APST

Asia-Pacific Journal of Science and Technology

https://www.tci-thaijo.org/index.php/APST/index

Published by the Research and Graduate Studies,
Khon Kaen University, Thailand

A deep learning model for air leak detection from a pipe fitting using an accelerometer

Thanakrit Kraising[1], Winai Wongthai[1,2], Thanathorn Phoka[1,2,*], Aimaschana Niruntasukrat[3] and Nuttanat Ruttanapahat[4]

[1]Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok Thailand
[2]Research Center for Academic Excellence in Nonlinear Analysis and Optimization, Naresuan University, Phitsanulok, Thailand
[3]NECTEC, National Science and Technology Development Agency (NSTDA), Pathumthani, Thailand
[4]Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand
[*]Corresponding author: thanathornp@nu.ac.th

Received 23 December 2021
Revised 14 February 2022
Accepted 15 April 2022

Abstract

Gas leaks from fittings of a pneumatic pipe system result in the breakdown or failure of the system. It's about half of the losses of output from production lines in the manufacturing sector. Deep learning (DL) methods can be used to detect gas leakage of the pneumatic pipe system. We propose the DL model for the detection of air leaks from pneumatic pipe system fittings using an accelerometer sensor system. We trained four models with four machine learning (ML) techniques with the data generated from our experimental pneumatic pipe. We augmented the collected data and used it to train all four models again and were able to mimic the natural behavior of the actual line and thereby augment the collected data, which was used in an enhanced training and testing process to create a better model. One of the trained models in which the augmented data was applied yielded the highly accurate result of 99.2%. Our main contribution to the field is our method of evaluating the accuracy of the model and the simple algorithm that one may use as a basis for building applications based on the model, together with the model's evaluation results. Our findings and contribution provide well-tested information to engineers and companies to avoid breakdowns in pneumatic pipe systems caused by air leaks. We claim that these contributions are new, and to the best of our knowledge have not previously been reported in the literature, thus are relevant and important contributions to the field.

Keywords: Compressed air leak, Pneumatic line leakage detection, Deep learning, Pneumatic pipe system, Decision Tree, Random Forest, Support Vector Machine, 1D- Convolutional Neural Networks

1. Introduction

In the manufacturing sectors of industry, many production components rely on compressed air, as discussed by Guenther et al [1] who also states that, while compressed air is helpful, providing it by way of various pneumatic pipe systems has a significant cost. Such systems are often subject to air leakages that reduce the efficiency of the system, causing breakdowns in manufacturing processes. The evidence is that this defect in the systems causes up to half of the losses of production and output in these manufacturing sectors [1]. Of concern too is that these leakages are often found in steel pipes in a production line. It stated that there are thousands of millions of pneumatic pipes in nuclear energy generator stations and chemical plants, and breakdowns in the compressed air systems in these particular industries especially, pose a significant risk [2]. Pneumatic pipes are also essential components for the production lines in many companies/factories.

Interviews with an engineer who we consulted in one company on these issues in pneumatic pipes provided us with good general information on the problems of points/locations in the company's production line. The technical terminology used in this paper includes:

- Point: a point comprises two pipes connected by a two-way fitting.
- Fitting: a device used to join pipes together in a two-way configuration.
- Receiver: a component of a Point connected at one end to a pipe and the other end to an air compressor. The receiver takes the compressed air generated by the compressor and passes it into the line.
- Compressor: equipment that provides compressed air to a pneumatic pipeline.
- Line: the entirety of the pneumatic pipeline that starts at the compressor and along which there a multiple points.
- Pipe: the physical pipe carrying compressed air from the compressor to the various points along the line.

The engineer who we consulted informed us that the significant problem is that the air can leak from the fittings that are located at various points along the line. The number of fittings can be large, depending on the length of the line. There are a variety of conditions that may cause these leaks, including the degradation over time of the fittings of the line. These kinds of leaks are also found in the fittings of plastic water pipes which are also subject to degradation for a variety of reasons [3]. These leaks are critical as they can cause the breakdown of the line. Clearly, in high-risk systems such as are located in nuclear facilities and the production lines of toxic, poisonous, and otherwise hazardous chemicals, the controlling engineer has to search for the leaks by manually inspecting all and each fitting in the entire Line once a breakdown of the line occurs. Not only does this search take considerable time, meaning an extended breakdown or shutdown of the entire system, but an extended breakdown can cause costly maintenance problems as well as production losses. These are unavoidable as the detection process is triggered after the leak has occurred [4]. A way to avoid these breakdowns is by anticipating the leak by early detection of the occurrence of the leak, and by swift and precise identification of the leaking point. A system to achieve these situations is urgently required.

Given this urgency, we propose a deep learning (DL) model for the detection of air leaks from fittings in a pneumatic pipe system line using an accelerometer sensor system. DL is a neural networks machine learning (ML) technique that can be used to create a DL model. The full details of this technique are in section 2.2. Briefly, for our experiments, reported in this paper, we set up a testing pneumatic pipe system line that consisted of two main components: pipes and fittings. This system was set up in our laboratory and data was collected from this test system. The actual configuration of this small system and its functions are described in the Materials and Methods section below. The constant and usual vibrations from the fittings are sensed and recorded and any variation in these vibrations is detected, recorded, and used to identify if any Fitting is in near-failure, or fail situations, thereby achieving early detection of the leaks from fittings, or forestalling the leaks before they occur. Future extensions to the DL model will allow detection of the location of the failing point, and guiding maintenance personnel to the point without the need for laborious manual inspection.

There are three contributions arising from our study discussed in this paper. The first is that we have found that all the models previously produced unacceptable levels of accuracy. We found that the test data that we recorded from our test system did not equate to the normal, natural, or actual operational data that was collected from the factory line. We identified this situation by comparing our test data with this factory production line data and by augmenting our test data with the factory production line data; we achieved a significantly higher level of accuracy from our DL model.

The second contribution claimed is that our proposed DL model achieved 99.2% accuracy. Part of this new DL model is a new algorithm that we developed. This is a simple but effective algorithm that significantly enhances the processing of the DL model.

Based on this model, the third contribution is that the DL model logic and the new algorithm that we developed can be used to build future applications to detect these actual and potential air leaks using an accelerometer sensor system. Our future research will be to build an accelerometer sensor system using this new DL model and a simple algorithm.

We are considering the two situations of a ‘near-failure’ situation and a ‘fail’ situation. The production line engineer needs to focus on both of these types of air leak situations. Using our methods, we could build an application that warns the engineer of the two situations: ‘near-failure’ or ‘fail’, providing the engineer with information to avoid the line breakdown, thereby preventing costly maintenance problems and production disruptions. Our research results are not limited to this one test company but can be of benefit to companies with the same air leak issues.

2. Materials and methods

2.1 Related works

Previous reports on the detection of leaks from water and air pipes were identified. For example, [1] introduced the automated detection of compressed air leaks using a scanning ultrasonic sensor system that enables steel pipes to be scanned and air leaks detected. Gas leakage fault detection of pneumatic pipe systems, using a DL is discussed in [2] where a sound meter instrument was used to analyze acoustic signals. The process of estimating the spectrum of leak noise of fittings in buried plastic water distribution pipes was discussed in [3] which used hydrophone sensors for acoustic measurements or accelerometer sensors remote from the leak for vibration measurements. The summary of this research, together with ours, is in Table 1. The table shows the aspects of each research project based on (1) the primary objective of an application, (2) the adoption of ML techniques, (3) the sensor/instrument, and (4) the subject (s) to be detected.

