applied sciences

MDPI

Article

An Applicable Predictive Maintenance Framework for the Absence of Run-to-Failure Data

Donghwan Kim [1], Seungchul Lee [2] and Daeyoung Kim [2,*]

[1] School of Industrial and Management Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea; donhkim9714@korea.ac.kr
[2] BISTel Inc., 128, Baumoe-ro, Seocho-gu, Seoul 06754, Korea; sclee@bistel.com
* Correspondence: dykim3@bistel.com; Tel.: +82-2-597-0911

Abstract: As technology advances, the equipment becomes more complicated, and the importance of the Prognostics and Health Management (PHM) to monitor the condition of the equipment has risen. In recent years, various methodologies have emerged. With the development of computing technology, methodologies using machine learning and deep learning are gaining attention, in particular. As these algorithms become more advanced, the performance of detecting anomalies and predicting failures has improved dramatically. However, most of the studies are cases that depend on simulation data or assumed abnormal conditions. In addition, regardless of the existence of run-to-failure data, the methodologies are difficult to apply to the industrial site directly. To solve this problem, we propose a Predictive Maintenance (PdM) framework based on unsupervised learning in this paper, which can be applied directly in the industrial field regardless of run-to-failure data. The proposed framework consists of data acquisition, preprocessing data, constructing a Health Index, and predicting the remaining useful life. We propose a framework that can create and monitor models even when there are no accumulated run-to-failure data. The proposed framework was conducted in two different real-life cases, and the usefulness and applicability of the proposed methodology were verified.

Keywords: Prognostics and Health Management (PHM); Predictive Maintenance (PdM) framework; health index; remaining useful life (RUL); autoencoder

Citation: Kim, D.; Lee, S.; Kim, D. An Applicable Predictive Maintenance Framework for the Absence of Run-to-Failure Data. *Appl. Sci.* **2021**, *11*, 5180. https://doi.org/10.3390/app11115180

Academic Editors: Eui-Nam Huh and Juan Carlos Cano

Received: 3 February 2021
Accepted: 28 May 2021
Published: 2 June 2021

1. Introduction

Recently, Developments in the Industrial Internet of Things (IIoT) have made it possible to collect and process large amounts of data in sensors and computer-connected machinery [1]. In addition, it is possible to analyze not only the process data generated during the production of the product but also the interest in the equipment's Prognostics and Health Management (PHM) based on the equipment data, which is the data of the equipment itself [2]. Initial PHMs, called reactive maintenance, are used until the equipment fails and the equipment is repaired after the failure. This method has a low repair cost but leads to a very expensive incident in the case of equipment failure. To overcome these shortcomings, time-based maintenance (TBM) and condition-based maintenance (CBM), called preventive maintenance, have emerged. TBM is a method of repairing equipment regardless of the failure over the known life of the equipment. The disadvantage of this is that more maintenance than necessary is performed because maintenance is performed at a predetermined time, even without an event of failure. In addition, CBM is a method to perform maintenance when an abnormal symptom of equipment occurs [3]. It is a more advanced method than TBM because it performs maintenance when an abnormal indication occurs. However, an abnormal indication does not necessarily mean that the equipment is malfunctioning because there is also the problem of false alarms. In order to overcome this drawback, Predictive Maintenance (PdM) has recently been proposed. PdM continually detects anomalies in the equipment through condition monitoring, and

then determines when the equipment is predicted to fail using predictive models [4]. This allows the engineer to maximize the uptime of the equipment and significantly reduce the cost of the incident by preparing parts for maintenance in advance [2].

These general PdM methodologies can be broadly classified into two categories: physical model-based and data-based methodologies [5]. First, the methodology based on the physical model predicts the failure of a facility by using a known failure model or by a mathematical model. These are methodologies that rely on the experience and knowledge of engineers, which are very limited and inadaptable to a variety of environments and complex real-world applications [6–8]. Conversely, the data-based methodology uses statistical or machine learning algorithms to calculate the health of the equipment based on the sensor data such as vibration, temperature, and pressure. This builds a health model of the equipment without any assumptions about the equipment, provided there are enough data to model it. This is a more flexible and applicable methodology since, in practice, the equipment is exposed to a variety of environments and is likely to behave differently from previously defined models. In addition, this methodology is essential because, for complex equipment, it is nearly impossible to produce realistic models [5,7]. Recently, research into data-based PdM has been actively conducted due to the development of sensors and computing technologies [9]. Data-based methodologies can also be classified according to whether run-to-failure data exist or not as follows: a supervised learning method with failure data, and an unsupervised learning method without failure data [7]. The methodology using supervised learning must have sufficient target value for learning; that is, the failure history data called so-called run-to-failure data. Reliable models created using supervised methods can only be trusted if there is a sufficient history of failure [8].

However, in the real world, there are very few cases where there is sufficient failure history data. First, the equipment data that occur in real-time is rarely stored as a whole. Second, it is unlikely that data will be present when the actual failure occurred by fixing the equipment before the failure. Third, it is difficult to store the data generated at the specific time of failure. Therefore, it is not easy to build an applicable model in this way [10]. To overcome these drawbacks, some methodologies define a target Health Index (HI) curve so that failure history data can at least be built on models [6,10–12]. This allows the model to learn data by looking at the time of failure as 0 and defining the Health Index degradation curve of the equipment [13]. However, this method also has the following problems. First, the predefined Health Index degradation curves may not fit the actual equipment. Second, even under the same conditions, different Health Index degradation curves may exist. Third, the Health Index degradation curve may vary depending on initial conditions, such as maintenance conditions [14]. Therefore, many existing studies have assumed that there is a failure history data or have generated failure data through simulation to prove the effectiveness of the proposed method [15,16]. However, this is also not applicable to the real methodology. Also, there are some methods for the unsupervised-based model [17–19]. These studies only applicable when there are specific data sources: time waveform data or Fast Fourier Transform (FFT) data. When a device acquired a vibration signal and its amplitude plotted against time, it is called a time waveform. And FFT data are induced by applying FFT to the time waveform, and it is described by amplitude against frequency domain.

In this paper, we propose a new PdM framework that applies to real situations. The proposed method is applicable even when there is no fault history data. In addition, since each model is made for each equipment, there is a possibility that the model can be elaborately updated if the failure history is accumulated later. The proposed method consists of three main elements. First, the raw data of the equipment are preprocessed to make it applicable to the model. Second, the model is made by inputting the preprocessed data into the autoencoder model as the training data, and the Health Index is made by comparing it with the training data when the new data are entered. Third, the remaining useful life of the equipment is predicted based on the HI pattern. The main contributions of this paper are summarized as follows:

- In order to be applicable in the field, we propose an autoencoder (AE) based methodology. This is an algorithm that can be modeled directly from the normal data without the failure data.
- In order to apply various models in the future, we divided the steps of making HI and of predicting RUL in the proposed framework. This is the standard applied in recent PdM studies [20].
- The proposed framework is applied to real data cases, not simulation data, to prove the practicality and feasibility of the proposed methodology.

The remainder of this paper is organized as follows: In Section 2, the basic theory of autoencoder and the prediction method is presented in detail. In Section 3, we describe the details of the proposed framework, which are applicable even in the absence of run-to-failure data. In Sections 4 and 5, we describe the application of the proposed framework to real cases to demonstrate the practicality of the proposed method. We conclude the paper in Section 6 and discuss future directions.

2. Background

2.1. Autoencoder (AE)

Autoencoder is one of the artificial neural networks used to learn data by unsupervised learning [21]. Unlike other neural networks, the basic purpose of an autoencoder is to learn the representation of the data. In other words, the autoencoder is trained to construct a neural network whose input and output are identical, learning the reconstruction of the input data while training the networks [13]. To enable this, the autoencoder consists of three layers: an input layer, a hidden layer, and an output layer. The basic structure of an autoencoder is shown in Figure 1 below.

