Ensemble Augmentation for Deep Neural Networks Using 1-D Time Series Vibration Data

Atik Faysal¹, Ngui Wai Keng², M. H. Lim³

¹²College of Engineering, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300, Gambang, Kuantan, Pahang, Malaysia.
³Institute of Noise and Vibration, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia.
Corresponding author: Ngui Wai Keng (e-mail: wkngui@ump.edu.my).

Abstract

Time-series data are one of the fundamental types of raw data representation used in data-driven techniques. In machine condition monitoring, time-series vibration data are overly used in data mining for deep neural networks. Typically, vibration data is converted into images for classification using Deep Neural Networks (DNNs), and scalograms are the most effective form of image representation. However, the DNN classifiers require huge labeled training samples to reach their optimum performance. So, many forms of data augmentation techniques are applied to the classifiers to compensate for the lack of training samples. However, the scalograms are graphical representations where the existing augmentation techniques suffer because they either change the graphical meaning or have too much noise in the samples that change the physical meaning. In this study, a data augmentation technique named ensemble augmentation is proposed to overcome this limitation. This augmentation method uses the power of white noise added in ensembles to the original samples to generate real-like samples. After averaging the signal with ensembles, a new signal is obtained that contains the characteristics of the original signal. The parameters for the ensemble augmentation are validated using a simulated signal. The proposed method is evaluated using 10 class bearing vibration data using three state-of-the-art Transfer Learning (TL) models, namely, Inception-V3, MobileNet-V2, and ResNet50. Augmented samples are generated in two increments: the first increment generates the same number of fake samples as the training samples, and in the second increment, the number of samples is increased gradually. The outputs from the proposed method are compared with no augmentation, augmentations using deep convolution generative adversarial network (DCGAN), and several geometric transformation-based augmentations. The results from the first increment showed that the classifiers with the ensemble augmentation have higher validation and test accuracy than no augmentation and have the highest mean validation-test accuracy than all the other augmentation techniques. In the second increment, for an increasing number of augmented data, the accuracy of the proposed method never decreased. On the other hand, the accuracy decreased for the other methods after a certain point as the augmentation methods might not be suitable for the data. Moreover, the classifiers with ensemble augmentation achieved the highest validation and test accuracy and the highest precision for each class in both increments. Robustness test conducted with noisy test samples and test samples from different loads showed that the classifiers can also obtain much higher robustness when trained with samples from ensemble augmentation. All the coding files used in the research can be found at https://github.com/atik666/AugmentationCode.

Keywords: data augmentation, transfer learning, condition monitoring, DCGAN, vibration signal.

1. Introduction

Time series is one of the common types of data representation in the field of data-driven intelligent methods. In machine condition monitoring, fault diagnosis using vibration analysis is the most common and effective method [1]. The vibration signals are another form of time series representation which are typically of 1-D in shape. The use of artificial intelligence for finding the fault pattern in vibration signals is a popular method [2]. Lately, data mining of time-series signals using deep learning has been much popularized owing to its feature learning ability [3] [4].
Generally, the DNN architecture is a handy tool for image processing and learning features from the input images. So, these models can obtain high accuracy when the input variables are images. Nevertheless, a fine-tuned DNN model would still provide different accuracy on different input image types. C. C. Chen et al. [5] used a 1-D convolution neural network (CNN) model to use the raw vibration signal from rolling bearing as a direct input. No pre-processing was conducted, but the model still obtained high accuracy. Mitiche et al. [6] applied 1-D CNN-based real-time fault diagnosis for the power assessment system. TL was utilized for training the 1-D CNN [7]. Nevertheless, 1-D vibration data reveals the information present only in the time domain, and it is much more feasible to extract information from a higher dimension. Hoang et al. [8] first converted vibration signals into a 2D form and produced grayscale images. The grayscale images were analyzed with CNN for bearing fault diagnosis. 2-D time-frequency image samples were used for bearing fault diagnosis using TL-CNN in [9]. Nevertheless, compared to grayscale images, RGB images contain more information as grayscale images have only one channel, whereas RGB images contain three channels.

The wavelet representation of images is the best fit because it can extract the most information from 1-D time-domain signals and represent them with both time and frequency information [10]. A scalogram approach is more suitable for the chore of fault classification owing to its detailed depiction of the signal. Wavelet scalogram representations have been proved to be effective and gaining more popularity over other representations [11]. Scalogram is defined as the time-frequency representation of a signal that depicts the obtained energy density using continuous wavelet transform (CWT) [12]. Verstraete et al. [13] applied CNN on two public datasets for bearing datasets for fault diagnosis. Three different input images were produced: short-time Fourier transform (STFT) spectrogram, wavelet scalogram, and Hilbert–Huang transform (HHT) image. Comparing the output accuracy from all the image types, the wavelet scalogram appeared to be the best fit. The wavelet scalogram images were also more robust in case of added noise. In [14], CWT scalograms were generated from the vibration signals. The features from the scalogram images were extracted by the corresponding layers of DNN. Later, all the layers were fused into a single layer to get all the features. The effectiveness of the proposed method was justified by the experimental bearing signals. All the previous research shows that the wavelet scalograms have a much higher advantage than the other image representations for 1-D time-series signals. Therefore, CWT scalograms are used for all kinds of input sample generation in this study.

Apart from all its perks, the DNN models possess a serious limitation. Typically, they require massive labeled data during the training phase [15]. However, in the practical field, faults occur abruptly and need attention within a very short time. To address this limitation, practically collecting a huge fault dataset is not probable. If enough data is not given during the training phase, the classifier suffers from the risk of overfitting and poor generalization. To introduce more data during the training phase, several data augmentation techniques are available. The two most common ones are (1) geometrical transformation (2) generative adversarial networks (GANs) based data augmentation. In geometrical transformation, the image is rotated, flipped, adjusted brightness, added noise, transitioned, cropped to augment new samples [16]. However, this approach would not work well with the wavelet scalograms because, unlike images of objects, scalograms are a graphical representation. A transformed representation of a graph would have a new meaning and may result in reduced accuracy. In the case of noise addition in geometrical transformation, some degree of noise (e.g., salt-pepper noise) is added with the images to introduce some abnormality in the training samples. This works well because the test data might be
noisy in the practical field and differ considerably from the training data. However, the added noise may conceal important information from the scalograms, producing poor accuracy when the augmented samples are increased. To solve this problem and increase model robustness, a Gaussian noise layer has been proposed in some literature [17]. However, that does not necessarily mean increasing the training samples and falls out of the scope of this study. So, the addition of noise to the training samples needs a more sophisticated method.

One of the most impressive and arising techniques in deep learning, GANs has been getting much attention ever since its arrival [18]. GANs is an unsupervised learning algorithm used for image generation where it takes some real images and produces similar outputs from complete white noise. The outputs contain a combination of training data of GANs input and may contain some degree of white noise and distortion. Thus, GANs can be used for data augmentation where the augmented data is never existed before and saves the classifier from repeated training samples. There are many types and improvements of GANs used by many researchers for data augmentation. Arjovsky [19] proposed Wasserstein GAN (WGAN), which is a big step towards the improvement of GANs to produce realistic images. However, WGAN works best for medical images, and its performance for vibration signals is unknown. Wang [20] proposed a fault diagnosis method for planetary gearbox by combining GANs and autoencoder. However, as it uses the vanilla GANs, the output is comparatively noisy, and autoencoders can lose necessary data, which needs additional research. Cycle-GAN [21] was proposed for unpaired image-to-image translation to generate more translational images with different shades and colors. However, the generated images have poor quality and limited diversity. Radford [22] proposed DCGAN, which uses convolution layers instead of dense layers and can produce high-resolution hyper-realistic images. Since its arrival, DCGAN has been a popular tool for data augmentation in medical imaging and fault diagnosis fields. Liang [23] used wavelet image samples of different bearing and gearbox load conditions as the input of the CNN classifier. After using the augmented image samples from DCGAN, the classifier performed more accurately for various load conditions and also obtained higher robustness. Owing to the strong support of the theoretical ground and reliable experimental analysis, DCGAN is deemed as the preferred GAN variety for data augmentation in this study.

Nevertheless, it is necessary to use a more suitable data augmentation technique that has the power of generating real-like samples to produce scalograms images that do not change their physical meaning. To achieve this goal, in this study, we proposed an ensemble augmentation method that uses the power of white noise in ensembles. The white noises are added to the original signal and averaged to vanquish the effect of the presence of white noise and generate real-like samples. Also, the proposed ensemble augmentation technique is compared with other existing methods of geometrical transforms such as flipping, noising, shifting, rotation, and others, along with the DCGAN to evaluate its performance.

