



Article

# Time-Series Explanatory Fault Prediction Framework for Marine Main Engine Using Explainable Artificial Intelligence

Hong Je-Gal <sup>1</sup>, Young-Seo Park <sup>1</sup>, Seong-Ho Park <sup>1</sup>, Ji-Uk Kim <sup>1</sup>, Jung-Hee Yang <sup>2</sup>, Sewon Kim <sup>1</sup>   
and Hyun-Suk Lee <sup>1\*</sup> 

<sup>1</sup> Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Republic of Korea; jagrhong@sju.ac.kr (H.J.-G.); eowkd27@sju.ac.kr (Y.-S.P.); shneal@sju.ac.kr (S.-H.P.); 22011893@sju.ac.kr (J.-U.K.); sewonkim@sejong.ac.kr (S.K.)

<sup>2</sup> Smart Ship Solution Department, Hanwha Ocean Co., Ltd., Seoul 04527, Republic of Korea; rell1010@hanwha.com

\* Correspondence: hyunsuk@sejong.ac.kr

**Abstract:** As engine monitoring data has become more complex with an increasing number of sensors, fault prediction based on artificial intelligence (AI) has emerged. Existing fault prediction models using AI significantly improve the accuracy of predictions by effectively handling such complex data, but at the same time, the problem arises that the AI-based models cannot explain the rationale of their predictions to users. To address this issue, we propose a time-series explanatory fault prediction framework to provide an explainability even when using AI-based fault prediction models. It consists of a data feature reduction process, a fault prediction model training process using long short-term memory, and an interpretation process of the fault prediction model via an explainable AI method. In particular, the proposed framework can explain a fault prediction based on time-series data. Therefore, it indicates which part of the data was significant for the fault prediction not only in terms of sensor type but also in terms of time. Through extensive experiments, we evaluate the proposed framework using various fault data by comparing the prediction performance of fault prediction and by assessing how well the main pre-symptoms of the fault are extracted when predicting a fault.

**Keywords:** explainable artificial intelligence; deep learning; fault prediction; long short-term memory; marine main engine; predictive maintenance; time-series



**Citation:** Je-Gal, H.; Park, Y.-S.; Park, S.-H.; Kim, J.-U.; Yang, J.-H.; Kim, S.; Lee, H.-S. Time-Series Explanatory Fault Prediction Framework for Marine Main Engine Using Explainable Artificial Intelligence. *J. Mar. Sci. Eng.* **2024**, *12*, 1296. <https://doi.org/10.3390/jmse12081296>

Academic Editors: Zengkai Liu

Received: 27 June 2024

Revised: 25 July 2024

Accepted: 29 July 2024

Published: 31 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Background

Predictive maintenance and fault prediction of ship engines are essential to increase the safety and economic efficiency of ship operations [1]. The 2021 Suez Canal obstruction, where a single vessel caused significant disruptions, had a substantial impact on global supply chains and the economy [2,3]. This is a crystal clear example that ship failures can cause unexpected accidents, which can lead to enormous economic losses and environmental damage. Therefore, the development of an accurate fault prediction system is significant in maximizing the efficiency of ship operation and increasing stability [4].

In particular, the importance of predictive maintenance is increasing in new generations of marine transportation such as smart ships and autonomous ships [5]. These modern ships have complex systems and configurations, so preventing breakdowns as quickly as possible is essential. Here, predictive maintenance refers to a technology that predicts failure in advance by monitoring and analyzing equipment status data in real-time. Applying the above technology can maximize equipment operation time and minimize maintenance costs by reducing unnecessary inspection and maintenance compared to the method of inspecting equipment according to a regular maintenance schedule.

In recent years, the advancement of artificial intelligence (AI) and the introduction of explainable AI (XAI) have significantly propelled the field of predictive maintenance [6–8].