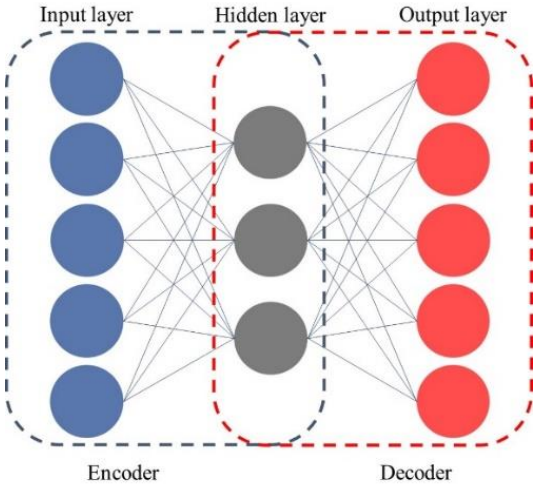


Figure 1. The basic structure of an autoencoder.

Autoencoder can be divided into two parts: the encoder and the decoder. The input layer and the hidden layer are called the encoder, and the hidden layer and the output layer are called the decoder. In this case, the autoencoder can be expressed as below.

Given an input dataset $\mathbf{x} = \{x_1, x_2, \ldots, x_p\} \in \mathbb{R}^{1\times p}$, the encoder transforms the input data to the hidden representation $\mathbf{h} = \{h_1, h_2, \ldots, h_n\} \in \mathbb{R}^{1\times n}$, the expression of an encoding process can be described as in (1)

$$\mathbf{h} = f_{activation}(\mathbf{W}_{xh}\mathbf{x} + \mathbf{b}_{xh}) \quad (1)$$

where $\mathbf{W}_{xh}$ and $\mathbf{b}_{xh}$ are the weight matrix and bias vector of the neural network, and $f_{activation}$ is the activation function which is the nonlinear mapping function.

Subsequently, the decoder takes the hidden representation from the encoder as the input and reconstructs the original input data $\mathbf{x}$. The decoder maps the hidden representation

to the original input data in the same way as the encoder maps the original input to the hidden representation. The expression of the decoding process can be defined as follows:

$$\mathbf{z} = g_{activation}(\boldsymbol{W}_{hx'}\mathbf{h} + \mathbf{b}_{hx'}) \tag{2}$$

where $\boldsymbol{W}_{hx'}$ and $\mathbf{b}_{hx'}$ is the weight and bias of the network, $g_{activation}$ is the nonlinear mapping function and $\mathbf{z} = \{z_1, z_2, \ldots, z_p\} \in \mathbb{R}^{1\times p}$ is called the reconstructed data.

The difference between original input data and reconstructed data is a reconstruction error as in (3).

$$\textbf{\textit{reconstuction error}} = ||\mathbf{x} - \mathbf{z}|| \tag{3}$$

The autoencoder learns to minimize this reconstruction error using the backpropagation algorithm. In general, the mean square error (MSE) or the mean absolute error (MAE) is chosen as the loss function of the autoencoder. In this paper, the mean absolute error is used as the loss function because the minimizing process using the mean square error can be extremely affected by the noise or outliers in the input data [13]. In this paper, we use standardized data as the input data, not normalized data. Unlike deep learning, especially the convolutional neural networks (CNNs), equipment sensor data often fall outside the normal range. The standardization formula is expressed in (4).

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{4}$$

where μ is the mean of given data, σ is the standard deviation of given data.

In addition, the nonlinearity of the autoencoder depends not only on the number of nodes in the hidden layer but also on the activation functions $f_{activation}$ and $g_{activation}$. In this paper, $f_{activation}$ uses tanh, one of the most commonly used functions, and $g_{activation}$ uses the linear function to reconstruct the input data. Since standardized data are used as the input data, the output may theoretically have a range of real numbers. Therefore, if using a function such as sigmoid or relu as $g_{activation}$, it would be difficult to restore the input data, and it is better to use the linear function as $g_{activation}$. Moreover, we use the difference between the input data and the output data of the autoencoder to determine whether the data are normal or not. This is quite reasonable because the autoencoder has learned the relationship between the input data variables. As the autoencoder learns the training data, if the test data are similar to the input data, the reconstruction error between the input and the output will be small. On the contrary, when data which differs from the training data are used as the test data, the reconstruction error between the input and output will be large [22]. Various studies use these characteristics [23–25]. However, autoencoders are often used for fault detection only [26], or feature extraction [14]. Even if the autoencoder is used to create the HI, the HI curve is assumed [13].

In this paper, we find the difference between the input data and the output data by calculating the variable-wise value using the mean absolute error (MAE) function as defined in the Health Index (HI). In other words, the Health Index is:

$$Health\ Index = \frac{1}{n}\sum_{i=1}^{p}|\mathbf{x} - \mathbf{z}| \tag{5}$$

where n is the number of sample size, p is the number of variables, $\mathbf{x}$ is the input data, and $\mathbf{z}$ is the output of autoencoder, the reconstructed data. Because the loss function and the HI have the same formula, a well-trained autoencoder will have smaller values for data that are similar to the training data, so it is natural to define a HI like this way. In this paper, the framework is proposed to be used when there is little run-to-failure data, and for this, the simplest type of autoencoder is proposed. If data have more variables or complex patterns, a deeper model can be used.

2.2. Regression

After building the HI, if the index score continues to rise, it can be assumed that the equipment is out of order. That is, if the HI of the new data is larger than the score of the

trained data, it can be said that abnormal data is generated from the equipment, and if there is a trend of the index score, the failure of the equipment can be predicted. Therefore, it is necessary to regard the HI score as time series data and predict the future score of the HI. Regression is a simple, fundamental method among the various algorithms used for time series prediction. In particular, linear regression is a linear approach to modeling the relationship between a dependent variable and one or more independent variables. Among them, the simple linear regression is the simplest prediction method that can be used when there is only one variable in the data.

Given data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x_p}\} \in \mathbb{R}^{n \times p}$, the general form of the linear regression can be expressed as follows:

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} +, \ldots, + \beta_p x_{ip} + \epsilon_i \\ &= \mathbf{x_i^T}\boldsymbol{\beta} + \epsilon_i, i \in \{1, \ldots, p\} \end{aligned} \tag{6}$$

where T denotes the transpose, so that $\mathbf{x_i^T}\boldsymbol{\beta}$ is the inner product between vectors $\mathbf{x}_i$ and β. And y_i is a dependent variable, x_{ip} is the p-th independent variable of timestamp i, respectively. Furthermore, β_0 is an intercept, β_p is the p-th coefficient of each independent variable and ϵ_i is an error term. In this paper, as the HI is adopted for the regression's input vector, a simple linear regression is used. A least-squares estimator is utilized to fit the model and tries to minimize the sum of squares of an error term. The least-squares estimator finds the optimal solution, seeking the slope β_1 and the intercept β_0 for $\arg\min_{\beta_0, \beta_1} S(\beta_0, \beta_1)$ as in (7)

$$\begin{aligned} S(\beta_0, \beta_1) &= \sum_{i=1}^{m} (\epsilon_i)^2 \\ &= \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned} \tag{7}$$

where m is the number of training data.

In this paper, the slope represents the trend of the HI. For example, if the health of the equipment is good, then there is no trend in the HI, the slope of the regression model can be either very small or negative. On the contrary, when an abnormality or aging occurs in the equipment, the trend of the HI may rise, and the slope value of the regression model becomes large.

3. Proposed Framework

The proposed methodology can be applied immediately regardless of whether run-to-failure data are less or not exists in the first place. The proposed framework is composed of the following three technical processes: Acquisition and preprocessing of data, building model for HI construction and RUL prediction. The proposed framework is shown in Figure 2.