For faster classification output, the TL strategy is introduced for classification in this study. The TL process is defined as using the learned weights of the previous classification in a new situation. Thus, a pre-trained TL model can be implemented to a similar task without additional training or fine-tuning, thereby saving training time. TL is a cutting-edge research field where substantial studies are being conducted for image recognition applications [15] [24]. In this study, we used three state-of-the-art TL models, namely, Inception-V3 [25], MobileNet-V2 [26], and ResNet50 [27], to justify our proposed method. All the models are pre-trained on the ImageNet dataset [28], which contains 1000 labeled classes. Since our intention is to justify the effectiveness of the proposed data
augmentation method, these TL learning models would facilitate obtaining the output performance within a short
time.

The main contributions of this study are inherent in the following:

1. A more fitting augmentation technique, namely, ensemble augmentation, is proposed for time-series data. The parameters selected for the ensemble augmentation is justified using a simulated signal.
2. The proposed method is validated with time-series bearing vibration using three state-of-the-art TL models where the inputs samples are scalogram images.
3. The performance of the proposed method is compared with classifiers without any augmented samples, DCGAN augmentation, and different geometric-based augmentation techniques.
4. Additional validation is conducted using noisy test samples at different SNR and test samples from different load conditions.

The rest of the paper is organized as follows: Section 2 provides the theoretical background for the scalogram samples used as the classifiers’ input. Some existing data augmentation techniques are described in section 3. The proposed augmentation technique is discussed in detail in section 4. Section 5 provides a brief discussion on the TL models. The experimental setup, classification design, data description and image generation are described in sections 6 to 10. Section 10 provides the results without any augmentation, and results with augmentation are provided in section 11. The robustness evaluation was mentioned in section 12. Section 13 ends with the conclusion.

2. CWT Scalogram Generation

CWT scalograms are one type of time-frequency representation that uses CWT analysis to obtain the energy density of the signal. In this study, the scalogram images are used as inputs for all classifiers. CWT decomposes the raw signal into a time scale which is represented by scaling and translating operations. Morlet wavelet is used as the mother wavelet because its shape is similar to the impulsive characteristics found in machine faults. The time-bandwidth product and symmetry parameter are set to 60 and 3, respectively [29].

For a given signal \( x(t) \), the CWT is obtained by integrating \( x(t) \) with the scaled and shifted from the mother wavelet \( \psi_{a,b}(t) \):

\[
W_{\psi}(a, b) = \int_{-\infty}^{\infty} x(t) \cdot \psi_{a,b}(t) dt
\]

Here the parameters ‘a’ and ‘b’ are the translation and dilation of the wavelets. By generating daughter wavelets \( \psi_{a,b}(t) \) from the mother wavelet \( \psi(t) \) more time-frequency information can be extracted, which is limited to finite space.

\[
\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left( \frac{t - b}{a} \right)
\]

Here, \( b \) is the time translation, and \( a \) is the dilation (scale) of the wavelet, and both are real numbers.
The colors in the scalogram plot show the relative values of the CWT coefficients. The light areas mean higher values of the CWT coefficients, and therefore, the signal is very similar to the wavelet. Whereas dark area means lower values of the CWT coefficients, and it shows that the corresponding time and scale versions of the wavelet are dissimilar to the signal. A red-green-blue (RGB) (3 channels) representation of the time-frequency image is better than a grayscale (1 channel) image because more channels contain higher information. So, only RGB scalograms are produced in this study. According to the range of energy of the wavelet in frequency and time, the minimum and maximum scales are automatically determined using 10 voices per octave [30]. Points out of the cone of influence have been handled by the approximation used in MathWorks MATLAB. The dimension of all the generated scalograms is $224 \times 224 \times 3$ pixels [31].

### 3. Existing Augmentation Techniques

Many transformation techniques have been conducted in previous research for DNNs. In this study, the comparable augmentation techniques fall into two main categories: DCGAN and Geometric transformation. The following subsections contain an elaborated description of those methods.

#### 3.1 DCGAN

The model architecture and the hyperparameters are primarily adopted from the original DCGAN paper [22]. The DCGAN model has a few deviations from the original GAN. The pooling layers are replaced with convolution layers. Batch normalization is used in the generator and discriminator. LeakyReLu activation function is used in all the layers except for the generator output layer, which is ‘tanh’. The generator model from the DCGAN has a 100-dimensional uniform distribution noise vector, $Z$, as input.

The other selected hyperparameters are also mostly followed from the paper. All weights were initialized from a zero-centered Normal distribution with a standard deviation (SD) of 0.02. In the LeakyReLU, the slope of the leak was set to 0.2 in all models. Adam optimizer is used with a learning rate of 0.0002 and momentum of 0.5. Binary cross-entropy is used as the loss function. The training samples from each fault type are given into the DCGAN model for the generator to imitate. Since we have very few training samples, the batch size while training DCGAN is only 32. It is challenging to generate fake samples using only a few training data. So, we have taken 10000 epochs to train the model and generated our desired number of fake samples once the highest number of epochs is reached. All the tuneable parameters are listed in Table 1.

| Hyperparameter name | Value               |
|---------------------|---------------------|
| Activation function | LeakyReLU           |
| Optimizer           | Adam                |
| Learning rate       | 0.0002              |
| Loss function       | Binary cross-entropy |
| Batch size          | 32                  |
| Epochs              | 10000               |
3.2 Geometric Transformation

The geometric transformation contains many types of augmentation techniques. The considered ones in this study are flipping, noise addition, shifting, rotation, zooming, shearing, brightening. The categories are described in the following subsections.

Two types of flipping are possible for image data generation, known as horizontal flipping and vertical flipping. It is more common to implement horizontal flipping than vertical flipping. This augmentation has proven helpful on datasets such as ImageNet, and CIFAR-10 is one of the easiest to implement. For vibration data representation using scalogram images, horizontal flipping makes more sense because a vertical flipping would change the meaning of a plot. Vertical flip is more suitable for samples like cosmology images, microscopic images.

In the geometric transformation-based noise augmentation technique, the noise is added directly to the input samples of the classifier. Different types of noise are added to the image samples like salt-pepper noise, Gaussian noise, speckle noise, periodic noise [32]. The neural network can look past through the noise and count the image as a new sample. This technique can help the classifier achieve more robustness for noisy test samples [33].

The shifting processing is conducted by moving all the images pixel in one direction while maintaining the same image dimension. In image data, horizontal and vertical shifting are possible. In this case, the shifted pixels will be cropped out of the images, and their empty places are filled with new pixel values from the neighboring pixels. However, intuitively, this will not perform very well for scalograms images because the important information may be missed from the cut-off regions.

Rotation augmentations are done by rotating the image right or left on an axis between 0° and 360°. The rotation will likely rotate pixels out of the image frame and leave areas of the frame with no pixel data that must be filled in.

The brightness of the image can be augmented by either randomly darkening images, brightening images, or both. The different level of lighting in the samples fools the classifier to count it as a new sample. Brightening the image means increasing the RGB values close to 255, and darkening means decreasing towards 0.

The zoom augmentation type randomly zooms in or out the images by either interpolating the pixels values or adding new pixel values to the corresponding pixels. The zoom is conducted uniformly from a random region of the sample.

The shear transform slants the shape of the object in the images. Shear transformation can be applied along X-axis and Y-axis. The X-shear changes the X coordinate values, and the Y-shear changes the Y coordinate values. For both cases, only one coordinate changes its value while preserving the other one.

The parameters for geometrical data augmentations are presented in Table 2. There are no parameters available for the horizontal and vertical flip. All the samples are flipped in both directions separately. Salt-pepper noise (30% Gaussian noise) [34] was added to the images for noise augmentation. The width shift is conducted within the range of 50% in the positive and negative direction along the x-axis from the middle. For the height shift augmentation, the same criterion as the width shift is applied but along the y-axis. In rotation shift augmentation, the rotational angle is 1° to 89°. Because if it is rotated at 90°, that would count as a vertical flip. For the brightness
parameter, we both lightened and darkened the pixel values up to 50% of their current value. Similarly, 50% zoom in and zoom out is conducted for the zoom augmentation. Lastly, the shear shifting is conducted within 45° angles for both the x-and y-axis. The angle is 45° here because a much higher value would distort the physical meaning of the sample.