Table 1 The summary of research for detection of air leaks from fittings in a pneumatic pipe system.

Primary objective of an application	Adoption of ML techniques	Sensor/ instrument	The subject(s) to be detected	Reference
To detect air leaks	No	Ultrasonic	Steel pipe	[1]
To detect gas leaks	Yes	Sound meter	Pneumatic pipes	[2]
To detect noise leaks	No	Hydrophones or accelerometer	Fittings in buried plastic water pipes	[3]
To detect air leaks	Yes	Accelerometer	Fittings in a pneumatic pipe system	This study

From Table 1 we can see that the previous work cited is related to the detection of leaks. Only one of the researchers adopted an ML technique. The work of [3] is related to the detection of the fittings and using the accelerometer to analyze buried plastic water pipes, not pneumatic air pipes. Elsewhere, [5] used an MPU6050 accelerometer sensor to collect vibration data which was then used for vibration detection of water pipeline leaks. The MPU6050 sensor has also been validated with computed vibration [5]. ML is a crucial method for enabling artificial intelligence (AI). ML is a model that focuses on identifying the un-discovered structure, dependence, and function between input and output data [6]. ML can handle high-dimensional and multivariate data making it a potential technique for developing intelligent predictive algorithms [7]. For example, Yang Q, et al [8] increases the accuracy of air leak detection through ML.

2.2 ML techniques

The basic definition of ML is taken from [9]: ML is a subdomain of AI, and DL is a subdomain of ML. A neural network (NN) forms the backbone of DL algorithms. The number or depth of the node layers of an NN is significant in that a DL algorithm needs to have a depth of more than three. According to Rashid A, et al [6] an ML is a model that focuses on identifying the un-discovered structure within a set of data and the dependencies and functionality between input and output data. An ML can handle high-dimensional and multivariate data, making it a potential method for developing intelligent predictive algorithms [7]. An ML, applied in [8], increased the accuracy of air leak detection. According to Ran et al [10], there are many ML techniques/algorithms that can be classified in two ways. The first is what we can call traditional MLs which include decision tree (DT), random forest (RF), and support vector machine (SVM) techniques, and the second includes DL and ML methods, such as convolutional neural networks (CNNs). We focused on DT, RF, SVM, and CNNs (we called it CNN from here) as the primitive techniques for our experiment in which we applied all four techniques as the supervised ML techniques for the classification tasks. DT is a divide-and-conquer technique for regression and classification of data, as defined in [11] who applied DT to learn features from identifiable patterns in large databases. These features and patterns are crucial for differentiation and predictive modeling and can be used to train and generate a DT model. According to Ran et al [10], a group of DT models can be used to train and then generate an RF model.

An SVM is an ML method that is effective when i) the underlying process of the real-life system is unidentified, or ii) the mathematical relation requires intensive computation and is therefore costly to be acquired because of the increased impact by having several mutually dependent influences [10].

However, we used DT, RF, and SVM for primitive analysis of our collected data, along with CNN. We discuss below, in Section 3.1, example results from this analysis to gain insights into the collected data, as Feature importance's of the RF training process. Before the DT, RF, and SVM training processes, the collected data is extracted and the extraction results are called 'input features', and these input features are for the DT, RF, and SVM training processes.

It states that "DL allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction" [12]. DL is an example technique of deep learning applying a CNN [10]. A CNN is a recently defined technique of growing importance for addressing the issue of pattern recognition [13]. The architecture of the CNN that was applied in this research was defined in [14].

The CNN model training architecture consists of three main types of layers: a convolutional layer, a global average pooling layer, and a SoftMax classification layer [14]. The first two types of layers are for feature extraction of the collected data. The third is for classification or we can call this layer as the output layer. This layer can produce a SoftMax fully connected classifier with C neurons equal to the number of classes in the dataset. These three layers constitute the main processes of our experiment.

We will discuss the details of the entire layer processing in the experiment in Section 2.5.2. The experiment generated vibration data from the air leaks from the fittings in the test pneumatic system. The researchers from [15] argued that vibration data, such as ours, can be considered as one-dimensional/1D data, not two-dimensional/2D data, which is more appropriate for image processing, and one-dimensional-CNN (1D-CNN) is suitable for 1D data. Thus, we also applied 1D-CNN to extract abstract features from the collected vibration data. We classified that vibration data into six classes, which are detailed in Section 2.5.2. 1D-CNN was also used by Qin et al [16] for fault diagnosis of the vibration signals from roller bearings. 1D-CNN can generate a model with a level of classification accuracy as high as 95%. This accuracy can be evaluated by computing the confusion matrix which is developed to examine the shortfall in accuracy from the 100% maximum level, for example, the other 5% accuracy. A model classifies the data according to a defined classifier. A confusion matrix is a two-dimensional tabular summary of the number of correct and incorrect predictions made by a classifier. It is used to measure the performance of a classification model. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score. Also, Sammut et al [17] discusses how a confusion matrix is used to review the performance of a classifier concerning some test data when this classifier is performing the classification. In a confusion matrix, the first dimension represents the actual class of an object, while the second dimension represents the predicted class [18]. Section 3.3 will explain complete details of how we used a confusion matrix to produce our main experimental results.

2.3 Air pipe system in production lines and the problem

The company engineer who we consulted gave general information about the pipes and fittings of a point or location in the production line at his company. Figure 1 shows the pipe system before and after the experimental design step. Figure 1A shows the pipe system before the experimental design step, and Figure 1B shows the experimental pipe system. Figure 1B will be thoroughly discussed in Section 2.5.

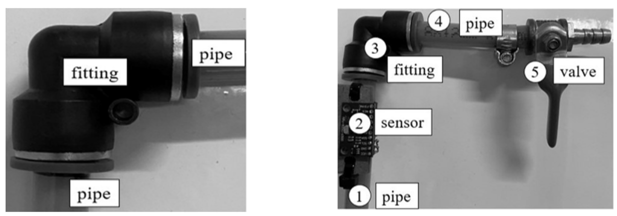

Figure 1 (A) The pipe system before the addition of the experimental components, and (B) The enhanced experimental pipe system with added components.

Figure 1A shows the point consisting of two air pipes connected to a two-way fitting, or Connector. One pipe connects to an Air Receiver with the other end connected to the Compressor (not shown in the figure). The other pipe leads into the production line. The Compressor, which runs continuously, generates compressed air which is stored in the Receiver and then forced through the Line to the pneumatic devices on the production line. Figure 1B shows the enhanced line used for testing. The two added components that are shown in Figure 1B include a valve (labeled as 5) and a sensor (labeled as 2).

The problem to be addressed is that the air can leak from the fitting because of the degradation of the fitting, causing a breakdown of the Line. The engineer must search for the leak or leaks along the entire line by manually inspecting and touching each fitting in the line, which may, in some cases, be hundreds in a very long line. The search can take a long time with the consequential loss of output from that production line. Research has shown that air leaks cause up to half of production line losses in the manufacturing sector [1].