AI-based predictive maintenance models can predict failures by analyzing large amounts of data, making it possible to detect ship engine problems in advance. However, existing deep learning-based AI models have the limitation that they are models that cannot explain the cause of predicted results [9]. This absence of explanation makes it difficult for users to trust the predicted results because they do not understand the underlying reasons for the predictions. Simultaneously, the explainability of the ship and the equipment design is also substantial because the ship design and the equipment design need to be the basis of the ship and the equipment design modification and expansion. To address these issues, XAI techniques have emerged, making the prediction processes and the basis behind AI model outcomes easier to understand. Predictive maintenance models using XAI not only predict failures but also explain why specific results are derived, thereby increasing the reliability of the predictions [10]. This allows users to take more appropriate actions based on the insights provided.

### 1.2. Literature Review

Predicting and detecting ship engine failure is vital to increase operational efficiency and reduce economic losses due to failure. Recently, there has been great progress in this field due to the improved performance of AI and XAI. In particular, AI technologies have been actively utilized to enhance the performance of multi-channel sensor data monitoring technology for diagnosing the condition of ship engines [11,12]. In addition to advancements in diagnostic techniques through monitoring, research on data-driven methods using multi-channel sensor data and AI models for diagnosing the condition and enhancing the energy efficiency of ships is also actively being conducted [13–16]. In related research, Velasco-Gallego et al. [17] proposed the real-time anomaly detection intelligent system framework, which combines AI technologies such as long short-term memory (LSTM) and variational autoencoder (VAE) with multi-level Otsu's thresholding, achieving high anomaly detection accuracy. Marins et al. [18] proposed a system that utilizes a random forest classifier for automatic fault detection and classification in oil wells and production/service lines. Tan et al. [19] evaluate various AI-based multi-label classification algorithms for diagnosing concurrent faults in marine machinery using single fault data to enhance fault detection without the need for extensive simultaneous fault datasets. Xu et al. [20] identified failure locations using characteristic curve methods and fault diagnosis indices. Karatuğ et al. [21] propose an innovative approach to improving ship energy efficiency by combining an engine optimization model with a data-driven adaptive neuro-fuzzy inference system to predict and control ship performance based on operational data.

Furthermore, AI technologies have been widely used not only for accurately detecting the current fault condition of ships but also for developing fault prediction models to forecast future failure.

Ji et al. [22] improved prediction accuracy by designing the convolutional neural network (CNN)-bidirectional LSTM (BiLSTM)-attention hybrid model. In addition, Han et al. [23] enhanced fault prediction accuracy by using recurrent neural network (RNN) and LSTM models with time-series data. These models reflect the time-series characteristics of the data, resulting in improved accuracy. Sun et al. [24] introduced a method to improve fault prediction accuracy by combining time-series data with support vector machines. Tong et al. [25] proposed using an optimized back propagation neural network for fault diagnosis of marine diesel engines, improving fitting and classification performance through genetic algorithm optimization. Previous studies have focused on predicting failures using multi-channel sensor data, typically employing models that consider the characteristics of multivariate time-series data.

In the context of AI models, simple traditional machine learning (ML) models, including decision trees and linear regressions, are called white-box models since they can inherently explain their prediction process. However, as AI models have become more complex for better performance, it gets more difficult to reveal the cause of the prediction. Such



models are called black-box models, and support vector machines (SVMs), random forests, and deep learning models are representative black-box models. Thus, XAI has emerged, which refers to methods that make the decision-making processes of complex AI models understandable to humans. Specifically, XAI explains the factors that lead an AI model to make a specific decision. Some early works on XAI try to explain SVMs and random forests by approximating them linearly as in the white-box models. From the perspective of predictive maintenance, Lazakis et al. [26] proposed a method combining failure modes and effects analysis (FMEA) and fault tree analysis (FTA) to identify the causes of machine failures. FMEA systematically analyzes components, functions, failure modes, and system failures, while FTA, which is a white-box model, examines how systems or components may contribute to failures. Recently, however, deep learning models have been widely used for fault prediction instead of traditional ML models whose representational capability is insufficient to represent the complex non-linear behavior of machine failures. Therefore, to explain such a deep learning-based fault prediction model, Park et al. [7] proposed an XAI methodology that utilizes sensors for anomaly detection and root cause analysis in marine engines. This study uses Shapley additive explanations (SHAP) techniques to analyze data correlations and identify important factors, thereby improving the performance of the model. However, there is a lack of XAI techniques for fault predictions focusing on time-series characteristics.