Table 2 List of parameters for Geometric transformation Based data Augmentations.

| Augmentation type | Parameters                                      |
|-------------------|-------------------------------------------------|
| Horizontal flip   | N/A                                             |
| vertical flip     | N/A                                             |
| Noise             | Salt-pepper noise (30% Gaussian noise)          |
| Width shift       | right limit: 50%, left limit: 50%              |
| Height Shift      | Upper limit: 50%, lower limit: 50%             |
| Rotation          | Angle: 1° – 89°                                  |
| Brightness        | Up to 50% brighten and darken                   |
| Zoom              | Up to 50% zoom in and out                       |
| Shear             | Angle: 45° for both x-axis and y-axis           |

4. Proposed Ensemble Augmentation

The ensemble augmentation technique uses ensembles of white noise added directly to the original signal. The white noises are basically the random distribution of Gaussian noise, which coalesce the whole signal. Although it sounds counterintuitive, when the ensembles are added with the original signal and averaged, the random distribution of noise counters each other and obtains a new original signal alike augmented sample. The process is given as follows:

1. Obtain the original signal, $y_{m,n}$, where $m$ is the signal length and $n$ is the dimension which is 1 in the case of time-series vibration signal.
2. Define the upper bound, $Std_{\text{max}}$, and lower bound $Std_{\text{min}}$ of the noise signal SD.
3. Determine the particular noise SD $std_{i,j}$ of each individual ensemble using the following equation:

$$std_{i,j} = (Std_{\text{max}} - Std_{\text{min}}) \times \text{rand}(0~1)_{1,n} + Std_{\text{min}}$$

Where $i$ is the instance of ensemble and $j$ is the total number of ensembles. $\text{rand}(0~1)$ represents random number generation between 0 to 1.
4. Instantiate a Gaussian normal distribution of mean 0 and SD 1. Then, produce ensembles of white noise within the range of predefined SD $std_{i,j}$

$$x_{ij} = \text{rand}(0~1)_{m,n}$$

$$w^{1}_{i,j} = \left( x_{i,j} - \frac{1}{j} \sum_{i}^{j} x_{i} - \frac{1}{j-1} \sum_{i}^{j} \left( x_{ij} - \frac{1}{j} \sum_{i}^{j} x_{i} \right)^{2} \right)$$
Similarly, produce $w_{2,ij}$ using equation 4 and 5.

Here, the $w_{1,ij}$ and $w_{2,ij}$ are the desired Gaussian distribution.

5. Add the Gaussian distribution, $w_{i,j}$ with the original signal in positive and negative pairs.

$$y_{1,ij} = y_{m,n} + w_{1,ij}$$  \hspace{1cm} 6

$$y_{2,ij} = y_{m,n} - w_{2,ij}$$  \hspace{1cm} 7

6. Obtain the mean of all the ensembled positive and negative pairs.

$$y_1 = \frac{1}{j} \sum_{i} y_{1,i}$$  \hspace{1cm} 8

$$y_2 = \frac{1}{j} \sum_{i} y_{2,i}$$  \hspace{1cm} 9

7. Finally, obtain the augmented sample, $z$, by averaging the positive and negative pairs.

$$z = (y_1 + y_2)/2$$  \hspace{1cm} 10

8. Use the augmented signal, $z$, to generated scalogram images.

The pseudo-code for the ensemble data augmentation procedure is provided below:

```
[1] start
[2] Initialize the original signal, $y_{m,n}$.
[3] Define $Std_{\text{max}}$ and $Std_{\text{min}}$.
[4] for each ensemble $i$ of $j$ do
[5]    Determine $std_{i,j}$ using Eq. (3).
[6]    Calculate $w_{1,ij}$ and $w_{2,ij}$ from Eq. (4) and Eq. (5).
[7]    Obtain $y_{1,ij}$ and $y_{2,ij}$ from Eq. (6) and Eq. (7).
[8]    Update the value of sum of $y_{1,ij}$ and sum of $y_{2,ij}$.
[9] end for
[10] Calculate the mean ensemble, $y_1$ and $y_2$ from Eq. (8) and Eq. (9).
[11] return augmented signal, $z$, from Eq. (10).
[12] call function GenerateScalogram($z$)
[13] end
```

The tunable parameters of the ensemble augmentations are upper bound, $Std_{\text{max}}$ and lower bound $Std_{\text{min}}$ of the noise signal SD and the number of ensembles. If the upper bound is selected too high, the augmented samples might deviate too much from the main samples. This is equivalent to training the classifier with too much noisy samples. On the other hand, if the lower bound is too low, the generated samples will be almost identical to the original samples. In that case, there will be no variation in the augmented samples. So, the range of upper bound and lower bound of SD should be selected in such a way that the upper bound is not too high and the lower bound is not too low. The choice of the number of ensembles is another important factor because the more ensembles are used, the more precise the augmented samples become. However, a very large ensemble would take higher
computation time. On the other hand, if the ensemble number is too small, it would not be enough to cancel out the white noise.

The selection of parameters for the ensemble augmentation is validated by using a simulated signal and producing augmented samples of it. The simulated signal $X(n)$ consists of three components which are a continuous absolute tone plus a gapped one, namely, $X_1(n)$, $X_2(n)$ and $X_3(n)$, where they have a greater frequency, cosine signal, and a quadratic trend, respectively [35]. The three-component signals and the combination of them are obtained using the following equations and their waveforms in the time-domain are plotted in Figure 1.

\[
X(n) = X_1(n) + X_2(n) + X_3(n) 
\]

\[
X_1(n) = \begin{cases} 
0, & \text{if } 1 \leq n \leq 500 \\
\sin(2\pi \times 0.3(n - 501)), & \text{if } 501 \leq n \leq 750 \\
0, & \text{if } 751 \leq n \leq 1000 
\end{cases} 
\]

\[
X_2(n) = \cos(2\pi \times 0.05(n - 1)) 
\]

\[
X_3(n) = \frac{n}{1000} + \left(\frac{n}{1000}\right)^2 
\]

Figure 1 Samples from signals (a) $X_1(n)$; (b) $X_2(n)$; (c) $X_3(n)$; (d) $X(n)$ and (e) augmented sample.
In the simulated signal, the \( X(n) \) is obtained by combining the first three signals. Then, the augmented sample is generated, which looks very much alike of \( X(n) \). The experiment for the selection of parameters is conducted on this augmented signal to find out the suitable parameters. The original signal is considered as the reference signal. In order to select the suitable range of upper bound and lower bound, the value of correlation coefficient (CC) is considered. CC indicates how close the reference signal is to the original signal. A CC of 1 indicates the reference signal is exactly the same as the original signal, and a low CC indicates a weak correlation between the signals.

In Table 3, the CC values for different ranges of SD are obtained. Two different numbers of ensembles were used, i.e., 50 and 100. In both of them, for the SD range of 0.0 to 0.1, the CC value is 1, meaning the augmented signal is completely similar to the original signal. However, this is not the desired output because the classifiers will not get to learn any new features from the identical samples. For the SD range of 0.1 to 0.2, the CCs are 0.9993 and 0.9991, respectively, for 100 and 50 ensembles. These are good CCs because the augmented signals are not identical to the original signal and do not deviate too much from the original. After this range, for every next range, there is a steep fall in the CC values. So, choosing higher upper and lower bound values may produce a much-deviated signal from the original signal. The trend of CCs for both 50 and 100 ensembles is similar, indicating the number of ensembles does not affect the noise SD. Therefore, the values of \( \text{Std}_{\text{max}} \) and \( \text{Std}_{\text{min}} \) obtained in this study are 0.2 and 0.1, respectively, so that the new signals do not deviate much from the original one but still maintain some learnable characteristics.

The choice of the number of ensembles is another important factor because the more ensembles are used, the more precise the augmented samples become. However, a very large ensemble would take higher computation time. On the other hand, if the ensemble number is too small, it would not be enough to cancel out the white noise.

The signal-to-noise ratio (SNR) is adopted in this study to determine the ideal number of ensembles. The augmented signal is used to calculate the SNR value with respect to the original signal for the different number of ensembles using the following equation [36]:

\[
\text{SNR} = 10\log_{10}(p_1/p_2)
\]

Where, \( p_1 \) is the power of the augmented signal, and \( p_2 \) is the noise power obtained by subtracting original signal power from \( p_1 \). The SNR values for ensemble numbers of 10 to 200 are obtained in Figure 2. It can be seen that there is a sharp increase of SNR from 70 to 80 ensembles and another from 90 to 100 ensembles. However, after 100 ensembles, there is no significant change in the SNR values. So, increasing the ensemble numbers after 100 would just add liability to the computational time without any significant change in noise cancellation. Therefore, in this study, the number of ensembles is kept at 100.