2.4 Vibration data and machine degradation detection

Machine degradation can be detected by the change in vibrations of the machine [4]. An accelerometer sensor can measure the acceleration values which are then represented as values on three directional axes, referred to as the X-axis, Y-axis, and Z-axis. The acceleration values of these three axes are the vibration data. Air leaks from the fittings create vibrations different from the normal background vibrations of the working Line. We used the accelerometer sensor to continuously measure the vibration of the pipe system (Figure 1A) and record the vibration data of the tested pipe system. The vibrations from an air leak showed as aberrations in the normal pattern of vibrations. The design and experiment will be detailed in Figure 1B in Section 2.5.

The use of 1D-CNN is discussed in Qin et al [16] who use a 1D-CNN for fault diagnosis of vibration data from roller bearings. Considering that information, we focused on also using 1D-CNN in our experiment to classify the vibration data collected from our test system, and the production data from the factory whose engineer was advising us.

2.5. Design and experiment

2.5.1 Experimental design

The experimental design was based on specific parts of the actual production line of the company. The design of our small test system was checked by the engineer of that company. We used original components, two fittings and a Pipe, provided by the company, in the design of the testing system. We purchased the other components from appropriate retailers, ensuring that their specifications were approved by the engineer. To test situations, a proper working fitting and a potentially faulty fitting, one brand-new and one used fitting was used.

Figure 1B illustrates the complete pipe system that we used in our tests. The valve (5) controls the opening and closing of the airflow which mimics the usual operation of the production system. The sensor (2) is an MPU6050 accelerometer sensor that measured the vibrations generated from the experimental pipe system, which was recorded and stored in a memory card for further analysis. The MPU6050 has been validated as a well-tested device for the detection of computed vibrations [5]. The MPU6050 recorded the vibration data with acceleration values on the three axes (X-axis, Y-axis, and Z-axis), in the usual manner of accelerometers.

For the experimental component connections, Figure 1B also illustrates the components' connections in the experimental pipe system. The Compressor is always turned on, and the air is always flowing from the Receiver to the left pipe (1) then through the sensor and then fitting into the right pipe (4) and on into the production line (which was not part of our test setup). The valve (5) controls the airflow into the line.

There were three connection configurations tested for the fitting and the left pipe: (i) When the fitting (3) is tightly connected to the left pipe (1). This generates silent noises, probably very low vibrations, indicating no air leaks from the Fitting, (ii) When the fitting is mid-tightly connected to the left pipe (1) due to some actual or potential malfunction, resulting in a minor air leak which generates low noise and probably some extra vibration beyond the normal, and (iii) When the fitting is loosely connected to the left pipe (1), which generated a high level of noise, and large vibrations caused by substantial air leaks from the fitting. We used the brand-new fitting for (i) and used fittings for (ii) and (iii). We created six classes of data for the data that was being collected, based on the knowledge of the engineer. The three configurations of the connections stated above as (i), (ii) and (iii) were classified as in Table 2.

Table 2 Three connection configurations tested for the fitting.

Configuration	Name	Fitting/Situation	Valve
(i) Fitting is tightly connected to the left pipe (number 1 in	Class 0	Tightly connected	Closed
Figure 1B)	Class 1	Tightly connected	Opened
(ii) When the fitting is mid-tightly connected to the left pipe	Class 2	Mid-Tightly connected	Closed
due to some actual or potential malfunction	Class 3	Mid-Tightly connected	Opened
(iii) When the fitting is loosely connected to the left pipe,	Class 4	Loosely connected	Closed
which generated a high level of noise, and large vibrations.	Class 5	Loosely connected	Opened

The hardware and software configurations for the experiment included an AMD Ryzen 5 3600 CPU, RAM Speed 3200 MHz, with a GPU NVIDIA GeForce GTX 1070, and Python 3.8.5, time-series features extraction library or TSFEL [19], scikit-learn [20], and Keras [21].

2.5.2 Experiment

The seven steps in the experiment are illustrated in Figure 2. The vibration data was collected and classified into the six classes in 7 steps:

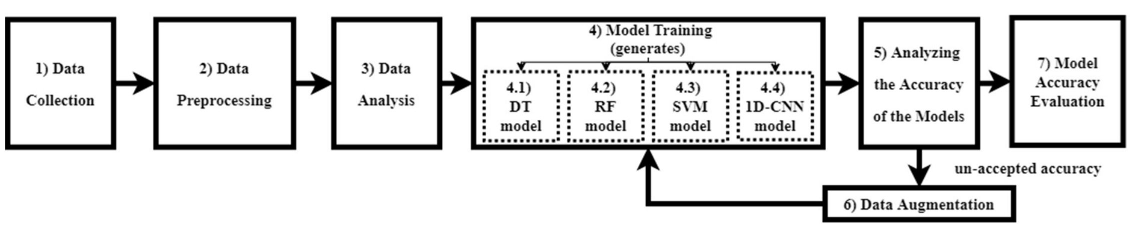

Figure 2 The seven steps of the experimental process.

2.5.2.1 Data collection

The vibration data was collected from the accelerometer sensor, labeled as (2) in Figure 1B, for the brand-new and used fittings. To collect the data for each class, we followed this process:

1) With the valve closed on a tightly connected fitting, we collected the data for Class 0 for 1 min and the data was saved into a .csv. During this 1 min, the fitting was sampled every 50 ms, giving 3,000 data points in the .csv, and each data point is called a timestep. This was repeated 19 times, giving 20 separates .csv files giving us 60,000 data points for Class 0.

2) Similarly, data was collected for Class 2 (mid-tightly connected fitting with the valve closed) giving another 20 .csv files with Class 2 data, with 60,000 data points.

3) The data for Class 4 (loosely connected fitting with the valve closed) was also collected, giving another 20.csv files, with 60,000 data points.

Then, with the valve opened, the procedure was repeated for Class 1 (tightly connected fitting, the valve opened), Class 3 (mid-tightly connected fitting with the valve opened), and Class 5 (loosely connected fitting with the valve opened. So, for each of these classes we had 20 .csv files, each with 60,000 data points.

This gave 120 .csv files of data in total with the .csv data representing 360,000 timesteps for all classes.

2.5.2.2 Data preprocessing

The data was cleaned, prepared, and manipulated appropriately to fit into the six classes of acceleration data. This included manipulating the data into data sub-sequences. The 60,000 timesteps for one class were divided into 500 timestep sub-sequences, giving 120 sub-sequences for each class. We refer to these subsequences as signals. Thus, we had a total of 720 signals (i.e., 120 signals x 6 classes). Note that the overall number of 60,000 timesteps, 50 ms sampling rate, and 500 timesteps for each signal was agreed to by the engineer so have practical and professional veracity. The 720 signals were randomly divided into 80% training set data and 20% testing set data.

2.5.2.3 Data analysis

These signals, now the primary data for our tests, were analyzed with non-ML processes such as data plotting. This helped us to understand the behavior of the data that had been collected and preprocessed. These plots of the primary data enabled us to differentiate all the classes from each other. At this point, we were able to visually inspect the data plots and ascertain that the plots differed significantly between classes, as can be seen in Figures 3. We found that we could not differentiate Class 0 from Class 2.