If there is little or no failure data available, it can be detrimental to creating a fault prediction model. Therefore, generating failure data is essential for training and evaluating fault prediction models. Qi et al. [27] generate failure data based on event logs from log data, which is then used to train fault prediction models.

Synthesizing prior research, most research has focused on predicting failures using multi-channel sensor data, with relatively less attention given to explaining the prediction results, especially in terms of time-series characteristics. This lack of focus on explanation can hinder the reliability of AI prediction results and pose challenges to the practical implementation of predictive maintenance systems. To resolve these issues, this study aims to apply advanced explainable AI techniques to investigate the causes of AI predictions and enhance system reliability.

### *1.3. Contribution of the Paper*

In this paper, we propose a time-series explanatory fault prediction framework that predicts a failure while simultaneously providing the rationale for the predictions. The proposed framework consists of a dimensionality reduction method, a fault prediction model, and a time-series interpretation module. In particular, it can address the key question of this study considering the time-series characteristics via the interpretation module: "Why did the fault prediction model predict a failure?". The interpretation module investigates the significant features that affect the failure prediction from a time-series point of view. The contributions of this paper are as follows:

- We propose an improved dimensionality reduction method that uses both traditional statistical and XAI techniques. The traditional dimensionality reduction techniques extract features highly correlated with the target feature to be predicted, assuming that data follows a specific parametric distribution. However, such approaches based on the parametric distribution are often likely to be inaccurate when the data does not follow the assumed distribution or contains outliers. Therefore, we combine a non-parametric XAI-based technique with the traditional statistical techniques.
- We propose a novel fault prediction procedure that can provide a time-series explainability to a fault prediction model. In the procedure, XAI techniques examine which parts of input data led to fault predictions by the fault prediction model. In particular, the proposed procedure considers both feature and time domains of data when interpreting the fault prediction model contrary to existing XAI application studies that only consider the feature domain. For each fault prediction, the proposed procedure

can analyze and visualize the region of the input data, from which the prediction is significantly influenced, in terms of both feature and time domains.

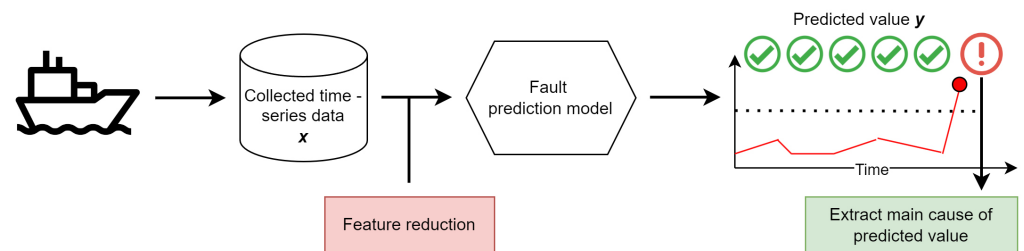
- To show the time-series explainability of the proposed framework, we construct various types of synthetic fault conditions and apply them to a real dataset for main engine maintenance. Through experiments using the dataset, we demonstrate that the proposed framework not only predicts a fault but also examines the rationale of fault predictions. For each fault prediction by the fault prediction model, the interpretation module accurately identifies and visualizes the artificial symptoms that indicate the possibility of a fault in the future.

#### 1.4. Paper Structure

This paper is organized as follows. Section 2 provides the definition of the problem considered in this paper. In Section 3, we propose a time-series explanatory fault prediction framework, and in Section 4, we provide an application scenario of the proposed framework. We provide experimental results in Section 5 and finally conclude in Section 6.

### 2. Problem Definition

We study an explanatory fault prediction framework for marine main engine failure considering the time-series characteristics of the data using XAI. Most previous fault prediction studies have focused only on accurately predicting future fault states using multi-channel sensor data or operational data. In contrast, the explanatory fault prediction framework presented in this paper not only provides predictions of future fault states, as in the previous studies but also visualizes the reasoning behind its predictions to enhance explainability with respect to a time-series perspective. The process of providing predictions and visualizing reasoning behind predictions is depicted in Figure 1. In figure, the collected time-series data represents raw data measured from the vessel. This data undergoes feature reduction and is then used for training and prediction in the fault prediction model. The trained fault prediction model continuously predicts target values, and when a predicted value falls within an abnormal range, as indicated by the exclamation mark in the figure, the main cause of the predicted value is interpreted. To this end, this framework follows three main steps: (1) effective feature reduction through data analysis, (2) training a fault prediction model, and (3) extracting the main cause of fault predictions using XAI.