Table 3 CCs for different ranges of SDs.

| SD range  | 0.0-0.1 | 0.1-0.2 | 0.2-0.3 | 0.3-0.4 | 0.4-0.5 | 0.5-0.6 | 0.6-0.7 | 0.7-0.8 | 0.8-0.9 | 0.9-1.0 |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CC (50 ensemble) | 1       | 0.9991  | 0.9978  | 0.9971  | 0.9959  | 0.9946  | 0.9933  | 0.9921  | 0.9912  | 0.9901  |
| CC (100 ensemble) | 1       | 0.9993  | 0.9981  | 0.9975  | 0.9968  | 0.9961  | 0.9953  | 0.9941  | 0.9932  | 0.9924  |
5. Pre-trained Classifiers

Three pre-trained classifiers, i.e., Inception-V3, MobileNet-V2, and Resnet50 applied in this study, are described in the following subsections.

5.1 Inception-V3

The Inception-V3 model is a superior classifier for object recognition to the previous model, GoogleNet (Inception-v1). The Inception-V3 model contains three main parts: the main convolutional layer, the improved Inception module, and the classifier. The main convolutional layer with max-pooling layers is used for feature extraction from the inputs. The improvements in the architecture are inspired by Network In Network [37], where multi-scale convolutions are conducted parallelly, and the results are concatenated. By using the auxiliary classifiers, better convergence and training accuracy is obtained. Moreover, the overfitting problems and vanishing gradients are avoided. 1 × 1 kernels are widely used in Inception-V3 to reduce the feature channels and speed up the convergence. Moreover, the large convolution is divided into small convolutions to reduce the number of parameters. So, the Inception-V3 model is a state-of-the-art object recognition architecture that is widely used in TL.

5.2 MobileNet-V2

The MobileNet-V2 is a lightweight architecture that has impressive accuracy. The model architecture includes a traditional convolutional layer and 17 inverse residual modules. Each inverse residual module has a 1 × 1 convolutional layer, 3 × 3 depth-wise separable convolutional layer, batch normalize, and ReLu activation function. The output feature map is connected to the input feature map while maintaining the size. This architecture can prevent gradient vanishing issue for a deeper neural network during the backpropagation process. In the vanilla MobileNet, the feature map is down-sampled by a 1/16-scale convolution layer which causes poor performance for small objects. In MobileNet-V2, this problem is overcome by applying 1/16 and 1/32 scales.

Figure 2 SNR values for different ensemble numbers.
Also, four inverse residual modules are included after the backbone network at 1/64, 1/128, 1/256, and 1/512 scales.

5.3 Resnet50

ResNet, the winner of the ImageNet classification competition in 2015, is a robust deep learning architecture. The ResNet50 is very effective for computer vision tasks as it has 50 processing layers. The ResNet50 architecture incorporates skip-connections to solve the vanishing gradient descent problem. The feature extraction operation is conducted by convolution and max-pooling layers, followed by stacked convolutions. Finally, the architecture has an average pooling layer and a fully connected layer. The ResNet50 has more than 23 million trainable parameters.

5.4 Parameters Selection

Training the TL models corresponds to a bunch of optimizable parameters. Among the tunable parameters, the most prominent are learning rate, optimizer, loss function, batch size. An ideal set of these parameter selections for an individual architecture leads to better classification performance. In terms of parameter tuning, we applied the grid search method [38] and selected a common set of parameters for all three classifiers. In all the classifiers, the shape of the input channel is $224 \times 224 \times 3$ because we used $224 \times 224$ pixels size of RGB images for all the classifiers. Adam optimizer is applied to optimize the loss function where the learning rate is set at 0.001. Cross-entropy is used as the loss function. The classifier is training for 50 epochs where the batch size is 32.

6. Experimental Setup

The CWRU bearing data has been using as the benchmark of bearing fault diagnosis study by many researchers [39]. Acceleration data was measured from a 2 hp reliance electric motor bearings. Drive end data is taken from bearing model SKF 6205-2RS JEM. The sampling rate is 12kHz. The rotor shaft's rotating speeds are 1797, 1772, 1750 and 1730 rpm and the motor loads are 0, 1, 2 and 3 hp. The setup of the test rig is shown in Figure 1.

![Experimental setup for the bearing data collection.](image-url)
7. Classification Design

Collected vibration signals include the following operating conditions: (1) normal condition, (2) inner race fault, (3) ball fault, and (4) outer race fault. Each fault condition includes three different fault sizes, 0.007, 0.014, and 0.021 inches. In total, 10 different conditions (1 normal, 9 fault conditions) were considered for the fault diagnosis and the fault categories are presented in Table 3.

Table 4 Different bearing fault conditions considered in this study.

| Fault | Normal | Inner race | Outer race | Ball |
|-------|--------|------------|------------|------|
| Severity | N/A | 0.007" | 0.014" | 0.021" | 0.007" | 0.014" | 0.021" |
| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

8. Data Description

Each bearing sample in this study contains 600 data points which provide 200 total samples for every condition. The training samples are collected from the first 100 samples of each class of 0 hp motor load condition. The validation set contains the following 50 samples, and the remaining 50 samples are allocated for testing. We also obtained 50 samples for each class from the load 1 hp, 2 hp and 3 hp conditions for the testing purpose only. The breakdown of the design of data distribution and partition is presented in Table 4.

Table 5 Data distribution from the datasets for different classes.

| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Load |
|-------|---|---|---|---|---|---|---|---|---|---|------|
| Train | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 0 |
| Validation | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 0 |
| Test | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 0 |
| Test | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 1 |
| Test | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 2 |
| Test | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 3 |

Table 6 Different steps of incrementation for augmented samples.

| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total augments |
|-------|---|---|---|---|---|---|---|---|---|---|----------------|
| Increment 1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 1000 |
| Increment 2I | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 2000 |
| Increment 2II | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 3000 |
| Increment 2III | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 4000 |
| Increment 2IV | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 5000 |

The augmented data is produced in two increments. First, the augmented samples are generated for every single sample in the training set. Since we used only samples from load 0 hp as the training data, all the augmented data are also produced using the no-load condition only. In the beginning, we have ten different augmentation
techniques to make the comparison. So, in the first increment, each technique gets 100 more augmented samples for every class using the existing 100 original train samples. Therefore, the number of original samples and augmented samples is the same in the first increment. In the second increment, we raise the number of fake samples in four steps [40] to observe the classifier performance. The increment is conducted in steps of 100 new augmented samples for each class. However, as it is not possible for geometric augmentations like horizontal and vertical flipping to produce fake data more than once, we combined all the types of geometric transformation as one to produce fake data more multiple times. At the end of increment 2, the number of total augmented train samples becomes 5000, whereas the original train samples are 1000. All the increment steps and the individual number of samples are presented in Table 5.

9. Image Samples Generation

All the image samples are generated as CWT RBG scalograms. The generated samples using augmentation techniques along with the sample from the original signal are presented in Figure 2. For visualization, only samples from the normal condition are presented.

Figure 4 Samples from different augmentations.

Comparing the augmented samples with the original signal as seen from Figure 1, the proposed ensemble augmented sample looks very alike to the original signal. The sample from DCGAN still has much presence of noise in it. The horizontal flip seems to be comparatively better; however, the higher coefficient regions of CWT, aka, the brightness regions, have been shifted, which may somewhat alter the meaning of the graph. The vertical flip probably changes the impulse location of the graph as it entirely changes its physical meaning. The noise augmentation may have concealed some information of the image sample. The width and height shift both lose some portion of the plot, which may contain important information. The rotational augmentation also cuts off some regions of the plot and changes its physical meaning by the degree of rotation. As the brightness augmentation dims or lightens the image, it ought to suffer identifying the higher coefficients from the lower ones correctly. The zoom and shear augmentation also lose some regions of the plot, and the shear augmentation changes the physical meaning because of distortion. In summary, only ensemble and horizontal augmentation are very close to the original sample. Moreover, the ensemble augmentation sample is almost identical to the original sample.
10. Classifiers’ Performance without Augmentation

First, the performance of the classifiers is evaluated using the original train samples without any data augmentation. All the output accuracies for the three classifiers are presented in Figure 3. The training accuracies for all three classifiers were 100%. The validation and test accuracy for the Inception-V3 was the lowest, 96.8% and 95.4%, respectively. The MobileNet-V2 model achieved 98.4% validation and 98% test accuracy. Lastly, the ResNet50 obtained 99% validation and 98.4% test accuracy. Also, the respective convergence curves of the classifiers are obtained in Figure 4. It shows only a small difference in the train and validation accuracy, meaning the models are significantly less overfit, indicating that the classifiers are well generalized for this task.

![Figure 5 Accuracy of the classifiers with no augmentation.](image)

11. Classifiers’ Performance with Augmentation

The results from the two steps of augmentation, i.e., first increment and second increment, are provided in the following subsections.