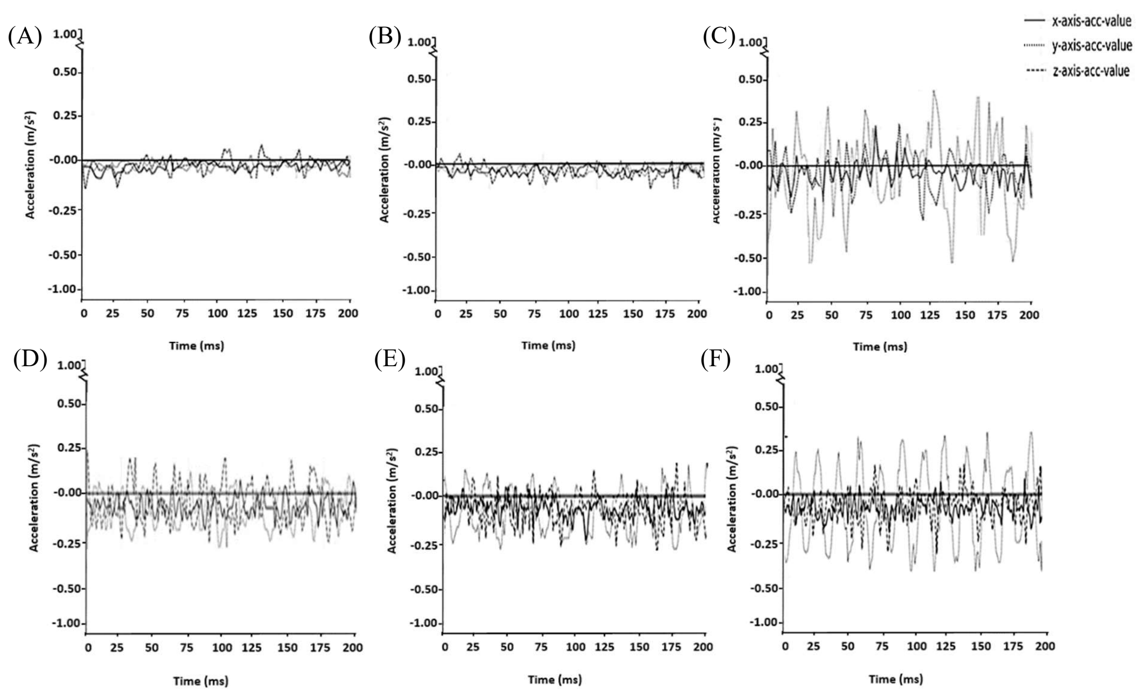

Figure 3 Data Plotting: (A) class 0 (fitting tightly connected-valved closed), (B) class 2 (fitting mid-tightly connected-valved closed), and (C) class 4 (fitting loosely-tightly connected-valved closed), (D) class 0 (fitting tightly connected- valve opened), (E) class 3 (fitting mid-tightly connected-valve opened), and (F) class 5 (fitting loosely-tightly connected-valve opened).

Figure 3 are examples of the collected and preprocessed data in each of the six classes. Each class has three plot-line graphs in each figure. The X-axis of each graph represents the timesteps from 0 ms to the 200th ms in 1 ms intervals. The Y-axis represents the vibration values generated from the experimental pipe system (Figure 1 (B)) measured by the sensor. The range of the vibration values is -1 to +1 in m/s^2. This range is the result of the normalization of the actual vibration values. The three lines of each graph are the three acceleration values of the X-axis, Y-axis, and Z-axis, labeled as the X-axis-acc-value, the Y-axis-acc-value, and the Z-axis-acc-value, respectively.

From Figure 3, we can see that all the three lines of the Class 0 (tightly connected fitting with the valve closed) graph are very different from the three lines of Class 4 (loosely connected fitting with the valve closed) graph. However, we could not differentiate the Class 0 data from the Class 2 data as the shapes of graphs of both classes are almost the same. This means that, because we cannot classify the tight-fitting normal situation from the mid-tightly fitting near-failure situation, then we only need to build the application to inform the engineer when (i) the fitting is tightly connected, which is the normal situation (Class 0), and (ii) the fitting is loosely connected (Class 4), which represents both the failed or about to fail situations. Thus, we can classify how critical the air leaks are from the fitting in only two situations:

1) (Class 0) fitting tightly connected or the normal situation.
2) (Class 4) fitting loosely connected or the fail situation.

To enable the application to be more useful, the application needs to warn the engineer of the fail situation before it occurs. The situation extant before the fail situation occurs is when the fitting is mid-tightly connected, which is the near failure situation (Class 2). Thus, we needed only to classify Class 0 from Class 2, and for this we needed ML. Therefore, we examined and chose the appropriate ML techniques for classifying the six classes of our preprocessed data again. We chose DT, RF, SVM, and 1D-CNN.

2.5.2.4 Model training

We performed model training with the four chosen ML techniques. The preprocessed data were processed using the statistical feature extraction by TSFEL. The extracted features were then used as input data into the DT, RF, and SVM models. This preprocessed data was also directly used as the input for 1D-CNN.

For cross-validation purposes, the same 10% of the training data was used as the validation data applied in each technique.

The CNN model training architecture consists of three main types of layers: a convolutional layer, a global average pooling layer, and a softmax classification layer [14]. The first two types of layers are for feature extraction of the collected data. The third is for classification or we can call this layer the output layer. This layer can produce a softmax fully connected classifier with C neurons equal to the number of classes in the dataset. These three layers constitute the main processes of our experiment.

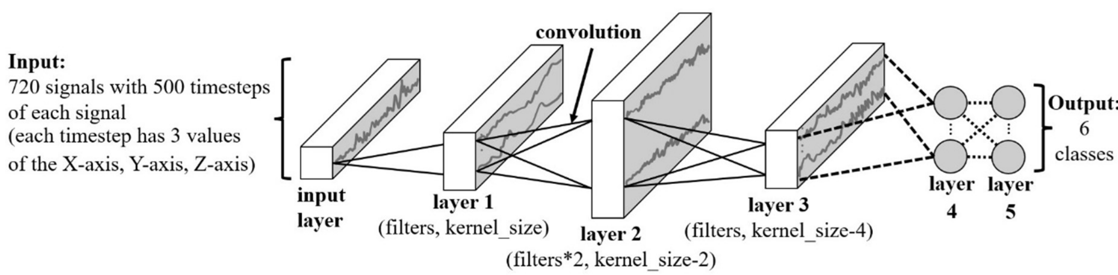

Figure 4 Architecture of the 1D-CNN adopted for the training process, adapted from [14].

Figure 4 shows the architecture of the 1D-CNN adopted for the training process in our tests, adapted from [14]. The architecture has both Input and Output. The Input to the input layer is the 720 signals with 500 timesteps of each signal (each timestep has the three acceleration values of the X-axis, Y-axis, and Z-axis). The architecture produces the Output as a probability distribution over the six possible classes (Class 0, Class 1, Class 5) in the training dataset. We can call this distribution as a softmax fully connected classifier or a trained 1D-CNN model.