**Figure 1.** The overview of a goal of the explanatory fault prediction framework. The check mark indicates that a prediction value falls within a normal range, while the exclamation mark indicates that a prediction value falls within an abnormal range.

Fault prediction is the process of predicting whether a system will fail after a certain amount of time, based on the current system information. Recently, as described in the literature review deep learning models have been widely used for fault prediction in a data-driven way. In general, a sample consists of an input time-series data  $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N\}$  and a target fault-related vector  $\mathbf{y} = (y_1, \dots, y_{d_Y})^\top$ , where the  $i$ -th input vector is defined as  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d_X})^\top$ ,  $N$  is the entire time series length of the input data, and  $d_X$  and  $d_Y$  denote the dimensions of  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. The input vector describes the current system state information while the target fault-related vector describes the fault state after a certain amount of time. The target fault-related vector can be not only categorical but also

numerical. For example, the fault-related vector can be binary, where 0 indicates that the system does not fail at the given time, and vice versa. In addition, the fault-related vector can be numerical if the fault of the system is defined based on its value (e.g., pressure and temperature). Then, a dataset for fault prediction given by the set of multiple samples as  $D = \{(x_j, y_j) | j = 1, 2, \dots, D_N\}$ , where  $j$  denotes the index of samples and  $D_N$  is the number of samples.

For a practical application scenario of fault prediction, we consider a dataset collected for monitoring the condition of a liquefied natural gas carrier (LNGC) vessel with a two-stroke diesel engine, specifically the Doosan engine 5G70ME-C9.5-GI-TII model. The data was measured during the vessel’s voyage from 1 January 2023 to 31 July 2023. Table 1 presents the principal specifications of the main engine considered in this paper, including the model type, the maximum power output, the number of cylinders, and the normal continuous rating (NCR) at 65.8 revolutions per minute (RPM).

**Table 1.** Main engine principal specifications.

Main Engine Model Type	Max Power (kW)	Number of Cylinders	NCR
Doosan engine 5G70ME-C9.5-GI-TII	12,050	5	65.8 RPM

The measured data is collected over time (i.e., it is time-series data) from 76 different sensors attached to the main engine. Since the data is collected to monitor the engine’s condition, it includes information such as the RPM of the cylinders and the coolant temperature in the main engine. In particular, we consider a failure condition of cylinders based on exhaust gas temperature. This type of failure occurs when the exhaust gas temperature within a cylinder of the main engine rises excessively. Therefore, the fault-related vector in this case becomes numerical, which indicates the exhaust gas temperature after a certain amount of time. We will describe the details of the application scenario in Section 4.

### 3. Proposed Methodology

#### 3.1. Dimensionality Reduction of Correlation Analysis and XAI

Most existing studies on multi-channel signal data analysis have used statistical analysis methods, such as Pearson correlation, to evaluate relationships between data features [28]. These statistical analysis methods are efficient for analyzing correlations between data features because they consider only a few factors, such as mean, standard deviation, and covariance, requiring less computation time. However, these methods are parametric, assuming that target data follows a specific data distribution. Therefore, the analysis results based on the methods are likely to be inaccurate if the analyzed data do not follow the specific distribution or the correlation between variables is non-linear, due to their theoretical background [29].

Data analysis using XAI, on the other hand, employs non-parametric methodologies that do not assume any specific data distribution during correlation analysis. Therefore, XAI can provide more accurate correlation analyses regardless of the data distribution or non-linear relationships between features. However, a limitation of XAI-based analysis is the variability in analysis results depending on the extent of training of the AI model, which leads to potential instability in the analysis results.

We here propose a complementary correlation analysis method that combines statistical correlation analysis with XAI-based analysis. This approach aims to complement the limitations of each individual method by leveraging their respective strengths.