3.1. Acquisition and Preprocessing of Data

The first step is the acquisition and preprocessing of data, which are the foundation of the framework. In this step, if there is no historical failure data, the equipment data measured by the sensor, i.e., vibration, current, temperature data, etc., in real-time are required. At this time, selecting an area of data that can be estimated as normal based on an engineer or a priori knowledge should be determined. Correct definitions of normal data can be used to build sophisticated models. Second, it may be necessary to select important variables or extract meaningful features. Third, remove non-continuous variables or remove constant variables. Besides, the performance of the model can be improved through preprocessing for the model, i.e., standardization and summary statistic in the case of autoencoder, and the resulting training data.

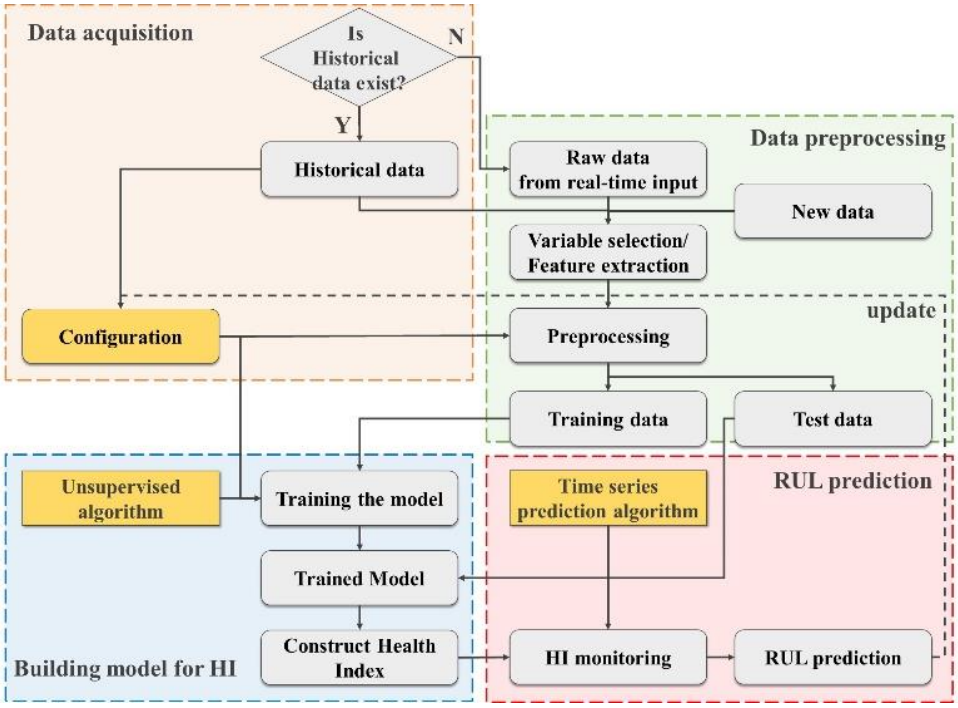


Figure 2. The proposed framework for an applicable Predictive Maintenance System in the absence of run-to-failure data.

The configuration comprises the information you need to set the model. For example, it includes information such as the time of failure, variable information, and the threshold to determine failure. A threshold is defined by the value of HI when the equipment fails. This same process should be followed if historical run-to-failure data are present. The difference is that there is a failure history so that various information about the failure can be saved in the configuration for later model use.

3.2. Building a Model for HI

The unsupervised learning model learns based on preprocessed training data. In this paper, a trained autoencoder is learning the structure of normal data. When the model is built, the following steps are taken:

- As mentioned in Section 2, it can be calculated by defining the HI with the MAE of the input and output data. Observe the calculated HI which is based on the result of the trained autoencoder. This is to guarantee how the HI is normal and to generate the initial threshold. In general, the HI for the train data will usually be small and uniform if the training is done well.
- The determination of an initial threshold is very challenging. If there is historical run-to-failure data, the initial threshold can be set using the failure data [13]. However, if there is no fault history, the exact threshold is unknown. So, in general, it mostly assumes an arbitrary threshold [6]. The initial threshold can be provided by an expert or determined based on the HI calculated from the training data. In this paper, it is proposed to use a gaussian distribution-based value because z-normalization was applied to the preprocessing method. The general manufacturing process manages each variable with 3-sigma based on the process control method [27]. That is, it is heuristically proposed that the threshold is 3 based on the data normalization and the manufacturing process control method. For example, if the 2-sigma method is utilized, we can use 2 as the threshold. Therefore, it is a method that can be used even if run-to-failure data do not exist, and it is expected that the threshold can be updated

when run-to-failure data accumulates. In this case, we use the most common 3-sigma method in process control; a value of 3 can be used as a threshold.

- Whenever new data are collected from the equipment, it can be preprocessed based on the training data, and HI can be calculated using the trained autoencoder. In other words, preprocessing is based on the mean and standard deviation of the training data. If there are some outliers or noise in the new data, a high value of HI can be made despite the normal condition of the equipment. Therefore, it is necessary to focus on the overall trend rather than focusing on each HI value.

3.3. Predict the Remaining Useful Life (RUL)

As mentioned in Section 2, when aging or abnormality occurs in the equipment, the trend of the HI may arise. Then, it is possible to predict the future value of the HI using time series prediction algorithms. When an incident ends, the configuration can be updated based on the failure data. In this way, the more the proposed framework is applied, as knowledge of failure data is accumulated, the performance can be improved [28]. In this paper, simple linear regression is used. Similarly, RUL can be calculated using regression. The RUL is the difference between the current time (index) and the predicted failure time where the regression prediction line meets the threshold. In addition, regression has the advantage of providing more accurate predictions because it can give confidence intervals for future time points. This is illustrated in Figure 3.

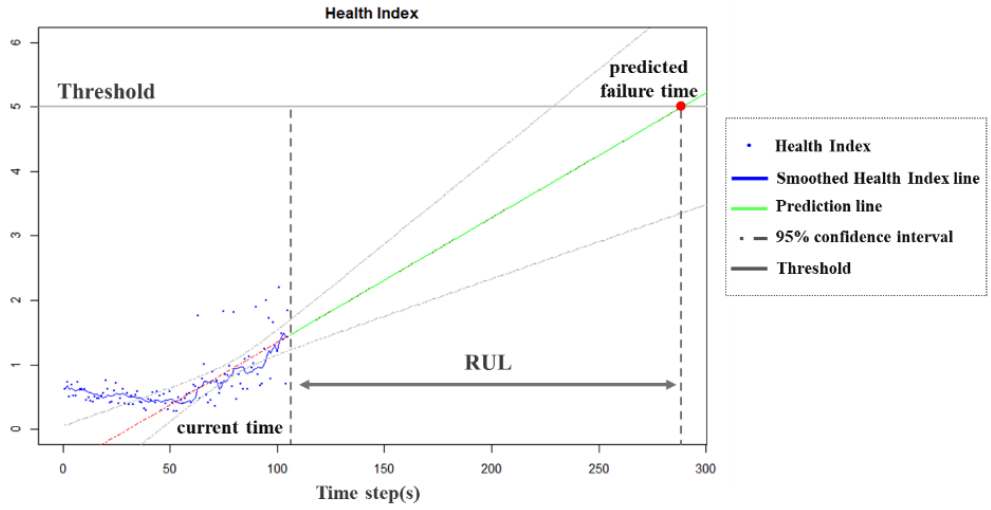


Figure 3. The example of the Health Index and the RUL prediction.

In this case, the number of samples for fitting the regression line is important. This is because in general, the closer the failure point, the more spurred abruptly the HI changes. Figure 4 compares the results of the last 50 samples with the total sample. Fitting to the full sample makes it difficult to predict the exact RUL even if the failure is imminent while fitting to the latest sample can predict a more accurate RUL.

To demonstrate the performance of the proposed methodology, In Sections 4 and 5, we conducted a case study on real data cases: a pump and a robot arm case.