From Figure 4 we can also see that the architecture consisted of (i) three convolutional layers, (layers 1-3), (ii) a global average pooling layer (layer 4), and (iii) a softmax classification layer (layer 5). The settings of all the convolutional layers and their hyperparameters (in Table 3) were also adapted from [14]. Thus, we set an initial, certain value for 'filters and another value for kernel size in layer 1. Then, in layer 2, we increased the 'filters' to be twice the value from layer 1 (filters*2) and halved the kernel size from the value in layer 1 (kernel_size 2). Finally, in layer 3, the filters were set back to the same as in layer 1, and the kernel size was decreased by four from the value in layer 1 (kernel_size 4).

Table 3 Hyperparameters tuning of each technique.

ML techniques	Hyperparameter	Range of value
DT	criterion	('gini', 'entropy')
	splitter	('best', 'random')
	max_depth	(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None)
	min_samples_split	(2, 5, 10)
	min_samples_leaf	(1, 2, 4)
RF	n_estimators	(200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000)
	max_features	('auto', 'sqrt')
	max_depth	(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None)
	min_samples_split	(2, 5, 10)
	min_samples_leaf	(1, 2, 4)
	bootstrap	(True, False)
SVM	C	(2.0, 2.8, 3.6, 4.4, 5.2, 6.0, 6.800, 7.600, 8.4, 9.2, 10.0)
	kernel	('linear', 'poly', 'rbf', 'sigmoid')
	gamma	(0.1, 0.19, 0.28, 0.37, 0.4599, 0.5499, 0.64, 0.73, 0.82, 0.9099,1.0)
	probability	(True, False)
	degree	(3, 4, 5, 6, 7, 8, 9, 10)
	coef0	(2.0, 2.8, 3.6, 4.4, 5.2, 6.0, 6.8, 7.6, 8.4, 9.2, 10.0)
1D-CNN	filters	(256, 128, 64, 32, 16, 8)
	kernel_size	(15, 13, 11, 9, 7)

Table 3 shows the tuning of the hyperparameters of each ML technique. For DT, RF, and SVM. The details of all the hyperparameters are adapted from [22], and the hyperparameters for 1D-CNN are adapted from [14]. This fourth step generates the four trained models of DT, RF, SVM, and 1D-CNN, as shown by the four dotted boxes in Figure 2. The results of this step are stated in Section 3 and discussed in Section 4.

2.5.2.5 Analyzing the accuracy of the models

In Step 5, the predictions from the four trained models were taken as inputs and the level of accuracy of each model was generated. We found that the accuracy of all the four trained models; DT (81.61%), RF (88.23%), SVM (87.50%), and 1D-CNN (86.30%), were all considered to be unacceptable. These unacceptable accuracies were caused by the unnatural behavior of our collected data which we could identify when the test data was compared to the actual data from the production line of the factory. The full details of the unnatural and natural behavior of data will be discussed in the sixth step.

To gain better accuracy, we needed to augment the collected from Step 1 and the preprocessed data from Step 2 to be closer to the actual data, in the sixth step.

2.5.2.6 Data augmentation

The sixth step performs data augmenting of the preprocessed data (the 720 signals) then we get the new augmented data as 6,000 signals. Recall that Class 0 and Class 1 include data from the tightly connected fitting, Class 2 and Class 3 include data from the mid-tightly connected fitting, and Class 4 and Class 5 include data from the loosely connected fitting. The preprocessed data was divided into 80% as training set data and 20% testing set data.

From Step 4 (Model Training) and illustrated in Figure 5, this figure composes of both the Input and Output and the Input to the input layer is the training set data. This data comprised the 720 signals, each of which was composed of the 500 timesteps.

To describe the augmenting process, each signal from 720 signals represents the x, y, and z data of the five 100 timesteps of the class represented as a 5-digit string. If we assume that we have a signal of '00000' as one signal data of Class 0, we will call this signal 'signal s0' which has 500 timesteps or, more to the point, 5 'sets' of 100 timesteps. Thus, the first 0 number in signal s0 represents the data x, y, and z of Class 0 of the first set of 100 timesteps. This data of this set of 100 timesteps can be seen as [x0, y0, z0], [x1, y1, z1], [x99, y99, z99]'. The second number, again '0' in signal s0, represents the x, y, and z data for the second set of 100 timesteps of Class 0. This data of these second set of 100 timesteps can be seen also as [x100, y100, z100], [x101, y101, z101], [x199, y199, z199]'. Thus, signal s0 or '00000' has 500 timesteps and each timestep has its different data of x, y, and z. Thus, all the data of signal s0 or '00000' represents the 500 timesteps and can be seen as [x0, y0, z0], [x1, y1, z1], [x499, y499, z499]. And this data is only one signal of all the 720 signals of the training data and this signal is only of Class 0.

As a reminder to the reader, we publish here the definitions of the six classes from previously in Table 2 in this paper.

So, the collected data now being used as the training data from all six classes above is in the form of six signals represented as, for example, 00000, 11111, 22222, 33333, 44444, and 55555.

The description of signal s0 above can apply to each signal of all these six signals. And these six signals represent the unnatural behavior of our collected data from the test line. There are three situations as there are three situations as tightly connected, Mid-Tightly connected, and loosely connected. If we are concerned that two signals of the six signals are in the same situation, we can have three pairs of all the six signals. Every 100 timesteps of a signal can be only shuffled/augmented with another signal in the same situation. Then after the augmenting process of collecting data of the six signals, this augmented data can be considered to have a natural behavior. This is similar to the behavior of the data of the actual line and is to be ready to be the input for Step 4 (Model Training) again.

Thus, for example, the two signals (00000, 11111) of Class 0 and Class 1, which both are data from the same fitting tightly connected situation, can be shuffled/augmented by steps (a) and (b). (a) Both signals can be shuffled. Thus, the results are that the two signals '00000, 11111' can be shuffled as '00011, 10101'. (b) The shuffled result of signal '00011' is newly considered to be a new augmented signal of Class 0. This is because the sum of the elements '0' in '00011' is 3, which is greater than the sum of elements '1' in '00011', which is 2. The shuffled result of '10101' is considered to be a new augmented signal of Class 1. This is because the sum of the elements '1' in '10101'is 3, which is greater than the sum of elements '0' in '10101', which is 2. In step (c), we applied steps (a) and (b) to our preprocessed data for Classes 0 and 1 until, we had 500 augmented signals for each class. Then, in further step (d), we applied steps (a) to (c) for our training set data of Class 2 and 3, and Class 4 and 5. Thus, all our original collected data as '00000, 11111, 22222, 33333, 44444, and 55555' can be augmented to be, for example, '00011, 11100, 22233, 33322, 44554, and 45554'.

We thereby had 3000 new and augmented signals (500 signals per class of six classes) as the new training set data. For the testing set data, we applied (a) to (d) again to this data, giving another new 3000 augmented signals as the new testing set data. Both new sets were then input into Step 4 (model training) again. The augmented data (the augmented 6000 signals) should be close to the natural behavior of the actual data from the line. However, in Step 5 (Analyzing the accuracy of the models) the newly generated augmented data could look like the original data set because of the phenomenon of randomization in this augmenting process.