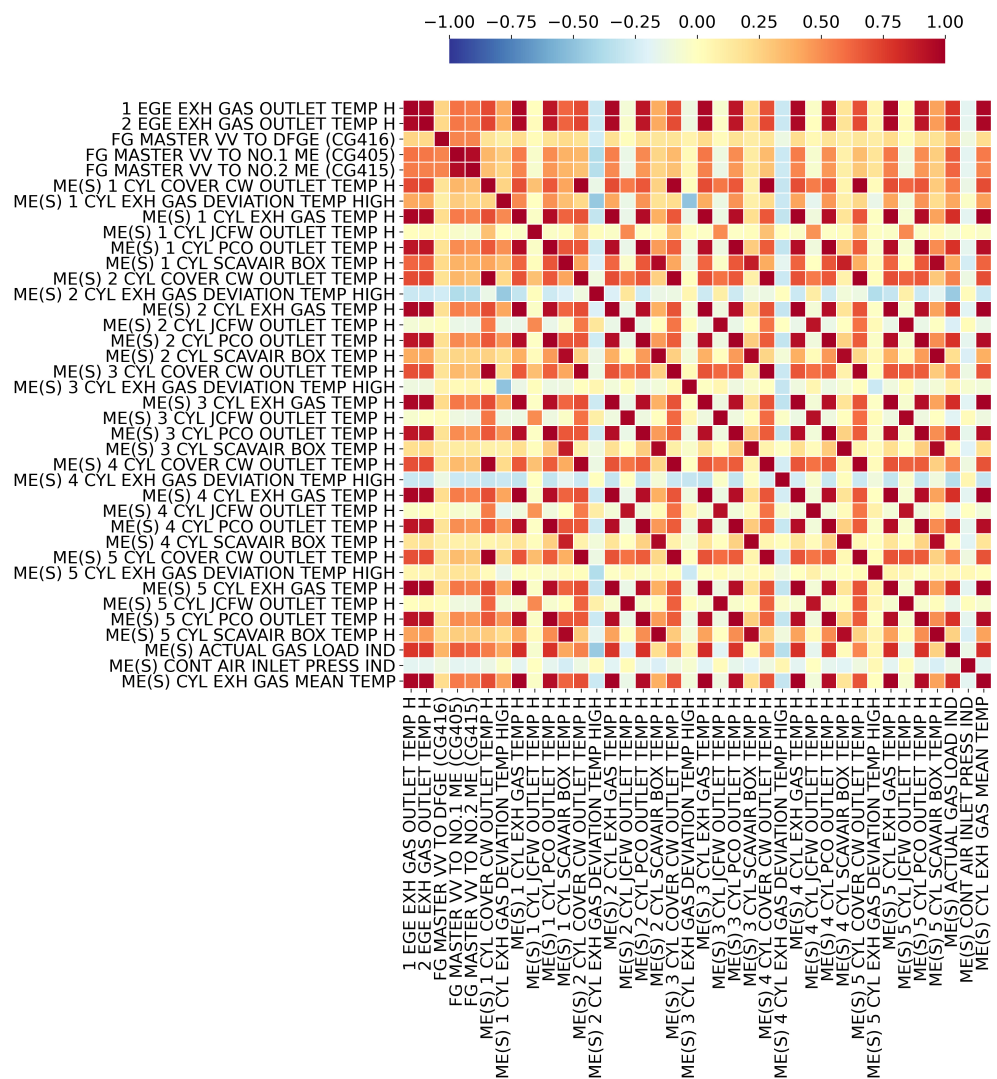
##### 3.1.1. Pearson Correlation Analysis

We first employ the Pearson correlation analysis as an initial analysis method to investigate the linear correlation among features in the dataset. Through the Pearson correlation analysis, the Pearson correlation coefficients over the features are calculated, which indicates the correlation between features. Specifically, each Pearson correlation

coefficient is obtained by statistically measuring the linear relationship between features. The equation for computing the coefficient between features  $x_k$  and  $x_l$  is given as follows.