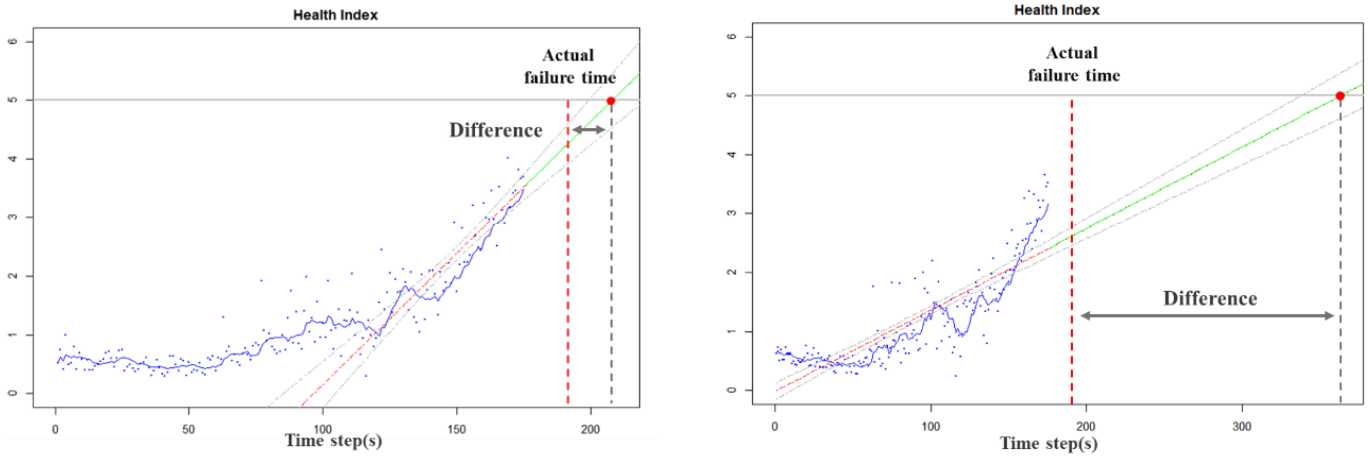


Figure 4. The example of the Health Index and the RUL prediction difference. (**a**) fitting results of the last 50 samples; (**b**) fitting results of the total samples.

4. Case Study 1

Normally a pump is used to improve the environmental conditions for manufacturing facilities. As time passes, pump aging causes problems with the components of the pump. The purpose of the analysis is to monitor the condition of the pump in real-time and detect abnormalities based on the plant data. It also predicts the RUL in advance to maximize the pump's available time. We run the code on a laptop machine with an Intel(R) Core^{TM} i7-7700HQ (2.80 GH) CPU and 16 GB of system memory. Programming language is R 3.6, and the model uses Keras library (https://keras.rstudio.com/, accessed on 14 July 2020) to implement the autoencoder.

4.1. Data Description

The data is generated from the pump equipment and consists of pressure, temperature, vibration, power. One pump is composed of a A-pump and B-pump and the example of the overall structure is in Figure 5.

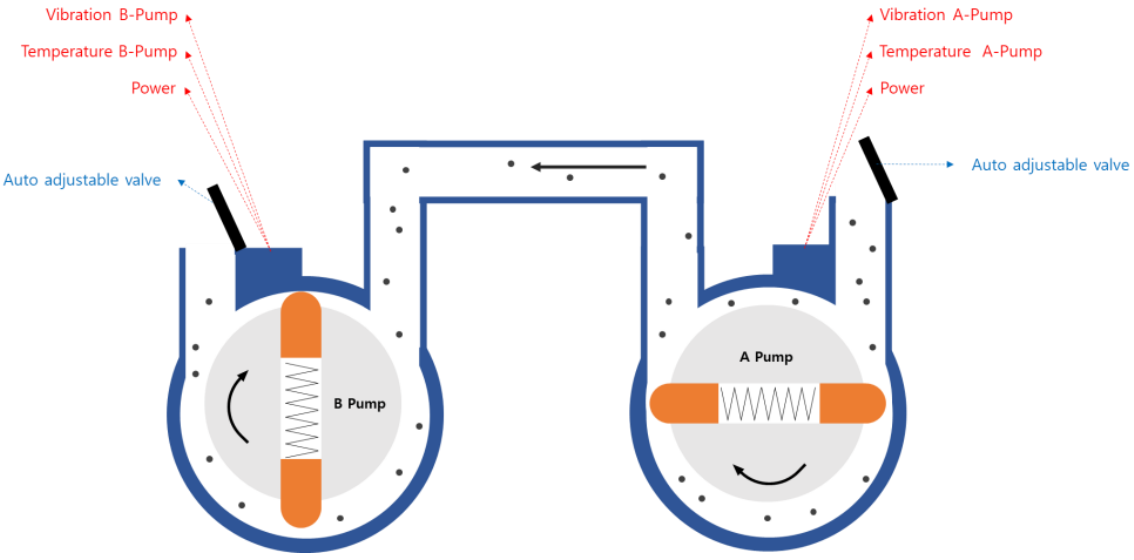
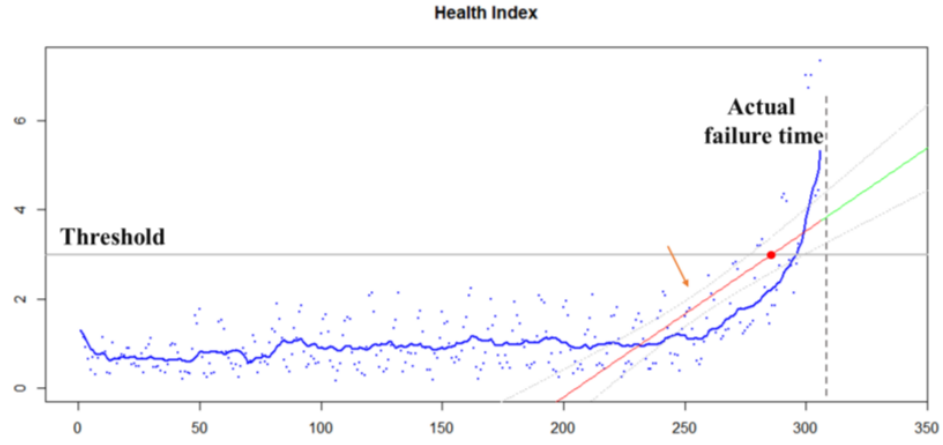


Figure 5. The example of an overall structure of a pump.

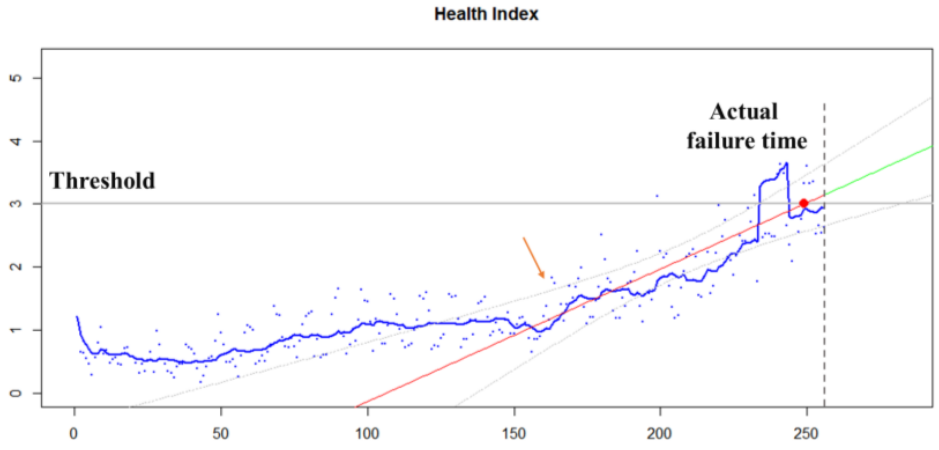
Due to the security policy, all of the variable names are masked. Data description are as follows: The first equipment is about 6 h of data and consists of 10 variables. The second equipment is about 5 h of data and consists of 10 variables. Finally, the third equipment is 50 h of data and consists of 8 variables. The data are collected at 1-min intervals and use the equipment sensor data to monitor the condition of the pump. It consists of summary data rather than high-period data such as vibration spectrum data. All the pumps are of different types and the failures have occurred in different parts. Because of the very small amount of failure data, modeling could not be done by supervised learning. In addition, the small amount of data made it difficult to apply CNN-like methods.