11.1 First Increment

Next, in increment 1, we add the same number of augmented training samples as the original train samples. This doubles the training samples volume. After training the three TL models with the new train data (original ones and the same number of the augmented ones), it is validated and tested with the previous validation and test samples. The results are obtained from a total of ten different types of augmentation techniques. The findings are presented in Figure 5, Figure 6, and Figure 7 for all the outputs. Here, in several cases, either the validation or the test accuracies appear to be the same. So, the validation-test average (VTA) accuracy was also counted as an important performance parameter to establish a clear distinction.

In Figure 5, the outputs from all the techniques show that only ensemble, horizontal flip, vertical flip, and width shift helped to improve the classifier performance. The highest validation accuracy is combinedly obtained by ensemble, horizontal flip and width shift which is 97.4%. However, only the ensemble augmentation obtained the highest test accuracy, i.e., 96.4%. Therefore, the ensemble augmentation technique has the highest VTA accuracy which is 96.9%, improved from 96.1%. The closest one was the VTA accuracy from the horizontal flip, which is
96.8%. The lowest accuracy was obtained by the brightness augmentation, which was 93.2% VTA accuracy. Apparently, ensemble augmentation along with the geometric transforms such as horizontal flip and width shift can retain most of the features in the augmented samples. So, for the Inception-V3 classifier the ensemble augmentation obtained the highest test and VTA accuracy, beating all the compared methods.

Figure 6 Model convergence curve from (a) Inception-V3 (b) MobileNet-V2 (c) ResNet50 classifiers.

Figure 6 presents the output accuracy from the MobileNet-V2 classifier. This time, most of the augmentations had more improved the classification accuracy than the no augmentation method. The highest validation accuracy is obtained by the ensemble augmentation which is 98.8%, followed by validation accuracy from DCGAN, horizontal flip, noise, width shift, height shift and rotation shift which are 98.6%. However, the highest test
accuracy was jointly obtained by the ensemble, horizontal flip, width shift and rotation shift. Nevertheless, the difference becomes apparent in VTA accuracy where the highest, 98.7%, is obtained by the ensemble augmentation. The second highest VTA accuracy is obtained combinedly by the horizontal flip, width shift and rotation shift. Therefore, the proposed ensemble augmentation also obtained the validation and VTA accuracy in MobileNet-V2 classifier.

The outputs from ResNet50 are illustrated in Figure 7. The ensemble augmentation obtained the highest validation accuracy which is 99.2%. The second highest validation accuracy, 99%, is obtained by the DCGAN, width shift and shear augmentation all together. Compared to the validation accuracy of original samples which is 98.6%, these are significant improvements. The test accuracy for the original samples was 98.2% and its improved as high as 98.4% for the ensemble and horizontal flip augmentations. However, for the VTA accuracy, only the ensemble augmentation obtained the highest accuracy which is 98.8%. The second highest accuracy is 98.6%
obtained by width shift and shear augmentation. So, the VTA accuracy improved from a 98.4% to highest 98.8% using the proposed augmentation technique.

Figure 9 Performance of ResNet50 on different data augmentation techniques.

In summary, for most of the classifiers, there were multiple augmentations that obtained the highest validation accuracy and test accuracy. However, after considering the VTA accuracy, in all of them the ensemble augmentation achieved the highest accuracy. Although, these results primarily establish the superiority of the ensemble augmentation, some additional analyses are also conducted. To have better insights into the classifiers’ performance and breakdown of output accuracy, we obtained all the confusion matrices (CMs) from all the classifiers of the first increment of data augmentation along with the no augmentation models. Furthermore, the precision values from all the CMs are also obtained for comparison of each individual class. In this study, we highlighted the precision values because it is more important to identify the particular fault type in condition monitoring correctly.

Table 7 Class precision of Inception-V3 classifier.

| Class          | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------------|---|---|---|---|---|---|---|---|---|---|
| No Aug.        | 1 | 1 | 0.96 | 1 | 1 | 0.94 | 0.98 | 0.90 | 0.94 | 0.82 |
| Ensemble       | 1 | 1 | 0.98 | 1 | 1 | 0.94 | 0.98 | 0.92 | 0.94 | 0.88 |
| DCGAN          | 1 | 1 | 0.98 | 1 | 1 | 0.90 | 0.98 | 0.92 | 0.88 | 0.74 |
| Horizontal flip| 1 | 1 | 0.98 | 1 | 1 | 0.94 | 0.98 | 0.90 | 0.94 | 0.88 |
| Vertical flip  | 1 | 1 | 0.96 | 1 | 1 | 0.94 | 0.98 | 0.90 | 0.90 | 0.88 |
| Noise          | 1 | 1 | 0.98 | 1 | 1 | 0.94 | 0.96 | 0.92 | 0.92 | 0.88 |
| Width shift    | 1 | 1 | 0.98 | 1 | 1 | 0.94 | 0.98 | 0.92 | 0.92 | 0.88 |
| Height shift   | 1 | 1 | 0.96 | 1 | 1 | 0.94 | 0.98 | 0.90 | 0.84 | 0.86 |
| Rotation shift | 1 | 1 | 0.96 | 1 | 1 | 0.92 | 0.98 | 0.90 | 0.88 | 0.86 |
| Brightness     | 1 | 1 | 0.92 | 1 | 1 | 0.88 | 0.96 | 0.90 | 0.82 | 0.76 |
| Zoom           | 1 | 1 | 0.98 | 1 | 1 | 0.90 | 0.98 | 0.90 | 0.92 | 0.88 |
| Shear          | 1 | 1 | 0.98 | 1 | 1 | 0.92 | 0.98 | 0.88 | 0.92 | 0.84 |
In Figure 8, the CMs from all three classifiers are presented. A close observation shows that our proposed ensemble augmentation technique achieved the highest individual correctly identified samples in all three
classifiers. This inevitably means the proposed method is not only accurate but also precise. Also, it indicates that our proposed method has the highest precision value for every class among all the classifiers. For a better understanding, next, the precision value tables are obtained separately for Inception-V3, MobileNet-V2, and ResNet50, respectively.

Table 8 Class precision of MobileNet-V2 classifier.

| Class     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----------|---|---|---|---|---|---|---|---|---|---|
| No Aug.   | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.92 | 0.92 |
| Ensemble  | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.96 | 0.94 |
| DCGAN     | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.94 | 0.94 |
| Horizontal flip | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.96 | 0.94 |
| Vertical flip | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.94 | 0.90 |
| Noise     | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.94 | 0.94 |
| Width shift | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.96 | 0.94 |
| Height shift | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.94 | 0.92 |
| Rotation shift | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.96 | 0.94 |
| Brightness | 1 | 1 | 1 | 1 | 1 | 0.96 | 1 | 0.98 | 0.96 | 0.92 |
| Zoom      | 1 | 1 | 1 | 1 | 1 | 0.96 | 1 | 0.98 | 0.96 | 0.92 |
| Shear     | 1 | 1 | 1 | 1 | 1 | 0.94 | 1 | 0.94 | 0.96 | 0.92 |

Table 9 Class precision of ResNet50 classifier.

| Class     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----------|---|---|---|---|---|---|---|---|---|---|
| No Aug.   | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.94 | 0.96 |
| Ensemble  | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.94 | 0.96 |
| DCGAN     | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.94 |
| Horizontal flip | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.94 | 0.94 |
| Vertical flip | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.94 | 0.92 |
| Noise     | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.96 |
| Width shift | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.96 |
| Height shift | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.94 | 0.94 | 0.96 |
| Rotation shift | 1 | 1 | 0.96 | 1 | 1 | 1 | 0.94 | 0.94 | 0.84 | 0.92 |
| Brightness | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.94 | 0.94 |
| Zoom      | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.94 |
| Shear     | 1 | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.96 |

In Table 6 for the Inception-V3 classifier, the precision values of ensemble augmentation are the highest among all the methods. Moreover, no method obtained greater or similar precision values than ensemble augmentation for all the classes. However, classes 0, 1, 3, and 4 achieved the ideal precision value in all the methods. In Table 7, for the MobileNet-V2 classifier, the ensemble, horizontal flip, Width shift, and Rotation shift obtained the highest precision values for every class. This time, class 0 to 5 and class 6 obtained the ideal precision value in
all the classes. In Table 8 for the ResNet50 classifier, only the ensemble and horizontal augmentations obtained the highest precision values in all the classes. In the model, classes 0, 1, 3, 4 performed with ideal precision in all the methods. From the tables, it is seen that the ensemble augmentation technique obtained the highest precision for all classes in all three classifiers. Some other augmentation techniques also obtain the highest precision but not in all the models. However, for comparatively better-performing methods like horizontal flip, it is not possible to do any more augmentations. It is also conspicuous that using the ensemble technique, precision increases significantly than the no augmentation classifier. Nevertheless, since some of the methods provide the same test accuracy and precision values, additional evaluation is required to justify the effectiveness of the proposed augmentation method. Moreover, it is important to investigate the performance of the augmentation techniques for the increasing and optimum number of samples.