$$P_{x_k, x_l} = \frac{\sum_{i=1}^N (x_{i,k} - \bar{x}_k)(x_{i,l} - \bar{x}_l)}{\sqrt{\sum_{i=1}^N (x_{i,k} - \bar{x}_k)^2} \sqrt{\sum_{i=1}^N (x_{i,l} - \bar{x}_l)^2}}, \tag{1}$$

where  $x_{i,j}$  denotes the value of feature  $j$  in sample  $i$  and  $\bar{x}_j$  is the mean value of feature  $x_j$  over the samples (i.e.,  $\bar{x}_j = N^{-1} \sum_{i=1}^N x_{i,j}$ ). The computed value of  $P_{x_k, x_l}$  signifies the correlation coefficient between features  $x_k$  and  $x_l$ , ranging from  $-1$  to  $+1$ . A value closer to  $+1$  indicates a strong proportional relationship between features  $x_k$  and  $x_l$ , while a value nearer to  $-1$  indicates a strong inverse proportional relationship. Moreover, a value closer to  $0$  suggests a weaker correlation between features  $x_k$  and  $x_l$ . In Figure 2, the computed correlations among 38 features, half of 76 features in the dataset that will be used in the practical application are presented.



**Figure 2.** The example of the Pearson correlation analysis of half of the entire feature results.

### 3.1.2. Shapley Additive Explanations Analysis

In the previous subsection, the Pearson correlation analysis method is used as the initial analysis for data correlation analysis. The simplicity, intuitiveness, and low computational cost of the method allow us to conveniently identify the overall correlation trends in the data. However, in general, real-world data often exhibit non-linear relationships between

features and are likely to contain outliers that hinder the correlation analysis. In such cases, the Pearson correlation analysis, which investigates linear relationships between features, may derive inaccurate correlation analysis results from the data [30].

Here, we aim to address the limitations of the Pearson correlation analysis by using the SHAP technique, a method within the domain of XAI. Due to its non-parametric nature, the SHAP technique is capable of analyzing datasets more accurately compared to the Pearson correlation analysis, even when the data contains non-linear relationships and outlier samples [31]. Suppose that the SHAP technique is used to analyze the correlation between a target feature and the others. To this end, first, a prediction model is trained that predicts the value of the target feature based on the given values of the others. The trained model inherently approximates the relationship between the target feature and the other input features. Then, by applying the SHAP technique to the model, we can quantify the impact of each input feature on the prediction for the target feature based on Shapley values from game theory. This approach assumes only that each input feature in the dataset is independent. The influence of an input feature on the target feature can be calculated using the following equation:

$$\text{Shapley value of feature } k = \sum_{\mathcal{S} \subseteq \mathcal{N}_X \setminus \{k\}} \frac{|\mathcal{S}|(d_X - |\mathcal{S}| - 1)!}{d_X!} (v(\mathcal{S} \cup \{k\}) - v(\mathcal{S})), \quad (2)$$

where  $\mathcal{N}_X = \{1, 2, \dots, d_X\}$  denotes the set of the features of the input vector,  $|\mathcal{X}|$  denotes the cardinality of a set  $\mathcal{X}$ ,  $\mathcal{S}$  represents all subsets of  $\mathcal{N}_X$  not containing feature  $k$ , and  $v(\mathcal{S})$  denotes the contribution to the prediction by the input features in set  $\mathcal{S}$ . This equation allows us to decompose the model's prediction into the contribution of each input feature, making it possible to analyze the influence of each input feature on the target feature's prediction.

The Shapley value in (2) considers all features based on the predictions. Therefore, even if the relationships between the features are non-linear, the SHAP technique can evaluate the correlations between the target feature and the others by analyzing their complex interactions through the prediction model. At the same time, however, this makes the effectiveness of the SHAP technique for correlation analysis highly dependent on the representational performance of the prediction model. If the prediction model represents the complex non-linear relationships between features well, it can also evaluate the correlations well. On the other hand, if the model is poorly trained, it may produce inaccurate analysis results. To overcome this risk, we jointly use Pearson correlation-based data correlation analysis, which can analyze the overall linear relationships in the data, thereby reducing potential errors.

In summary, we propose a feature reduction method that aims to improve feature reduction by combining the results of the SHAP technique and Pearson correlation analysis. To identify only the features that are clearly related, it selects the features that demonstrate consistently high proportional or inversely proportional relationships with similar magnitudes in both analyses. This approach leverages the complementary strengths of these two techniques to evaluate non-linear correlations between features while minimizing error.