4.2. Experimental Design

The run and idle states of the pump are delivered by the engineer and preprocessed based on the run state only, and the normal state is arbitrarily specified and applied to the autoencoder model. Also, the initial threshold was arbitrarily assigned. In this experiment, we assumed that the starting 100 data points of a given dataset were stable and normal. Only standardization and removing constant variables were used without further pretreatment. The number of nodes in the hidden layer of the autoencoder was set to 4, which is half the number of variables. The optimizer used RMSprop. The entire data including the training data are applied to the trained autoencoder to monitor the HI trend at the time of failure. The experimental results are shown in Figure 6 below.

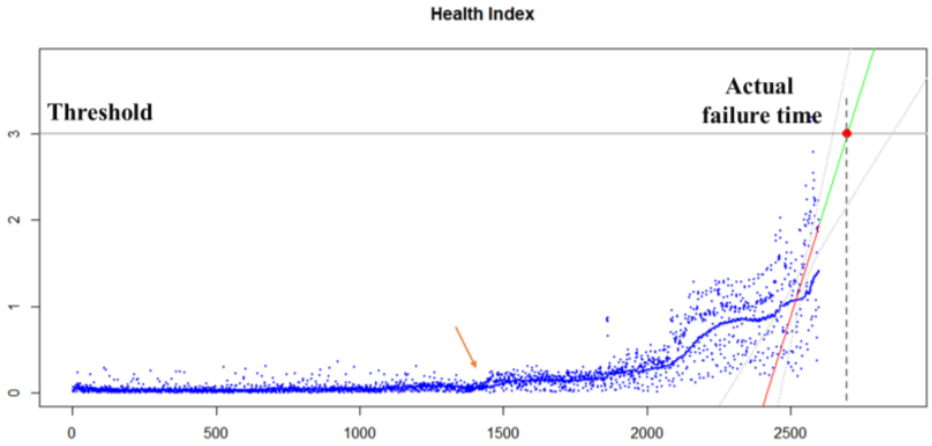


(**a**)



(**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. The result of pump data. (**a**) Constructed HI model on pump No. 1; (**b**) Constructed HI model on pump No. 2; (**c**) Constructed HI model on pump No. 3.

4.3. Experimental Result

As shown in Figure 6, the HI model created after setting up the normal interval detected the failure well. In particular, the HI value increased before the failure occurred, enabling the engineer to predict the failure in advance.

In addition, as indicated by the orange arrow in Figure 6, there is a section where the HI rises sharply. This is difficult to detect with the Statistical Process Control (SPC). In other words, the autoencoder learns the nonlinear structure of the data so that the correlation of the data is reflected. Thus, the change in correlation between the variables affected the rise of the HI value, and when there was a trend, the RUL could be predicted.

It is also confirmed that each fault class has a different threshold. This can be affected by hyper-parameters and configurations [29]. However, having a different threshold for each failure type means that failure types can be classified if a lot of failure data are accumulated later. In other words, the proposed framework can construct HI, predict RUL, and provide expected failure types.

5. Case Study 2

One of the most used pieces of equipment in the production line is the robotic arm. In today's automated production lines, robot arms play a significant role, since they are used in hazardous and repetitive work environments [30]. In particular, the production line of automobiles is connected to numerous pieces of equipment. The cost of failure is very high because the entire line must be stopped if one piece of equipment fails.

5.1. Data Description

The data are equipment sensor data generated from the vibration sensor which is attached to the edge of the robot arm. The data have three-axis: x, y, and z. For example, vibration variables were extracted from statistical variables such as data mean, standard deviation, and maximum for every two seconds.

Raw data are collected at an approximate rate of, on average, 1500 samples per day. Since failure rarely occurred and the data collection period was over three months, the daily average was used instead of seconds for ease of analysis. Due to the security policy, all of the variable names are masked. Detailed data descriptions are in Table 1 and Figure 7 shows the variables of the data. In particular, different types of failures occurred in one piece equipment.

Table 1. The details of robot arm data.

No.	Size of Data (Time Span)	Number of Variables	Fault Class	Name of Equipment
1	113 points (about 4 months)	20	N1	K1
2	258 points (about 8 months)	20	N3	K1
3	289 points (about 9 months)	20	N2	K2
4	134 points (about 4 months)	20	N3	K2
5	84 points (about 3 months)	20	N1	K2

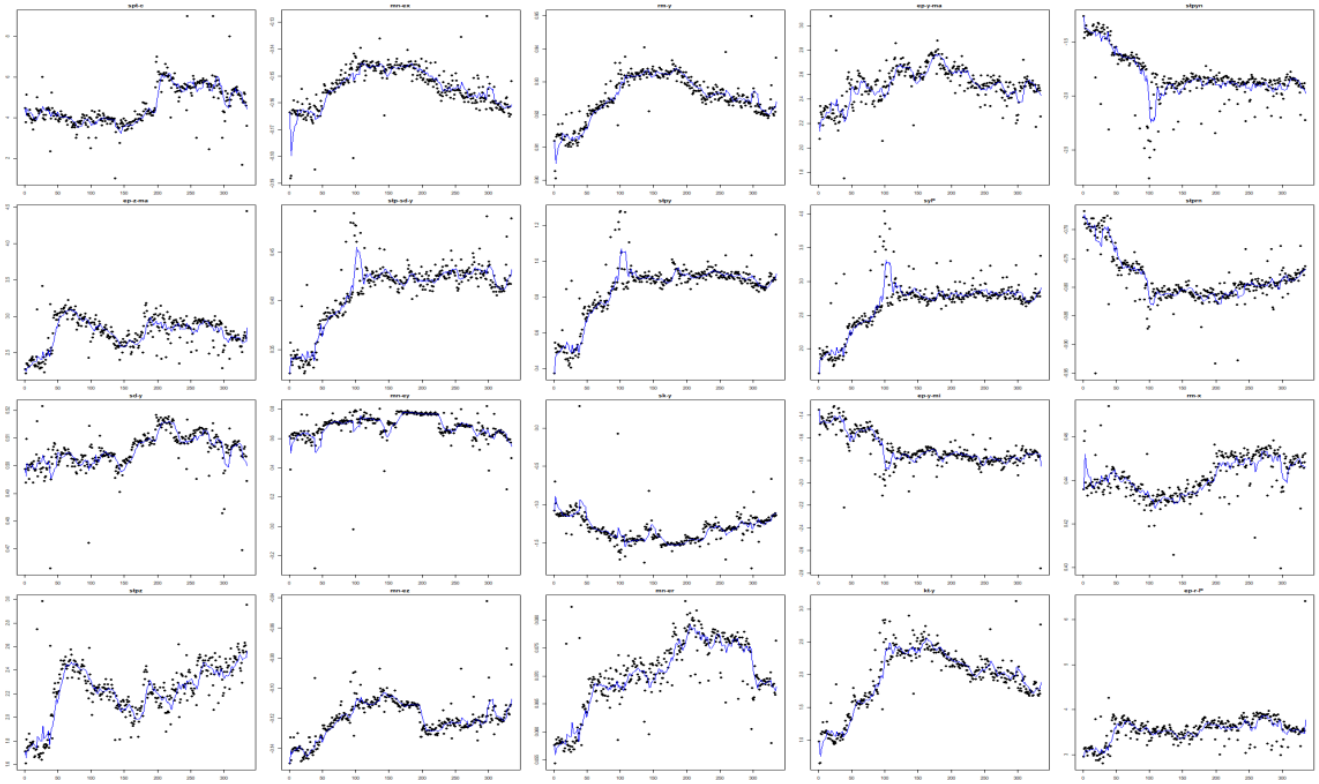


Figure 7. The variable plot of robot arm data.