In Step 4 (Model Training) we had found that the accuracies of the four trained models of DT (81.61%), RF (88.23%), SVM (87.50%), and 1D-CNN (86.30%), as indicated above, were unacceptable. So, we repeated Step 4 (Model Training) to train all the four models again with the new augmented. We repeated Step 5 (analyzing the accuracy of the models) on the augmented data again and achieved 99.2% accuracy only from the new trained 1D-CNN model. Thus, we could skip Step 6 (data augmentation) as the new model was within acceptable accuracy. Finally, we could go to Step 7.

2.5.2.7 Model accuracy evaluation

This step generated the model's confusion matrix which enabled us to know where the confused-predicted values were. The 99.2% accuracy was evaluated by computing a confusion matrix to identify the reason for the 0.8% shortfall from 100% accuracy.

These values can be used as a guideline for other applications, as will be discussed in Section 4. Such an application can be used to detect air leaks from fittings in a pneumatic pipe system using an accelerometer sensor system, and then warns the engineer about the leaks.

3. Results

The full details of the results of Steps 4, 5, and 7 are discussed in this section. As introduced in Section 2.5.1 above, the experimental data were classified into six classes.

3.1 Feature Importance's of the RF training process

We used DT, RF, and SVM for primitive analysis of our collected data, along with CNN. This section discusses the example results from this analysis to gain insights into the collected data, as fittings of the RF training process. Section 2.2 already discussed on the input features; a feature importance method is a method that gives a grade to the input features according to how effective the features are at predicting a target variable [23]. Applying this method produces a set of features each with its mean decrease in impurity value [24]. The higher the mean decrease in impurity value is the higher importance of that particular feature. Figure 5A shows some significant parts of a set of feature importance values from applying the feature importance's method in the RF training process. These feature importance values express the insights gained by calculating the feature importance values. The first four dominant features Importance's of the RF model are the minimum (min), median, variance (var), and standard deviation (STD) features.

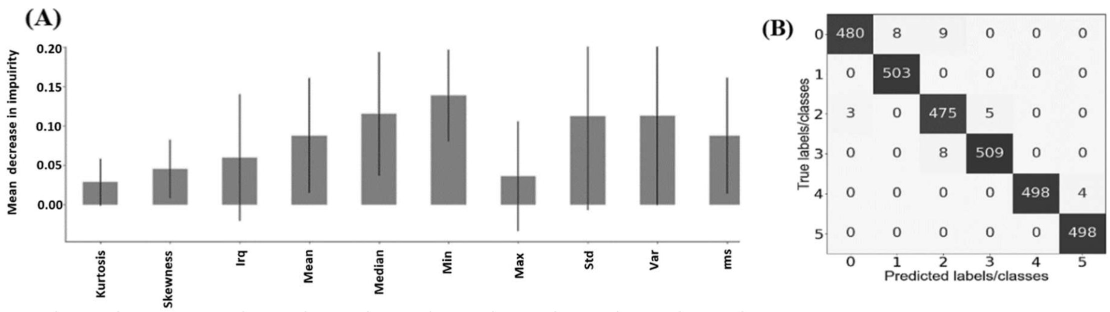

Figure 5 (A) Feature importance's of the RF training process, and (B) confusion matrix: predicted test classification with our 1D-CNN model.

3.2 Model accuracies

From Section 2.5, and before the data augmenting, we found that all the four accuracies of four trained models of DT, RF, SVM, and 1D-CNN were unacceptable. We found that the unacceptable accuracies were caused by the unnatural behaviors of our collected data when compared to the actual data from the production line. We then performed the data augmenting step from which we elicited the augmented data to be trained that were then used to obtain the four new models again. Accuracies of the old and new models (shown here as old %, new %, difference %) were DT (81.61%, 89.23%, +7.72%), RF (88.23%, 89.56%, +7.79%), SVM (87.50%, 93.13%, +5.63%), and 1D-CNN (86.30%, 99.2%, +12.9%).

The new 1D-CNN model had the highest accuracy and the highest difference (99.2%, +12.9%), so this model is appropriate to be implemented for the detection of air leaks from fittings in a pneumatic pipe system using an accelerometer sensor system for the engineer. The guidelines to implement this application will be discussed in Section 4.

3.3 Confusion matrix of our 1D-CNN models

In this section, the computation of the confusion matrix [17] to evaluate the accuracy of classification of our new 1D-CNN model is discussed.

In the discussion regarding the confusion matrix in Section 2.2, Section 2.5.2, and Section 3.2, the 99.2% accuracy of our new 1D-CNN model was evaluated by computing the confusion matrix which was used to examine the reasons for the 0.8% accuracy discrepancy. Our model used a classifier to perform the classification, and the confusion matrix was able to be used to review the performance of that classifier when the test data was being processed and classified.

This confusion matrix has two dimensions: the actual or true labels/classes of an object and the predicted labels/classes. Figure 5B shows the confusion matrix of the classifier of our new 1D-CNN model. From Figure 5B, the X-axis shows the predicted labels/classes, including Class 0 to Class 5. The Y-axis is the true or actual labels/classes, including Class 5 to Class 0. The summation of all the non-zero cells will be 3,000. This is the number of all signals of the data from the experiment. To analyze this matrix, we can firstly look at the X-axis and then look at the Y-axis of Class 0. The number at the intersection between both looking steps is 480. This number means that our 1D-CNN model has a correct prediction of Class 0 480 out of 483 times. Then, we can look at the X-axis at Class 0 again and at the Y-axis at Class 2. The number at the intersection between both looking steps is 3. This number means that the model has the false prediction of Class 0 (to be Class 2) three times.

From the matrix, we can look at the X-axis and Y-axis at Class 4. The number at the intersection between both looking steps is 498 and is the highest/best true prediction. Thus, the best one is the prediction of Class 4 with 498 times the true prediction and 0 times of false prediction. The lowest/worst true prediction is the prediction of Class 2, with 475 times of the true prediction but 17 (8+9) times of the false prediction. Overall, the true predictions of Class 0 through Class 5 are 480, 503, 475, 509, 489, and 498 times and the false prediction of Class 0 through Class 5 are 3, 8, 17 (8+9), 5, 0, and 4 times.

4. Discussion

4.1 Our 1D-CNN model as the guideline for addressing the concerns of classification of class 0 and class 2

Our 1D-CNN model can be the appropriate guideline for addressing the concerns of the classification of Class 0 (fitting is tightly connected, valve opened) and Class 2 (fitting is mid-tightly connected, valve opened). We have discussed above that we could not differentiate the situations of these two classes, meaning that we will have only two situations, normal or fail. Then, the application can warn the engineer of only the normal and fail situations of the fittings. This is not a useful application; we need to warn the engineer of 'normal' (Class 0), 'near-failure' (Class 2), and 'fail' (Class 4) situations; thus, we need to differentiate Class 0 from Class 2. This section will discuss how our 1D-CNN models can be the appropriate guidelines for classifying Class 0 from Class 2. In the experiment, this model can classify Class 0 from Class 2 with 98.78% accuracy and with this accuracy; the model can be considered the appropriate model. Thus, one can implement the detection of air leaks from fittings in a pipe system using an accelerometer sensor system, and then warns the engineer about the leaks based on this model.