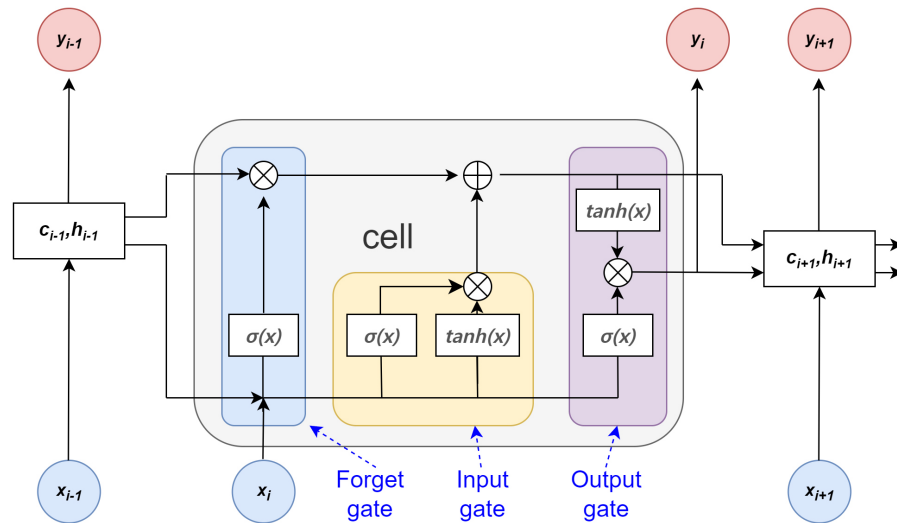
### 3.2. Fault Prediction Model Based on LSTM

A variety of deep learning-based prediction models can be used to perform fault prediction. In particular, for data measured over time, RNN performs well in predicting by reflecting the time-series characteristics of the data [32,33]. The goal of the fault prediction framework in this paper is to predict a fault that could possibly occur in the future considering the time-series characteristics of the measured sensor data. Therefore, we use a deep learning model based on LSTM, which is an enhanced version of traditional RNN models [34].

The LSTM structure is depicted in Figure 3. The sequence of data (i.e.,  $\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \dots$ ) is used as input. LSTM consists of three gates: the input gate, the output gate, and the forget gate, each of which operates differently. As in traditional RNN, LSTM recursively uses



its output as an input for the next data. However, it can address the vanishing gradient problem by using these three gates, which is the major significant difference between traditional RNN and the core functionality of LSTM.



**Figure 3.** The architecture of an LSTM cell.

First, the input gate determines how much of the current input information to retain and integrates this information into the existing memory based on input gate activation and candidate cell state. Second, the forget gate decides how much of the previous memory to retain at the current time step through forget gate activation, which influences the amount of information the LSTM remembers. Lastly, the output gate makes its memory of the short time step by output gate activation. These processes of the LSTM can be expressed as the following equations:

$$f_i = \sigma(W_{xf}x_i + W_{hf}h_{i-1} + b_f), \tag{3}$$

$$p_i = \sigma(W_{xp}x_i + W_{hp}h_{i-1} + b_p), \tag{4}$$

$$g_i = \tanh(W_{xg}x_i + W_{hg}h_{i-1} + b_g), \tag{5}$$

$$o_i = \sigma(W_{xo}x_i + W_{ho}h_{i-1} + b_o), \tag{6}$$

$$c_i = f_i \odot c_{i-1} + p_i \odot g_i, \tag{7}$$

$$\text{and } h_i = \tanh(c_i) \odot o_i. \tag{8}$$

In these equations, the index of a sequence of data  $i$  can be interpreted as the time step in time-series data. Accordingly,  $x_i$ ,  $p_i$ ,  $g_i$ ,  $f_i$ ,  $o_i$ ,  $c_i$ , and  $h_i$  represent the input, input gate activation, candidate cell state, forget gate activation, output gate activation, memory cell, and hidden state at time step  $i$ , respectively.  $W_{xp}$ ,  $W_{xg}$ ,  $W_{xf}$ ,  $W_{xo}$  are the weights multiplied by  $x_i$  at each gate, and  $W_{hp}$ ,  $W_{hg}$ ,  $W_{hf}$ ,  $W_{ho}$  are the weights multiplied by  $h_i$  at each gate. The symbol  $\odot$  indicates element-wise multiplication between two vectors, while  $\sigma(\cdot)$  and  $\tanh(\cdot)$  are the sigmoid and tangent hyperbolic activation functions, respectively.