5.2. Experimental Design

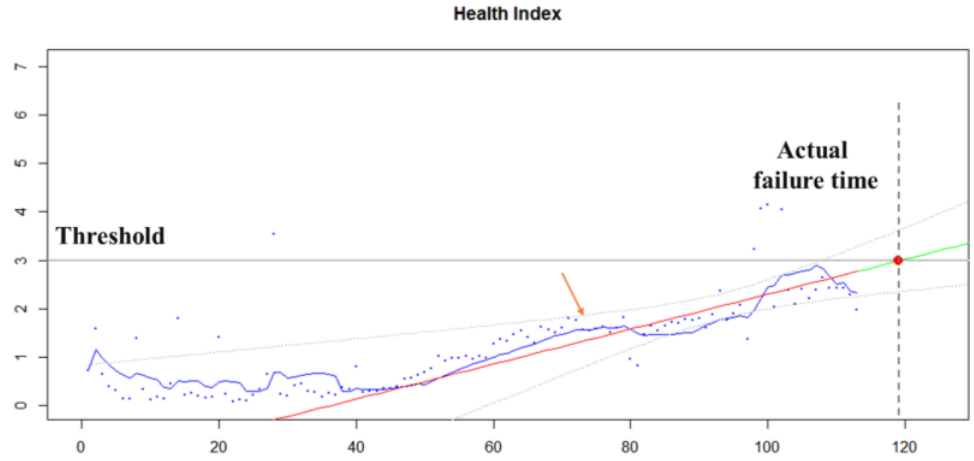
In this experiment, we assumed that the starting 50 data points of a given data that were stable and normal. The normal state is arbitrarily specified and applied to the autoencoder model. Also, the initial threshold was arbitrarily assigned. The rest of the other settings are the same in case study 1.

5.3. Experimental Result

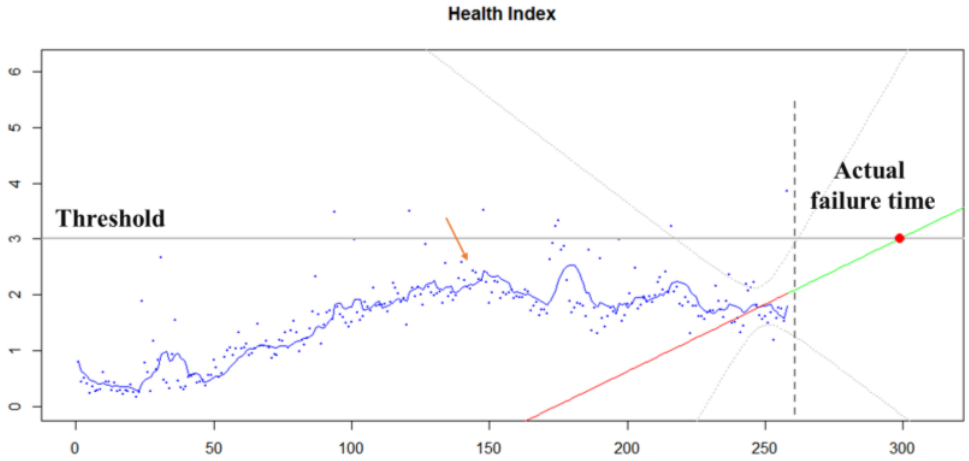
The experimental results are shown in Figure 8 below.

As shown in Figure 8, the HI model created after setting up the normal data points detected the failure well. Furthermore, the HI value rises before the failure occurs, allowing prediction of the failure in advance. In addition, at the orange arrow in Figure 8, there is a section where the HI rises sharply. These signs, which occur just before the failure, help engineers to detect anomalies and predict equipment failure.

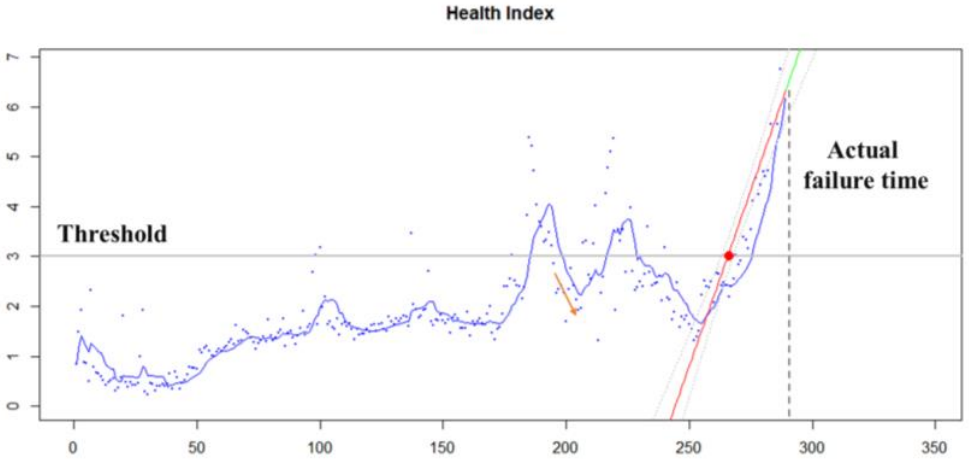
Originally, the same failure type was expected to have similar thresholds regardless of the equipment. However, there is a difference in the threshold for each fault class. It can be inferred that even with the same failure, the degree of failure is different. That is, it may have different thresholds due to the condition of the equipment the manufacturer of the equipment. This is one of the most difficult parts of predicting failure. The result is shown in Figure 9.



(**a**)

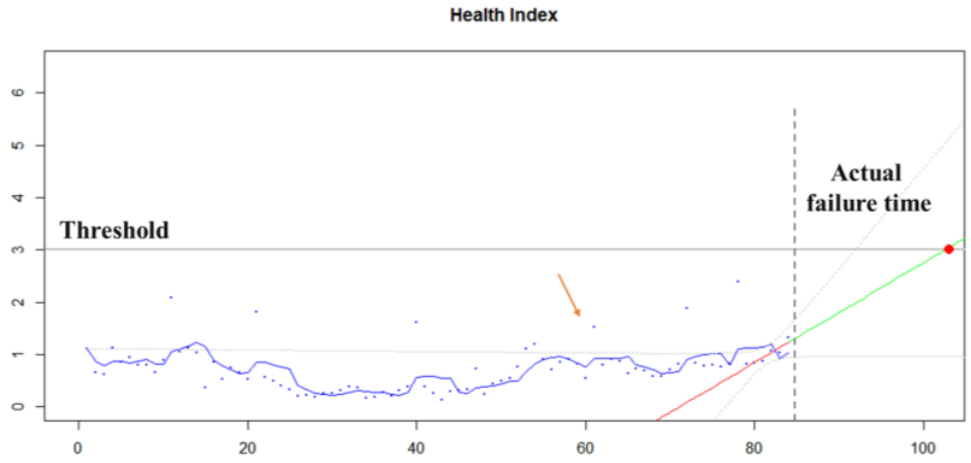


(**b**)

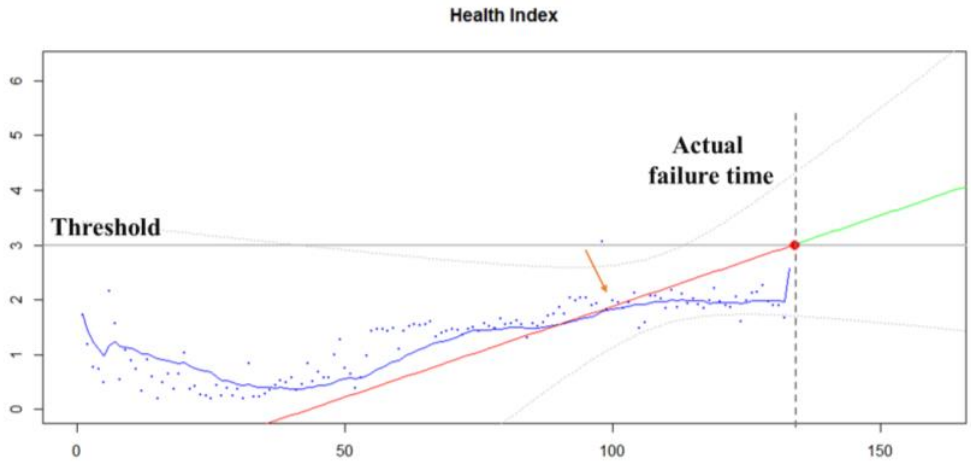


(**c**)

Figure 8. *Cont*.