11.2 Second Increment

The second increment is conducted to investigate the classifier performance for the increasing augmented samples and also to identify the optimum number of samples. Here, the number of augmented training samples is increased within a predefined range. For ensemble and DCGAN based augmentation, it is possible to produce augmented samples as required because these methods work with a random distribution of noise. However, for some geometrical transformation-based augmentations like horizontal flip and vertical flip, it is impossible to perform augmentation more than once. So, in this step, all the geometrical transformation methods are counted as one entity for augmentation [41], and the samples are increased accordingly. Therefore, in total, three different augmentation methods will be compared in this section. All the outputs from the classifiers are listed in Table 9, Table 10 and Table 11.

Table 10: Inception-V3 model performance for different numbers of augmented samples.

| Augments/class | Ensemble | DCGAN | Geometric transform |
|----------------|----------|-------|---------------------|
|                | Train    | Val.  | Test    | VTA    | Train    | Val.  | Test    | VTA    |
| 100 (increment 1) | 97.4     | 96.4  | 96.9    | 99.8   | 96.8     | 96.8  | 94      | 95.4   |
| 200 (increment 2I) | 97.8     | 96.6  | 97.2    | 99.68  | 96.8     | 95    | 95.9    | 98.45  |
| 300 (increment 2II) | 98.2     | 96.8  | 97.5    | 99.08  | 95.8     | 94.8  | 95.3    | 98.36  |
| 400 (increment 2III) | 98.8     | 97.2  | 98      | 98.7   | 95.8     | 94.6  | 95.2    | 97.74  |
| 500 (increment 2IV) | 98.8     | 97.4  | 98.1    | 98.9   | 95       | 94    | 94.5    | 97.45  |

The effect of increased augmented training samples on Inception-V3 classifier performance is presented in Table 9. It is observed that for the ensemble augmentation, the VTA accuracy improved for each step of increment. At increment 2IV, where the augmented samples per class are 500, the validation accuracy is 98.8% and the highest test accuracy is obtained which is 97.4%. At this stage, the VTA accuracy is the highest, 98.1%, for this predefined range of augmented samples. In the DCGAN augmentation, the validation accuracy remains the same at the first stage of second increment, but the test accuracy improves from 94% to 95%. At this stage, the highest VTA accuracy of 95.9% is obtained for DCGAN. However, for all the following stages of the second increment, the
VTA accuracy keeps decreasing. The possible reason is the outputs from the DCGAN contain much presence of white noise. So, if the classifier gets most training data containing considerable noise, it cannot generalize well for the test samples. It is almost equivalent to training the classifier with a low signal-to-noise ratio (SNR). As a result, we see the fall of validation, test and VTA accuracy in each step of increment using DCGAN. The validation and test accuracy fall from 96.8% and 95% to 95% and 94%. For the geometric transformation, the output accuracy decreases for each increment. Because it changes the physical meaning of scalogram samples of vibration signal and more augmented samples reduce the accuracy. As the number of augmented samples increases, the training samples pollute with more distorted graphical images. So, the classifier accuracy falls very low with every increment of geometrically augmented samples. This time the validation and test accuracy fall from 91.4% and 89.4% to 86.6% and 86.2%. The highest VTA accuracy is obtained at the first stage of increment, which is 90.4% and the lowest is obtained at the last stage, 86.4%. The step at which the model performs the best is highlighted for each augmentation technique, and at that point, it is considered the best number of training samples for that classifier. For the proposed method, a suitable number is obtained at increment 2IV, where there are 500 augmented samples per class and the total augmented samples are 5000, making the total training samples (original and augmented) 6000. For the DCGAN model, the turning point is at increment 2I where per class samples are 200, and total augmented training samples are 2000. However, in geometric transform, the accuracy decreases at the first stage, so the samples from its first step are taken as the turning point. Here, the number of samples per class is 100, and the total augmented training samples are 1000. So, for the given range of augmented samples, the test accuracy for ensemble augmentation increased in every step of increment, but for the other augmentations, the test accuracy decreased after a certain increment.

### Table 11: MobileNet-V2 model performance for different numbers of augmented samples.

| Augments/class   | Ensemble | DCGAN | Geometric transform |
|------------------|----------|-------|---------------------|
|                  | Train    | Val.  | Test | Train | Val. | Test | Train | Val. | Test | VTA |
| 100 (increment 1) | 100      | 98.8  | 98.6 | 98.7  | 99.86 | 98.6 | 98.4 | 98.5  | 99.90 | 98.2 | 98 | 98.1 |
| 200 (increment 2I)| 100      | 98.8  | 98.6 | 98.7  | 99.83 | 98.6 | 98.2 | 98.4  | 100   | 98  | 98 | 98   |
| 300 (increment 2II)| 100     | 98.8  | 98.6 | 98.7  | 99.87 | 98.4 | 98   | 98.2  | 99.76 | 97.8 | 97.6 | 97.7 |
| 400 (increment 2III)| 100   | 99    | 98.8 | 98.9  | 99.76 | 98.4 | 97.8 | 98.1  | 99.74 | 97.4 | 97.4 | 97.4 |
| 500 (increment 2IV)| 100    | 99    | 98.8 | 98.9  | 99.83 | 98.4 | 97.8 | 98.1  | 99.76 | 97.4 | 97.4 | 97.4 |

For the MobileNet-V2 classifier (Table 10), the accuracy decreases in every step for both DCGAN and geometric transformation augmentation. In DCGAN based augmentation, the validation and test accuracy fall from 98.6% and 98.4% in the first step to 98.4% and 97.8% in the last step. Also, the highest VTA accuracy obtained is 98.5%, and the lowest is 98.1%. The validation and test accuracy fall from 98.2% and 98% to 97.4% and 97.4% for geometric transformation augmentation. Here, the highest VTA accuracy was obtained in the first stage of increment. Using the proposed ensemble augmentation, the validation and test accuracy remain constant until increment 2II. The validation and test accuracy improved at increment 2III but remains the same in 2IV. In this case, the validation and test accuracy improve from 98.8% and 98.6% to 98.8% and 98.8%. Therefore, the VTA
accuracy raises from 98.7% to 98.9%. Although the performance of increment $2III$ and increment $2IV$ are the same, we picked increment $2III$ as the best step. Because at increment $2III$ the number of augmented samples per class is 400, and the total augmented samples are 4000. This number is much lower than the augmented samples in increment $2IV$ and would, therefore, take less classifier training time. At this stage, the total training samples (original and augmented) of the classifier is 5000.

Table 12 ResNet50 model performance for different numbers of augmented samples.

| Augments/class | Ensemble | DCGAN | Geometric transform |
|----------------|----------|-------|---------------------|
|                | Train    | Val.  | Test    | VTA  | Train    | Val.  | Test    | VTA  | Train    | Val.  | Test    | VTA  |
| 100 (increment 1) | 100 | 99.2 | 98.4 | 98.8 | 100 | 99 | 98 | 98.5 | **99.95** | **98.0** | **96.8** | **97.4** |
| 200 (increment 2I) | 100 | 99.4 | 98.6 | 99 | **99.97** | **99.2** | **98** | **98.6** | 100 | 97.6 | 96.2 | 96.9 |
| 300 (increment 2II) | **100** | **99.4** | **99.2** | **99.3** | 99.8 | 98.8 | 97.8 | 98.3 | 100 | 97.6 | 96 | 96.8 |
| 400 (increment 2III) | 100 | 99.4 | 99.2 | 99.3 | 99.96 | 98.6 | 97.8 | 98.2 | 99.90 | 97.4 | 96 | 96.7 |
| 500 (increment 2IV) | 100 | 99.4 | 99.2 | 99.3 | 99.83 | 98.4 | 97.8 | 98.1 | 99.95 | 97 | 96 | 96.5 |

Lastly, the ResNet50 classifier performance for increased training samples is presented in Table 11. Here, the validation accuracy of DCGAN improves in increment 2I, and test accuracy remains the same as the first increment. Also, the highest VTA accuracy is obtained at this stage which is 98.6%. However, in the next three increments, the validation accuracy falls in every step, and test accuracy becomes fixed at 97.8%. The validation and test accuracy fall from 99.2% and 98% to 98.4% and 97.8 using the augmented training samples from DCGAN. Here, the VTA accuracy falls as low as 98.1%. For the geometric transform, the performance falls in each step of increment. The validation and test accuracy fall from 98% and 96.8% to 97% and 96%. The highest VTA accuracy is 97.4%, and the lowest is 96.5%. On the other hand, in the proposed method, for 300 augmented samples/class, the validation and test accuracy hit the highest, 99.4% and 99.2%. In the next two steps, there is no change in the performance. Here, the highest VTA accuracy is 99.3%. So, we selected the third step, i.e., 300 augmented samples/class, for further analysis. At this stage, the total augmented samples are 3000, and the total training (original and augmented) samples are 4000. For DCGAN, the turning point is at increment 2I which have 200 augmented samples/class. On the other hand, increment 1 is the turning point for the geometric transform as this method does not improve the performance.

