correlation spends time slightly longer than DNN about 0.0015 s.

As the results, it is clear that the DNN is more a suitable technique than the correlation and R-squared in both accuracy and calculation time.

5. CONCLUSIONS

In this article, the DNN with 3 hidden layers is applied to classify the blast vibration for the coal mining in order to reduce the human process which causes time inefficiency and human error. The proposed DNN model classifies the vibration data into 2 classes which are the blast vibration (Class 1) and non-blast vibration (Class 2). The suitable learning rate for training process is

***Table 7**: Percentage of the class prediction in different learning rate and epochs.*

No.	Percentage of the class prediction in the training and testing process											
	α = 0.1, 300 epochs				α = 0.01, 5000 epochs				α = 0.001, 18000 epochs			
	1→1	1→2	2→1	2→2	1→1	1→2	2→1	2→2	1→1	1→2	2→1	2→2
Training process												
1-2	100	-	-	100	100	-	-	100	100	-	-	100
3	100	-	-	100	100	-	-	100	100	-	5	95
4-10	100	-	-	100	100	-	-	100	100	-	-	100
Testing process												
1	100	-	-	100	100	-	1.25	98.75	100	-	-	100
2	100	-	-	100	100	-	-	100	100	-	-	100
3	100	-	2.5	97.5	100	-	-	100	100	-	3.75	96.25
4	100	-	-	100	100	-	5	95	100	-	5	95
5	100	-	1.25	98.75	100	-	2.5	97.25	100	-	7.5	92.5
6	100	-	5	95	100	-	-	100	100	-	-	100
7	100	-	-	100	100	-	1.25	98.75	100	-	2.5	97.25
8	100	-	2.5	97.5	100	-	-	100	100	-	1.25	98.75
9	100	-	2.5	97.5	100	-	3.75	96.25	100	-	6.25	93.75
10	100	-	-	100	100	-	1.25	98.75	100	-	3.75	96.25

Remark
1→1 means the input class 1 and the predicted as class 1
1→2 means the input class 1 and the predicted as class 2
2→1 means the input class 2 and the predicted as class 1
2→2 means the input class 2 and the predicted as class 2

***Table 8**: Calculation time for testing 160 data sets.*

Algorithm	Calculation time [s]					
	1	2	3	4	5	Ave.
DNN	0.0038	0.0051	0.0044	0.0053	0.0044	0.0046
Correlation	0.0056	0.0065	0.0056	0.0065	0.0064	0.0061
R-squared	0.2519	0.1478	0.1259	0.1259	0.2116	0.1713

0.001, and the trained DNN model can classify the blast vibration with the 100% accuracy under the condition that the non-blast vibration has discrete and multiple frequencies. In addition, DNN outperforms Pearson correlation coefficient and coefficient of determination (R-squared).

Furthermore, this work could be extended in the future to modify the DNN structure by reducing the weight size or the number of hidden layers which can reduce the calculation time. This work can also be applied to automatic blast vibration reports with a high level of security which makes data reporting via blockchain technology more reliable to inspectors from an environmental organization.

REFERENCES

[1] C. Shao, Q. Wu, and G. Xin, "The Research on Safety Monitoring System of Coal Mine based on Spatial Data Mining," in *Proceeding of the 2nd International Workshop on Knowledge Discovery and Data Mining*, Moscow, Russia, 2009, pp. 126-129.

[2] R. Lv, B. Li, and Y. Xu, "Fuzzy Comprehensive Evaluation of Groundwater Pollution in Coal Mining Area," in *Proceeding of the International Symposium on Water Resource and Environmental Protection*, vol. 1, Xi'an, China, 2011, pp. 20-23.

[3] V. Henriques, R. Malekian, and D. Capeska Bogatinoska, "Mine safety system using wireless sensor networks," in *Proceeding of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2017, pp. 515-520.

[4] D. C. Mazur, J. A. Kay, and K. D. Mazur, "Advancements in Vibration Monitoring for the Mining Industry," *IEEE Transactions on Industry Applications*, vol. 51, no. 5, pp. 4321-4328, Sept.-Oct. 2015.

[5] W. Srisawasdi, T. W. Tsusaka, E. Winijkul, and N. Sasaki, "Valuation of Local Demand for Improved Air Quality: The Case of the Mae Moh Coal Mine Site in Thailand," *Atmosphere*, vol. 12, no. 9, pp. 1-27, Sept. 2021.

[6] Instantel, *Minimate Plus Operator Manual*, 2013.

[7] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, Springer, New York, 2017.

[8] H. S. Razzaq and Z. M. Hussain, "Instantaneous Frequency Estimation of FM Signals under Gaussian and Symmetric-Stable Noise: Deep Learning versus Time–Frequency Analysis," *Information*, vol. 14, no. 1, pp. 1-43, Dec. 2022.

[9] Y. Cui and F. Wang, "Research on audio recognition based on the deep neural network in music teaching," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-8, May 2022.