(**d**)



(**e**)

Figure 8. The result of robot arm data. (**a**) Constructed HI model on robot arm No. 1; (**b**) Constructed HI model on robot arm No. 2; (**c**) Constructed HI model on robot arm No. 3; (**d**) Constructed HI model on robot arm No. 4; (**e**) Constructed HI model on robot arm No. 5.

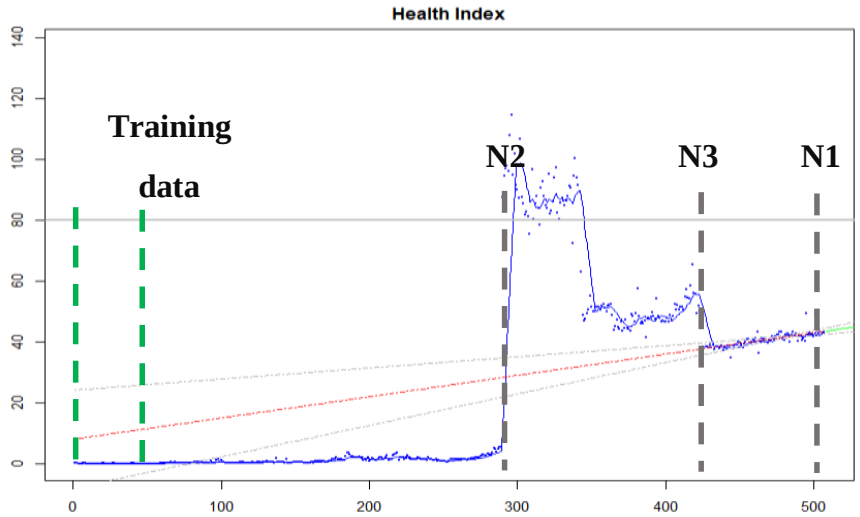


Figure 9. The result of HI without a model update.

Figure 9 depicts the results of training and testing the model with the first 50 data from equipment K2. In this case, it can be said that K2's initial trained model continued to be used after failure. Equipment K2 had the first failure of N2 and was restarted after maintenance. However, the HI value remains high after maintenance. In addition, even the HI value decreases at the time of N3 failure.

Comprehensively, it can be inferred that maintenance does not simply return the equipment to its original state. Even the same equipment may be in a different state from the normal state after maintenance. This implies that separate models are required for each equipment. Therefore, after the maintenance of the equipment, the model should be newly re-trained using the normal state data obtained after the new operation, rather than using the initial trained model. Then the failure can then be properly predicted as shown in Figure 9.

6. Additional Experiment

To verify the reliability of the proposed method, additional experiments were conducted. One additional experiment was a simple experiment investigating how the RUL was predicted, and an isolation forest was used as a comparison method [31,32]. Isolation forest is a tree-based ensemble method, which is widely used in outlier detection. The hyper-parameter of the isolation forest was selected through grid search, and the number of trees used was 500. The isolation forest outputs a probability value. In this case, the theoretically possible value is between 0 and 1, but in practice, the normal data have a value of about 0.5. Therefore, the failure threshold of this model is defined as the average of these, 0.75. In the general principles of reliability theory, this is a more sensitive threshold because it is assumed that about 15% of the total length are data in which failure occurs [33]. The experiments were all conducted under the same conditions as in the Sections 4 and 5. As the training data for regression, 10 min of data for the pump and 1 week of data for the robot arm were used, respectively. This was settled according to data length. The test samples consisted of data 1 h before failure for each pump, and 1 month before failure for each robot arm, and this was presented by the engineer. Hence, this figure is the starting point of the RUL prediction that the engineer wants to know. Root Mean Square Error (*RMSE*) and Mean Absolute Error (*MAE*) are used as measurements for evaluating performance [34,35], which can be computed by the following equations:

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|d_i| \tag{9}$$

where n is the total number of test data samples. $d_i = RUL_i' - RUL_i$ is the difference between the predicted RUL and the actual RUL for the i-th test data sample. All experiments were performed 10 times, the average was calculated, and the result is shown in the Table 2.

Table 2. The result of additional experiment.

	Pump Case 1		Pump Case 2		Robot Arm Case 1		Robot Arm Case 2		Robot Arm Case 3		Robot Arm Case 4		Robot Arm Case 5	
Method	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
Proposed method	378.67	158.07	47.64	15.72	92.08	68.82	78.32	57.75	116.89	71.59	336.05	90.90	20.65	5.01
Isolation forest	888.69	509.43	42.38	37.54	231.30	221.70	84.23	67.35	120.89	111.29	456.76	206.77	278.68	125

Table 2 shows the experimental results of the proposed method and the comparative method according to the measurements. The proposed method outperformed with 6 datasets and 6 datasets out of 7, where we used RMSE and MAE as measurements, respectively. In pump cases, the proposed method alternately showed good results in RMSE and MAE. In pump case 1, the threshold was somewhat inaccurate, resulting in relatively

poor results. In the robot arm cases, the proposed method showed better results in all cases. These results indicate the effectiveness of the proposed method. In particular, in robot arm 5, the threshold was fit, so an accurate result could be obtained. On the contrary, the reasons isolation forest produced poor results are as follows. First, the difference in the score of the failure compared to the normal is small; this is because the isolation forest does not try to minimize the reconstruction error of the normal data, unlike an autoencoder which tries to lower the score value of the normal data. The isolation forest, unlike the autoencoder, does not try to reduce the score value of the normal data but tries to identify the score difference between normal and faulty data. In addition, the change rate of score compared to normal data decreases as data goes to failure. That is, score-based RUL prediction is relatively difficult. Since it is relatively difficult to estimate the threshold, the usefulness is somewhat reduced when there is little fault data. Through the additional experiment, the effectiveness of the proposed method could be confirmed.

7. Conclusions

In this paper, we propose a Predictive Maintenance (PdM) framework that can be applied even in the absence of run-to-failure data. In particular, if only normal data can be defined, it is a methodology that can perform PdM. Many existing studies have relied on simulation data. In addition, because it is essential to have a considerable amount of run-to-failure data, it was challenging to apply instantly to the industry despite the high experimental accuracy. However, this paper proposes a framework based on autoencoder and simple linear regression to generate and monitor models even in the absence of run-to-failure data. Since the proposed methodology is roughly divided into the construction part and prediction parts of the HI, relatively simple algorithms are used to demonstrate the framework in this paper. Other unsupervised learning that suitable for the industry can be used in addition to the autoencoder. Furthemore, it is possible to improve the accuracy of the model by using a more appropriate prediction methodology in addition to simple linear regression. Furthermore, the configuration can be updated based on the failure data. In this way, a more the proposed framework is applied, as knowledge of failure data is accumulated, the more improve the performance.

The proposed framework was carried out in two different real cases, even though they are a completely different domain, confirming the usefulness and applicability of the proposed methodology. In case study 1, abnormal signs were detected prior to the time of failure. In addition, because the failure type has different thresholds, the possibility of failure type classification was also confirmed. If there is no run-to-failure history, the initial accuracy of the proposed methodology may be low, but we can increase the accuracy of the model by re-training the model, as shown in case study 2. More sophisticated models can be performed if there is a history of past failures or with the knowledge of the industry. In the case studies, we used summary data rather than complicated preprocessing, and the feasibility is increased. Since feature extraction generally promotes the capability of the network-based Predictive Maintenance model [31]. In addition, through a simple additional experiment, the effectiveness of the proposed method was confirmed.