Some additional analysis is conducted from the results of the second increment. In this analysis, the CM from the three augmentation techniques, i.e., ensemble augmentation, DCGAN, and geometric transformation, are obtained. Moreover, the precision values for each of them are also compared to highlight the difference in performance. The CMs from all the models in the second increment is presented in Figure 9. It can be observed that the proposed ensemble augmentation correctly identified the highest number of classes in all the three classifier models. It inevitably indicates the superiority of the proposed augmentation over DCGAN and geometric transform.
Table 13 Precision values for increased augmented samples of the Inception-V3 classifier.

| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|---|---|---|---|---|---|---|---|---|---|
| Ensemble | 1 | 1 | 0.98 | 1 | 1 | 0.94 | 1 | 0.94 | 0.94 | 0.94 |
| DCGAN | 1 | 1 | 0.96 | 1 | 1 | 0.92 | 0.98 | 0.90 | 0.88 | 0.86 |
| Geometric Transform | 1 | 0.96 | 0.96 | 1 | 1 | 0.78 | 0.98 | 0.84 | 0.78 | 0.64 |

Table 14 Precision values for increased augmented samples of the MobileNet-V2 classifier.

| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|---|---|---|---|---|---|---|---|---|---|
| Ensemble | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 1 | 0.96 | 0.94 |
| DCGAN | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.94 | 0.94 |
| Geometric Transform | 1 | 1 | 1 | 1 | 1 | 0.98 | 1 | 0.98 | 0.92 | 0.92 |

Table 15 Precision values for increased augmented samples of the ResNet50 classifier.

| Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|---|---|---|---|---|---|---|---|---|---|
| Ensemble | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.96 | 0.98 | 1 |
| DCGAN | 1 | 1 | 1 | 1 | 1 | 0.96 | 0.96 | 0.94 | 0.94 |
| Geometric Transform | 1 | 1 | 0.98 | 1 | 1 | 0.94 | 0.98 | 0.92 | 0.94 | 0.92 |

In Table 12, Table 13 and Table 14, the precision values from the three augmentation methods using the three classifiers are presented. In all of them, the ensemble augmentation obtained much higher precision for all the classes than the other two methods. In the second increment, the difference in precision has also increased in the ensemble method from DCGAN and geometric transform. It shows that by increasing the augmented samples...
using the ensemble technique, the classifier performance never decreases while it may decrease for the other techniques.

12. Robustness Evaluation

In the real application, when predicting new data, the data can be quite deviated from the original signal or may contain high noise. So, it is essential to ensure that the model can still hold good robustness when classifying new data. Moreover, the machines may not always run at a constant speed due to changes in voltage or current. These unexpected situations make fault diagnosis more challenging. In this section, we tested how robust and generalized the classifiers become by training on the original and best number of augmented samples combined from the previous section. The trained models are obtained from the selected best-performing models from the second increment of augmentation along with the models without any augmentation. However, for the geometric transform augmentations, the best models are the ones from the first increment only (1000 original and 1000 augmented training samples) because the accuracy decreases in the next steps.

The results are obtained from two different analyses:

1. Addition of different levels of SNR to the test samples.
2. Using test samples from a different motor load.

12.1 Robustness in noisy Data

In this step, different levels of SNR values are added to the test signals, and scalograms are generated from those noisy signals. The SNR range is -8 dB to 10 dB [23], which gives 10 different levels of noisy test samples in total. The comparisons are conducted among the no augmentation, ensemble, DCGAN, and geometric transformation models. All the results are presented in Figure 10, Figure 11 and Figure 12 for the three classifiers.

![Figure 12 Performance of Inception-V3 for different SNRs.](image-url)
In Figure 10, the Inception-V3 classifier’s performance for different SNR of test samples is provided. The results show that the robustness for the ensemble augmentation model was the highest throughout. The next position is followed by original samples where no augmentation was conducted. The DCGAN augmentation performance was very near to the no augmentation except for at the 4dB SNR, where it was significantly lower. The robustness of geometric augmentation was the lowest for most of the SNR values.

In Figure 11, for the MobileNet-V2 model, the proposed ensemble augmentation also achieved the highest robustness for all the values of SNR. However, the test accuracy for different SNRs has been seen to intertwine for the other three methods, i.e., no augmentation, DCGAN, and geometric transform. Furthermore, the robustness of the no augmentation samples is much lower, from 6dB to 10 dB.

In Figure 12, for the ResNet50 model, the robustness of the ensemble augmentation was also highest as well, but with a slight difference from the DCGAN all the way. The robustness of geometric transform was close too but
fell little behind after 0dB SNR. In this model, the robustness of no augmentation was remarkably lower than all the other methods, and the difference increases for a higher SNR value.

In summary, the ensemble augmentation had the highest test accuracy for different SNR values in all three classifiers. The other three methods, i.e., no augmentation, DCGAN and geometric transform, had varying and intertwined test accuracies. All the plots show the supremacy of ensemble augmentation.

12.2 Robustness in Different loads

The robustness test is also conducted using different motor load test samples in this section. The loads are 1 hp, 2 hp and 3 hp. The purpose is to see if the proposed methods can still obtain high robustness for different load data. The data is supposed to be more deviated as the load increases and thus making the task more challenging. The performance of Inception-V3, MobileNet-V2 and ResNet50 models are presented for the no augmentation, ensemble, DCGAN and geometric transform augmentations are presented in Figure 13, Figure 14 and Figure 15, respectively.

Figure 15 Performance of Inception-V3 for different loads.

Figure 16 Performance of MobileNet-V2 for different loads.
In Figure 13, for the Inception-V3 model, the highest test accuracy for 1 hp was 88% which was obtained by both the no augmentation and ensemble methods. The highest test accuracies for the 2 hp and 3 hp were also obtained by the ensemble method, which was 87.6% and 77.6%, respectively. It can be noted that for all three loads, the chronological order of robustness using Inception-V3 classifier was ensemble, no augmentation, DCGAN and lastly, geometric transform.

In Figure 14, the MobileNet-V2 model’s robustness for different loads is presented. For the 1 hp and 2 hp loads, the highest robustness was acquired by ensemble augmentation, followed by the DCGAN ones. In the case of 1 hp load, the test accuracy of the ensemble method is 91.8% and DCGAN’s is 91.6%. For the 2 hp, it is 91.6% and 90.4% using ensemble and DCGAN, respectively. On the other hand, for the load 3 hp, the highest accuracy was obtained by the ensemble augmentation, which is 77.2%. The second highest is geometric transform with 76.6% accuracy, and the lowest is the no augmentation with 76.2% accuracy.

In Figure 15, for the ResNet50 model, the ensemble augmentation achieved the highest accuracy as well for all three loads. For the 1hp, the highest accuracy was 91.6%, and the lowest one was 88% obtained by the original samples. In 2 hp test result, the highest accuracy is 87.4%, but this time the accuracy from the geometric transform was the lowest. For the 3 hp load, the highest accuracy obtained by the ensemble is 76.8%, and the lowest one was from DCGAN, which is 75%.

To sum it up, the model with ensemble augmentation performed with the highest test accuracy for all three different load samples. In all three classifier models, the test accuracy using ensemble augmentation improved from the no augmentation samples, but in some cases of DCGAN and geometric augmentation, the test accuracy decreased from the no augmentation samples. It can be concluded that the model with ensemble augmentation can still hold high robustness for test samples of different loads.