From the confusion matrix in Section 3.3 and Figure 5B, the predictions of Class 0 were false three times. The three false prediction values can be seen as '3' when X-axis at 0 and Y-axis at 2 from Figure 5B. The three false prediction values were Class 2 for all the three values instead of Class 0. These three false prediction values are about 0.62% of all the particular true and false prediction values of Class 0 here. 0.62% is calculated from '3 falses / (480 trues + 3 falses) * 100'. Note that, from Figure 5B, '480 trues' is the number from X-axis at 0 and Y-axis at 0. The predictions for Class 2 were false nine times. The nine false prediction values can be seen as '9' when X-axis at 2 and Y-axis at 0 from Figure 6B. The nine false prediction values were Class 0 for all the nine values instead of Class 2. The nine false prediction values are about 1.83% of all the particular true and false prediction values of Class 2 here. Thus, 1.83% is calculated from '9 falses / (475 trues + 9 falses + 8 falses)* 100'. Note that, from Figure 6 (B), '8 falses' is the number from X-axis at 2 and Y-axis at 3. From Figure 3, we have a problem with the classification of Class 0 and Class 2 when performing the classification without ML, even with DT, RF, SVM, and 1D-CNN ML techniques. However, from Figure 5B in this section here, 1D-CNN can generate the model with the accuracies of the correct prediction values of Class 0 as 99.38% (100% - 0.62%) and 98.17% (100%-1.83%) of Class 2, respectively. Then, the average is 99.38% and 98.17% is 98.78%. Thus, this model can classify Class 0 from Class 2 with 98.78% accuracy. Thus, our 1D-CNN model can be the appropriate guidelines for addressing the concerns of classification of Class 0 and Class 2 and for building the possible application discussed in the next section.

4.2 Class 4 and class 5 as in the same loosely fitting situation and the possible application

4.2.1 Class 4 and class 5 as in the same loosely fitting situation

This section discusses why Class 4 and Class 5 can be considered in the same loosely fitting situation. This same situation can enable the possible application. From Section 4.1 above, the model can address the concerns of classification of Class 0 and Class 2 and of building the possible application. Moreover, from Section 3.2, the model can classify Class 0, Class 1, Class 2, Class 3, Class 4, and Class 5 with 99.2% accuracy. Thus, the model can provide guidelines for detecting air leaks from fittings in a pneumatic pipe system using an accelerometer sensor system, based on the three situations of 'normal' (Class 0 and Class 1) 'near-failure' (Class 2 and Class 3) and 'fail (Class 4 and Class 5). In Section 2.5.1, we labelled the data of the pipe leaks into six classes. Class 4 is 'the fitting loosely connected-valve closed,' and Class 5 is 'the fitting loosely connected-valve opened'. Thus, Class 4 and Class 5 are both loosely fitting situations. Our 1D-CNN model can differentiate this loosely fitting situation (Class 4 and Class 5) from the mid-tightly situation (Class 2 and Class 3), discussed in Section 3.2. From the confusion matrix or Figure 6B, the 1D-CNN model did not make a false prediction of Class 4. However, the model made 4 false predictions for Class 5. The four false prediction values were 'Class 4', instead of Class 5. However, both Class 4 and Class 5 are about the loosely fitting situation. Thus, the model still knows that these four false prediction values of these 4 data/signals are still of the loosely fitting situation, not the mid-tightly or tightly fitting situation.

4.2.2 The possible application

Given the discussion in Section 4.2.1, the application is applicable when the engineer focuses on the two types of leak situations of the fittings: the mid-tightly and loosely fitting situations. The mid-tightly situation is when the fitting is mid-tightly connected (Class 2 and Class 3). The loosely connected situation is when the Fitting is loosely connected (Class 4 and Class 5). Suppose that, normally, the engineer did not focus on the normal (or the Fitting is tightly connected) situation which, by its very state, does not require attention. Thus, from the results in Section 3 and discussions about Class 4 and Class 5 here, we could build an application that can alert the engineer of the production line to the two types of leaks. The application can alert the engineer to the two conditions of the leaks with also two types of alerting messages as follows. The first is ’near-failure’. This is about the mid-tightly fitting situation (Class 2 and Class 3). The second is ’fail’. This is about the loosely fitting situation (Class 4 and Class 5).

The possible application can be, for example, when an accelerometer at the production line produces actual data/signals (with the series of values for each timestep continuously) for the application. Then, the application can feed every 500 timesteps of the data to the proposed model in real-time. The model will classify this 500 timesteps data as one of Class 0 to Class 5. Thus, the classification result of these 500 timesteps data can be only a value from 0, 1, 2, 3, 4, or 5, representing Class 0 to Class 5. If the application can obtain 5000 timesteps data and feed the data to the model, the model will give ten classification result values (as 0, 1, 2, 3, 4, or 5) of this 5000 timesteps data. The example of these ten classification result values is 3330123401. Thus, each number of 3330123401 is a classification result from the model of each 500 timesteps of the 5000 timesteps data.

The first scenario of the application is when the ten classification result values are 3330123401. Thus, there are 0 (Class 0) or 1 (Class 1) four times in 3330123401, 2 (Class 2) or 3 (Class 3) five times in 3330123401, and 4 (Class 4) or 5 (Class 5) for one time in 3330123401. Thus, the summation value of the occurrences of Class 0 and Class 1 is 4, Class 2 and Class 3 is 5, and Class 4 and Class 5 is 1. The proportion numbers of occurrences of Class 0 and Class 1 is 4/10, Class 2 and Class 3 is 5/10, and Class 4 and Class 5 is 1/10. It is noted that an alerting message generated from the application for the engineer can be triggered when, for example, any of the three proportion numbers is greater than or equal to 5/10 or 0.5. Thus, in this case, the message is ’near-failure’ (the mid-tightly fitting situation (Class 2 and Class 3)) because the proportion number of Class 2 and Class 3 is 5/10 (above) and greater than or equal to 0.5.

The second scenario of the application is when the ten classification result values are 4450223454, which indicates that there is data for Class 0 or Class 1 once, three times for Class 2 or Class 3, and six times for Class 4 or Class 5. Thus, the three summation values are 1, 3, and 6, respectively. The proportion numbers of occurrences of Class 0 and Class 1 is 1/10, Class 2 and Class 3 is 3/10, and Class 4 and Class 5 is 6/10. Thus, with the same alerting condition above, in this case, the message is ‘fail’ (the tightly fitting situation (Class 4 and Class 5)) because the proportion of Class 4 and Class 5 is 6/10, which is greater than or equal to 0.5 which is the threshold value.