In addition, this study raised the value of research by presenting difficulties in advance that can be experienced in actual application. However, there is some limitation of the proposed method. In this paper, the normal data points are determined more or less arbitrarily as is the threshold. In future research, the threshold needs to be defined based on the advanced algorithm, normal points in the data should be determined by someone who understands the industry well.

Author Contributions: D.K. (Donghwan Kim) proposed the idea and carried out all the experiments, S.L. validated the process and D.K. (Daeyoung Kim) guided the research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the World Class 300 Project (R&D) (S2641209, "Improvement of manufacturing yield and productivity through the development of next generation intelligent Smart manufacturing solution based on AI & Big data") of the MOTIE, MSS (Korea).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prieto, L.P.; Rodríguez-Triana, M.J.; Kusmin, M.; Laanpere, M. Smart school multimodal dataset and challenges. *CEUR Workshop Proc.* **2017**, *1828*, 53–59. [CrossRef]
2. Liu, Y.C.; Chang, Y.J.; Liu, S.L.; Chen, S.P. Data-driven prognostics of remaining useful life for milling machine cutting tools. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–5. [CrossRef]
3. Heng, A.; Zhang, S.; Tan, A.C.C.; Mathew, J. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mech. Syst. Signal Process.* **2009**, *23*, 724–739. [CrossRef]
4. Aggarwal, K.; Atan, O.; Farahat, A.K.; Zhang, C.; Ristovski, K.; Gupta, C. Two Birds with One Network: Unifying Failure Event Prediction and Time-to-failure Modeling. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1308–1317. [CrossRef]
5. Zheng, S.; Ristovski, K.; Farahat, A.; Gupta, C. Long Short-Term Memory Network for Remaining Useful Life estimation. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 88–95. [CrossRef]
6. Yoo, Y.; Baek, J. A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network. *Appl. Sci.* **2018**, *8*, 1102. [CrossRef]
7. Zhong, S.; Fu, S.; Lin, L.; Fu, X.; Cui, Z.; Wang, R. A novel unsupervised anomaly detection for gas turbine using Isolation Forest. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–6. [CrossRef]
8. Jin, B.; Chen, Y.; Li, D.; Poolla, K.; Sangiovanni-Vincentelli, A. A one-class support vector machine calibration method for time series change point detection. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–5. [CrossRef]
9. Zhao, G.; Zhang, G.; Liu, Y.; Zhang, B.; Hu, C. Lithium-ion battery remaining useful life prediction with Deep Belief Network and Relevance Vector Machine. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 7–13. [CrossRef]
10. Yao, Q.; Yang, T.; Liu, Z.; Zheng, Z. Remaining useful life estimation by empirical mode decomposition and ensemble deep convolution neural networks. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–6. [CrossRef]
11. Si, X.S.; Wang, W.; Hu, C.H.; Zhou, D.H. Remaining useful life estimation—A review on the statistical data driven approaches. *Eur. J. Oper. Res.* **2011**, *213*, 1–14. [CrossRef]
12. Ma, M.; Mao, Z. Deep recurrent convolutional neural network for remaining useful life prediction. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–4. [CrossRef]
13. Lin, P.; Tao, J. A novel bearing health indicator construction method based on ensemble stacked autoencoder. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management, San Francisco, CA, USA, 17–20 June 2019; pp. 1–9. [CrossRef]
14. Malhotra, P.; TV, V.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder. *arXiv* **2016**, arXiv:1608.06154. [CrossRef]
15. Hu, C.; Youn, B.D.; Wang, P.; Taek Yoon, J. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliab. Eng. Syst. Saf.* **2012**, *103*, 120–135. [CrossRef]
16. Shakya, P.; Kulkarni, M.S.; Darpe, A.K. A novel methodology for online detection of bearing health status for naturally progressing defect. *J. Sound Vib.* **2014**, *333*, 5614–5629. [CrossRef]
17. Yen, F.; Katrib, Z. WaveNet based Autoencoder Model: Vibration Analysis on Centrifugal Pump for Degradation Estimation. In Proceedings of the Annual Conference of the PHM Society, Online, 9–13 November 2020; pp. 1–6.
18. Li, J.; Wang, L.; Li, Y. Diagnosis method for hydro-generator rotor fault based on stochastic resonance. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–5. [CrossRef]
19. Liang, P.; Deng, C.; Wu, J.; Yang, Z.; Zhu, J. Intelligent fault diagnosis of rolling element bearing based on convolutional neural network and frequency spectrograms. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–5. [CrossRef]
20. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [CrossRef]
21. An, J.; Cho, S. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. *Tech. Rep.* **2015**, *21*, 1–18.

22. Sato, S.; Sanda, K. Degradation estimation of turbines in wind farm using denoising autoencoder model. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; pp. 1–6. [CrossRef]
23. Khan, S.; Yairi, T. A review on the application of deep learning in system health management. *Mech. Syst. Signal Process.* **2018**, *107*, 241–265. [CrossRef]
24. Kan, M.S.; Tan, A.C.C.; Mathew, J. A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mech. Syst. Signal Process.* **2015**, *62*, 1–20. [CrossRef]
25. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep Learning and Its Applications to Machine Health Monitoring: A Survey. *Mech. Syst. Signal Process.* **2016**, *14*, 1–14. [CrossRef]
26. Reddy, K.K.; Sarkar, S.; Venugopalan, V.; Giering, M. *Anomaly Detection and Fault Disambiguation in Large Flight Data: A Multi-modal Deep Auto-Encoder Approach*; Annual Conference of the Prognostics and Health Management Society: Denver, CO, USA, 2016; pp. 192–199.
27. Montgomery, D.C. *Introduction to Statistical Quality Control*, 6th ed.; Wiley: Manitoba, CA, USA, 2008; ISBN 978-0470169926.
28. Yoon, H.; Park, C.S.; Kim, J.S.; Baek, J.G. Algorithm learning based neural network integrating feature selection and classification. *Expert Syst. Appl.* **2013**, *40*, 231–241. [CrossRef]
29. Kim, D.; Park, S.H.; Baek, J. A Kernel Fisher Discriminant Analysis-based Tree Ensemble Classifier: KFDA Forest. *Int. J. Ind. Eng. Theory Appl. Pract.* **2018**, *25*, 569–579.
30. Lee, J.D.; Li, W.C.; Shen, J.H.; Chuang, C.W. Multi-robotic arms automated production line. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics, Auckland, New Zealand, 20–23 April 2018; pp. 26–30. [CrossRef]
31. Wu, T.; Zhang, Y.J.A.; Tang, X. Isolation Forest Based Method for Low-Quality Synchrophasor Measurements and Early Events Detection. In Proceedings of the 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (Smart-GridComm), Aalborg, Denmark, 29–31 October 2018; pp. 1–7. [CrossRef]
32. Staerman, G.; Mozharovskyi, P.; Clémençon, S.; d'Alché-Buc, F. Functional Isolation Forest. In Proceedings of the Asian Conference on Machine Learning, Nagoya, Japan, 17–19 November 2019; pp. 1–33.
33. Aremu, O.O.; Cody, R.A.; Hyland-Wood, D.; McAree, P.R. A relative entropy based feature selection framework for asset data in Predictive Maintenance. *Comput. Ind. Eng.* **2020**, *145*, 106536. [CrossRef]
34. Song, Y.; Shi, G.; Chen, L.; Huang, X.; Xia, T. Remaining Useful Life Prediction of Turbofan Engine Using Hybrid Model Based on Autoencoder and Bidirectional Long Short-Term Memory. *J. Shanghai Jiaotong Univ.* **2018**, *23*, 85–94. [CrossRef]
35. Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [CrossRef]