13. Conclusion

In this study, a noise-assisted ensemble augmentation technique is proposed and compared with other prevailed augmentations. The ensemble augmentation is targeted primarily for 1-D time-series data, where vibration data is
a very common type. So, the experimental validation is conducted with bearing vibration data for fault diagnosis. Total ten different bearing condition is classified using test samples from three load conditions. CWT scalograms are generated from the vibration data as the samples, and the augmentations are conducted on the training samples. The augmented samples are added in two increments, where the first step adds the sample number of fake samples, and the amount is increased in the second increment. Three TL classifiers are applied for validating the performance. The output results show that our proposed augmentation technique greatly improves the classifier accuracy than when there is no augmentation done. The ensemble method also performs better than all other common augmentations. Furthermore, unlike other techniques, increasing the augmented samples using ensemble does not reduce the classifier performance. Therefore, ensemble augmentation can be applied to produce multiples of desired augmented samples from the limited training samples. The performance is also evaluated for noisy test samples and test samples from different loads. The results show that classifiers trained with ensemble augmented samples can achieve higher robustness than other augmentation techniques in this study.

Acknowledgment

The authors are grateful to the Ministry of Higher Education Malaysia for the financial support provided under the Fundamental Research Grant Scheme (FRGS) No. FRGS/1/2018/TK03/UMP/02/24 grant (University reference RDU190157). The authors also acknowledge the Universiti Malaysia Pahang for their support through the Master Research Scheme (MRS) scholarship. Additional funding for this research also came from the Institute of Noise and Vibration, Universiti Teknologi Malaysia (RDU192303).

REFERENCES

[1] J. K. Sinha and A. R. Rao, “Vibration based diagnosis of a centrifugal pump,” Struct. Heal. Monit., vol. 5, no. 4, pp. 325–332, Dec. 2006, doi: 10.1177/1475921706067760.
[2] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” Mech. Syst. Signal Process., vol. 108, pp. 33–47, 2018, doi: 10.1016/j.ymssp.2018.02.016.
[3] P. Gangsar and R. Tiwari, “Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review,” Mech. Syst. Signal Process., vol. 144, p. 106908, 2020, doi: 10.1016/j.ymssp.2020.106908.
[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.
[5] C. C. Chen, Z. Liu, G. Yang, C. C. Wu, and Q. Ye, “An improved fault diagnosis using 1d-convolutional neural network model,” Electron., vol. 10, no. 1, pp. 1–19, 2021, doi: 10.3390/electronics10010059.
[6] I.; Mitiche, A.; Neshrit, S.; Conner, P.; Boreham, Morrisin, and Gordon, “D-CNN based real-time fault detection system for power asset diagnostics,” doi: 10.1049/iet-gtd.2020.0773.
[7] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Oct. 2018, vol. 11141 LNCS, pp. 270–279, doi: 10.1007/978-3-030-01424-7_27.
[8] D. T. Hoang and H. J. Kang, “Rolling element bearing fault diagnosis using convolutional neural network and vibration image,” Cogn. Syst. Res., vol. 53, pp. 42–50, 2019, doi: 10.1016/j.cogsys.2018.03.002.
[9] P. Ma, H. Zhang, W. Fan, C. Wang, G. Wen, and X. Zhang, “A novel bearing fault diagnosis method based on 2D image representation and transfer learning-convolutional neural network,” Meas. Sci. Technol., vol. 30, no. 5, p. 055402, May 2019, doi: 10.1088/1361-6501/ab0793.
[10] Y.-M. Hsueh, V. R. Ittangihal, W.-B. Wu, H.-C. Chang, and C.-C. Kuo, “Fault Diagnosis System for Induction Motors by CNN Using Empirical Wavelet Transform,” Symmetry (Basel.), vol. 11, no. 10, p. 1212, Sep. 2019, doi: 10.3390/sym11101212.
[11] H. Nasifoglu and O. Erogul, “Convolutional Neural Networks based OSA Event Prediction from ECG Scalograms and Spectrograms.”
[12] S. Jayalakshmy and G. F. Sudha, “Scalogram based prediction model for respiratory disorders using optimized convolutional neural networks,” Artif. Intell. Med., vol. 103, p. 101809, Mar. 2020, doi: 10.1016/j.artmed.2020.101809.
[13] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, “Deep Learning Enabled Fault Diagnosis Using Time-Frequency Image Analysis of Rolling Element Bearings,” 2017, doi: 10.1155/2017/5067651.
[14] D. T. Hoang, X. T. Tran, M. Van, and H. J. Kang, “A Deep Neural Network-Based Feature Fusion for Bearing Fault Diagnosis,” Sensors, vol. 21, no. 1, p. 244, Jan. 2021, doi: 10.3390/s21010244.
[15] M. Elgend et al., “The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric
Transformation Perspective,” Front. Med., vol. 8, p. 629134, Mar. 2021, doi: 10.3389/fmed.2021.629134.

[16] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, “Make Some Noise. Unleashing the Power of Convolutional Neural Networks for Profiled Side-channel Analysis,” IACR Trans. Cryptogr. Hardw. Embed. Syst., pp. 148–179, 2019, doi: 10.46586/techs.v2019.i3.148-179.

[17] I. J. Goodfellow et al., “Generative adversarial nets,” in Advances in Neural Information Processing Systems, 2014, vol. 3, no. January, pp. 2672–2680, doi: 10.3156/jsoft.29.5_177_2.

[18] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein Generative Adversarial Networks,” PMLR, Jul. 2017. Accessed: May 03, 2021. [Online]. Available: http://proceedings.mlr.press/v70/arjovsky17a.html.

[19] Z. Wang, J. Wang, and Y. Wang, “An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition,” Neurocomputing, vol. 310, pp. 213–222, Oct. 2018, doi:10.1016/j.neucom.2018.05.024.

[20] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in Proceedings of the IEEE International Conference on Computer Vision, Dec. 2017, vol. 2017-October, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[21] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016.

[22] P. Liang, C. Deng, J. Wu, and Z. Yang, “Intelligent fault diagnosis of rotating machinery via wavelet transform, generative adversarial nets and convolutional neural network,” Meas. J. Int. Meas. Confed., vol. 159, p. 107768, 2020, doi: 10.1016/j.measurement.2020.107768.

[23] D. Han, Q. Liu, and W. Fan, “A new image classification method using CNN transfer learning and web data augmentation,” Expert Syst. Appl., vol. 95, pp. 43–56, Apr. 2018, doi: 10.1016/j.eswa.2017.11.028.

[24] I. Kandel and M. Castelli, “Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review,” Applied Sciences (Switzerland), vol. 10, no. 6. MDPI AG, p. 2021, Mar. 01, 2020, doi: 10.3390/app10060201.

[25] C. Szegedy et al., “Going deeper with convolutions,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Oct. 2015, vol. 07-12-June-2015, pp. 1–9, doi: 10.1109/CVPR.2015.7289584.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image classification,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[28] “ImageNet.” https://www.image-net.org/ (accessed Jun. 28, 2021).

[29] J. Lin and L. Qu, “Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis,” J. Sound Vib., vol. 234, no. 1, pp. 135–148, Jun. 2000, doi: 10.1016/j.jsvi.2000.2864.

[30] M. Misiti, G. Oppenheim, J.-M. Poggi, and Y. Misiti, “Wavelet Toolbox Documentation,” 2001, Accessed: Jul. 01, 2021. [Online]. Available: https://www.mathworks.com/help/wavelet/ref/cwt.html.

[31] C. Torrence and G. P. Compo, “A Practical Guide to Wavelet Analysis,” Bull. Am. Meteorol. Soc., vol. 79, no. 1, pp. 61–78, 1998, doi: 10.1175/1520-0477(1998)079<0061:APGWTW>2.0.CO;2.

[32] C. Bonceler, “Image Noise Models,” in The Essential Guide to Image Processing, Elsevier Inc., 2009, pp. 143–167.

[33] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, “Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness,” Inf. Sci. (Ny.), vol. 247, pp. 1–20, Oct. 2013, doi: 10.1016/j.ins.2013.06.002.

[34] L. Jiao, R. Shang, F. Liu, and W. Zhang, “Multiobjective optimization algorithm-based image segmentation,” in Brain and Nature-Inspired Learning Computation and Recognition, Elsevier, 2020, pp. 301–349.

[35] M. Lin, Q. Chen, and S. Yan, “Network in network,” Dec. 2014, Accessed: Jul. 05, 2021. [Online]. Available: https://arxiv.org/abs/1312.4400v3.

[36] E. Irmak, “Implementation of convolutional neural network approach for COVID-19 disease detection,” Physiol. Genomics, vol. 52, no. 12, pp. 590–601, Dec. 2020, doi: 10.1152/physiolgenomics.00084.2020.

[37] C. W. R. University, “Bearing Data Center,” University, Case Western Reserve. https://csegroups.case.edu/bearingdatacenter/home (accessed May 05, 2021).