To sum up, our 1D-CNN model classified the data collected from our experimental pipe system Class 0, Class 1, Class 2, Class 3, Class 4, and Class 5 with 99.2% accuracy. Thus, the 1D-CNN model is effective for detecting air leaks from fittings in a pneumatic pipe system using an accelerometer sensor system. The detection can enable an application to alert the engineer about the three situations of the fittings in the pipe system of the company’s line with the three main types of messages. They are ‘normal’ (or the fitting is tightly connected or Class 1 and Class 2), ‘near-failure’ (or the fitting is mid-tightly connected or Class 2 and Class 3), and ‘fail’ (or the fitting is loosely connected or Class 4 and Class 5). As our analysis showed, the data categorized in Classes 2 and 3 are effectively equal to the data in Classes 4 and 5, allowing the application to view these two situations as the same. This is because a failed fitting will usually, if not always, be heralded by a near-failing fitting situation. These alerting messages can facilitate the engineer to faster search the leaks and reduce the breakdowns of the line. Then, this can avoid low production of the line of this company. This hopefully benefits other companies with the same air leak issues as this company. The compressed air leaks can cause half of the losses in manufacturing sectors [1].

5. Conclusion

We developed a convolutional neural network DL machine learning model for the detection of air leaks from fittings in a pneumatic pipe system using an accelerometer sensor system. The first finding was that the unacceptable accuracies identified in all trained models, before augmenting the collected data, were caused by the unnatural behavior of our collected data as compared to the actual data from the production line. We augmented the collected data by mimicking the actual data. The augmenting process also increased the population of the augmented data which created a better training and testing process when used, to create a better model. The second finding from our testing is that our model has 99.2% accuracy after the augmenting process, compared to 86.30% accuracy with our pre-augmentation data. The third finding is that the evaluation of the accuracy and the discussions of the simple algorithm validly enable the building of applications based on the model and the evaluation results. We can therefore suggest that the application reliably alerts manufacturing line operators to the 'near-failure' or 'fail' situations of a pipe system's fittings. Leak detection methods can overcome some of the damaging issues associated with air leaks in pneumatic air systems, including production system downtime, process imprecision, and noise [25]. Our future work will further develop the application and enable its deployment into actual production lines to cope with the issues. We will also continue to develop the model to include the ability to identify the location of the leak to overcome the significant problem of requiring the production line supervisor to manually inspect the fittings along the line until the faulty fitting is identified, which can create significant delays in finding and fixing the faults.

6. Acknowledgments

This research was supported by a scholarship from the National Science and Technology Development Agency in a scholarship contract to support postgraduate education in Thailand Graduate Institute of Science and Technology Project No. SCA-CO-2563-12152-TH and the research grant R2565E069 from Faculty of Science, Naresuan University. The research was also supported by Thai Gow Gai Group Co., Ltd, the manufacturer that provided the knowledge of a pneumatic pipe system and the consult engineers for this research. We also thank Mr. Phanthawat Phiphatkamtorn for his support, and Mr Roy I. Morien of the Naresuan University Graduate School for his efforts in editing the grammar, syntax and general English expression in this paper.

7. References

[1] Guenther T, Kroll A. Automated detection of compressed air leaks using a scanning ultrasonic sensor system. IEEE Sensors Applications Symposium (SAS); 2016 Apr 20-22; Catanai, Italy. New Jersey: IEEE Xplore; 2016. p. 1-6.

[2] Zhang S, Asakura T, Hayashi S. Gas leakage fault detection of pneumatic pipe system using neural networks. JSME Int J. 2004;47(2):568-572.

[3] Scussel O, Brennan MJ, Almeida FCL, Muggleton JM, Rustighi E, Joseph PF. Estimating the spectrum of leak noise in buried plastic water distribution pipes using acoustic or vibration measurements remote from the leak. Mech Syst Signal Process. 2021;147:107059.

[4] Vafaei N, Ribeiro R, Matos CL. Fuzzy early warning systems for condition based maintenance. Comput Ind Eng. 2019;128:736-746.

[5] Ismail MI, Dziyauddin RA, Salleh ANA, Sukki MF, Aini Bani N, Mohd Izhar MA, et al. A review of vibration detection methods using accelerometer sensors for water pipeline leakage. IEEE Access. 2019;7:51965-51981.

[6] Rashid A, Abduljabbar H, Alhayani B. Coronavirus disease (COVID-19) cases analysis using machine-learning applications. Appl Nanosci. 2023;13(3):2013-2025.

[7] Carvalho TP, Soares FA, Vita R, Francisco RP, Basto JP, Alcala SG. A systematic literature review of machine learning methods applied to predictive maintenance. Comput Ind Eng. 2019;137:106024.

[8] Yang Q, Guo B, Lin M. Differential pressure prediction in air leak detection using RBF neural network. Conference on Artificial Intelligence and Computational Intelligence; 2010 Oct 23-24; Sanya, China. New Jersey; IEEE Xplore; 2010. p. 211-213.

[9] Kavlakoglu E. AI vs. Machine learning vs. deep learning vs. neural networks: what's the difference?, https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks [accessed 27 November 2021].

[10] Ran Y, Zhou X, Lin P, Wen Y, Deng R. A survey of predictive maintenance: systems, purposes and approaches. arXiv EESS 2019; arXiv:1912.07383. doi.org/10.48550/arXiv.1912.07383.

[11] Myles AJ, Feudale RN, Liu Y, Nathaniel A, Woody N, Brown S. An introduction to decision tree modeling. J Chemometrics. 2004;18:275-285.

[12] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436-444.

[13] Valueva MV, Nagornov NN, Lyakhov PA, Valuev GV, Chervyakov NI. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Math Comput Simul. 2020;177:232-243.

[14] Fawaz IH, Forestier G, Weber J, Idoumghar L, Muller PA. Transfer learning for time series classification. IEEE International Conference on Big Data (Big Data); 2018 Dec 10-13; Seattle, United States. New Jersey: IEEE Xplore; 2019. p. 1367-1376.

[15] Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ. 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process. 2021;151:107398.

[16] Qin H, Xu K, Ren L. Rolling bearings fault diagnosis via 1d convolution networks. IEEE International Conference on Signal and Image Processing (ICSIP); 2019 Jul 19-21; Wuxi, China. New Jersey: IEEE Xplore; 2019. p. 617-621.

[17] Sammut C, Webb GI. Encyclopedia of machine learning. 1st ed. New York: Springer Publishing; 2011.

[18] Deng X, Liu Q, Deng Y, Mahadevan S. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inf Sci. 2016;340-341:250-261.

[19] Barandas M, Folgado D, Fernandes L, Santos S, Abreu M, Bota P, et al. TSFEL: Time series feature extraction library. SoftwareX. 2020;11:100456.

[20] Kramer O. Scikit-learn. In: Kramer O, editor. Machine learning for evolution strategies. 1st ed. New York: Springer Charm; 2016, p. 45-53.

[21] Ketkar N. Introduction to Keras. In: Chollet F, editor. Deep learning with Python. 1st ed. New York: Manning Publications; 2017, p. 97-111.

[22] Sklearn. Free software machine learning in Python, https://scikit-learn.org [accessed 27 November 2021].

[23] Brownlee J. How to calculate feature importance with Python, https://machinelearningmastery.com/calculate-feature-importance-with-python/ [accessed 27 November 2021].

[24] Sklearn. Feature importances with a forest of trees, https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html [accessed 27 November 2021].

[25] Thabet M, Sanders D, Becerra V, Tewkesbury G, Haddad M, Barker T. Intelligent energy management of compressed air systems. IEEE International Conference on Intelligent Systems (IS); 2020 Aug 28-30; Varna, Bulgaria. New Jersey: IEEE Xplore; 2020. p. 153-158.