[10] K. Kozu, Y. Tanabe, M. Kitakami, and K. Namba, "Low Power Neural Network by Reducing Sram Operating Voltage," *IEEE Access*, vol. 10, pp. 116982-116986, Nov. 2022.

[11] L. Zhang, C. Bao, and K. Ma, "Self-Distillation: Towards Efficient and Compact Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4388-4403, Aug. 2021.

[12] Y.Wen, S. J. Kim, S. Avrillon, J. T. Levine, F. Hug, and J. L. Pons, "A Deep CNN Framework for Neural Drive Estimation from HD-EMG Across Contraction Intensities and Joint Angles," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2950-2959, Oct. 2022.

[13] C. Sticht, *Power system waveform classification using Time-Frequency and CNN*, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), Tech. Rep., 2022.

[14] Q. Ma, M. Wang, L. Hu, L. Zhang, and Z. Hua, "A Novel Recurrent Neural Network to Classify EEG Signals for Customers' Decision-Making Behavior Prediction in Brand Extension Scenario," *Frontiers in Human Neuroscience*, vol. 15, pp. 1-13, Mar. 2021.

[15] C. Tanner Fredieu, A. Martone, and R. M. Buehrer, "Open-Set Classification of Common Waveforms using a Deep Feed-Forward Network and Binary Isolation Forest Models," *arXiv*, pp.1-8, Oct. 2021.

[16] S. Wang and L. Zhang, "A Supervised Correlation Coefficient Method: Detection of Different Correlation," in *Proceeding of the 12th International Conference on Advanced Computational Intelligence (ICACI)*, Dali, China, 2020, pp. 408-411.

[17] J. Kundrata, D. Fujimoto, Y. Hayashi, and A. Bari´c, "Comparison of Pearson Correlation Coefficient and Distance Correlation in Correlation Power Analysis on Digital Multiplier," in *Proceeding of the 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2020, pp. 146-151.

[18] S. Wane, B. Kieniewicz, J. B. Sombrin, D. Bajon, P. Poilvert, C.-A. Tavernier, and D. Floriot, "Scalable Modular Beamformers for Energy-Efficient MIMO Applications Using Correlation Technologies," in *Proceeding of the IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, Waco, TX, USA, 2023, pp. 1-5.

[19] P. Kumar, P. Priyanka, J. Dhanya, K. V. Uday, and V. Dutt, "Analyzing the Performance of Univariate and Multivariate Machine Learning Models in Soil Movement Prediction: A comparative study," *IEEE Access*, vol. 11, pp. 62368-62381, Jun. 2023.

[20] M. Awawdeh, T. Faisal, A. Bashir, and A. Sheikh, "Application of Outlier Detection using Re-Weighted Least Squares and R-Squared for IOT Extracted Data," in *Proceeding of the Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, United Arab Emirates, 2019, pp. 1-6.

Jirasak Thothong received his B.S.Tech.Ed. in Electrical Engineering and M.S.Tech.Ed. in Electrical Technology from King Mongkut's Institute of Technology North Bangkok, Thailand, in 1999 and 2004. From 2002 to 2022, he was a Senior Project Engineer in Siemens Limited, Bangkok, Thailand. He is currently studying Ph.D. in Electrical Engineering at Srinakharinwirot University, Thailand. His research interests in wireless communications and digital telecommunication.

Kanok Charoenchaprakit received his B.Eng. degree in Electrical Engineering from Chulachomklao Royal Military Academy, Thailand, in 2007, his M.S. degree in Electrical Engineering from the University of New Haven, CT, USA, in 2011, and his Ph.D. in Electrical Engineering from Srinakharinwirot University, Thailand, in 2021. From 2008 to 2009, he was a signal maintenance officer in the Royal Thai Army. Then, in 2012, he worked as a radio officer. Subsequently, he has been a lecturer in Electrical Engineering at Chulachomklao Royal Military Academy, a position he has held since 2014. His research interests in digital signal processing, communication system, and optimization.

Wekin Piyarat received his B.Eng. degree in Electrical Engineering from South-East Asia University, Bangkok, Thailand, in 1994. He received the M.Eng and D.Eng. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1998 and 2010. He is currently an Associate Professor at Srinakharinwirot University, Thailand. His research interests include power electronic, electric drives, renewable energy, and applications of Internet of Things.

Kampol Woradit received his B.Eng. (Hons.) and Ph.D. degrees in Electrical Engineering from Chulalongkorn University, Thailand, in 2002 and 2010, respectively. He was a visiting student in the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology (MIT), MA, USA, from 2007 to 2008, and was on a postgraduate research attachment in the Modulation and Coding Department at the Institute for Infocomm Research, Singapore. He was a faculty member in the Department of Electrical Engineering at Srinakharinwirot University, Thailand, from 2010 to 2020, and is currently a faculty member with the Department of Computer Engineering at Chiang Mai University, Thailand. His research interests include wireless communications, Internet of Things, and physical layer security. He received the ECTI-CON Best Paper Award in 2019, and the Royal Golden Jubilee Scholarship from the Thailand Research Fund for his Ph.D. program.