The memory of LSTM is composed of the long-term state,  $c_i$ , which remembers long-term sequences, and the short-term state  $h_i$ , which remembers short sequences. They are

updated at each time step based on the input  $x_i$  and the three gates, enabling the learning of temporal sequences. Then, the computation processes of each gate will be explained individually, followed by how these results are utilized to compute  $c_i$  and  $h_i$ . First, the forget gate computes the forget gate activation,  $f_i$ , using Equation (3). The forget gate activation determines how much of the memory cell from the previous time step,  $c_{i-1}$ , should be retained at the current time step, based on the current input,  $x_i$ , and the previous hidden state,  $h_{i-1}$ . Second, the input gate computes the input gate activation,  $p_i$ , and the candidate cell state,  $g_i$ , using Equations (4) and (5), respectively. The input gate activation integrates  $x_i$  and  $h_{i-1}$  to determine the proportion of  $x_i$  to be reflected in the memory cell,  $c_i$ , while the candidate cell state represents the current input information to be added to  $c_i$ . Lastly, the output gate computes the output gate activation,  $o_i$ , using Equation (6). It multiplies  $x_i$  and  $h_{i-1}$  by the weights  $W_{x0}$  and  $W_{h0}$ , respectively, and applies the  $\sigma$  function to determine how much of  $c_i$  to reflect in  $h_i$ .

In summary, the forget gate, input gate, and output gate compute their activation values  $p_i$ ,  $f_i$ , and  $o_i$ , respectively, using the  $\sigma$  function. These activation values determine how much information is reflected in the long-term state,  $c_i$ , and the short-term state,  $h_i$ . Exceptionally, the input gate computes not only its activation value,  $p_i$ , but also the candidate cell state,  $g_i$ , that contains the information from  $x_i$  to be reflected in  $c_i$ . Using the results of Equations (3)–(7) adjusts the previous memory cell  $c_{i-1}$  by  $f_i$  and combines it with  $g_i$  scaled by  $p_i$  to compute the long-term state  $c_i$ . Similarly, Equation (8) adjusts the memory cell  $c_i$  by passing it through the tanh activation function and scaling the activated value by  $o_i$  to compute the short-term state  $h_i$ . Through these processes, the LSTM learns time-series characteristics of data that can make accurate predictions in a time-series data environment. The data  $x_i$  represents sensor data with multiple features at each time step. The LSTM is designed to predict the target fault-related values at future time steps by considering the previous data and the order of each feature in the time series, as described above.

### 3.3. Perturbation-Based XAI Method Considering Temporal Dependencies

Recently, deep learning models have been widely used in various predictive maintenance domains. They enable the predictive models to achieve better performance, but cannot explain why such predictions are made. Thus, they are often referred to as black-box models. To overcome this limitation, we adapt XAI techniques to provide the rationale for the predictions made by such black-box models. In particular, the XAI technique for the fault prediction of the main engine should be able to consider the time-series characteristics of the measured data since the data for the fault prediction is a time-series data collected in a time-sequential order.

In the XAI literature, a method that interprets black-box models by utilizing the changes in the output according to the input that is slightly perturbed is called perturbation-based XAI. In this paper, we employ a perturbation-based XAI method, especially considering the time-series characteristics of the features. The flowchart of the method to interpret the prediction of a target black-box model for a target input is shown in Figure 4. We briefly explain the perturbation-based interpretation procedure step by step in the following: In each iteration, first, the target input of the target black-box model is perturbed by using a perturbation operator that reflects the importance of each data element described by a mask. The black-box model predicts the output (i.e.,  $Output_{Mask}$  in the figure) based on the perturbed input. Then, the mask is updated in a direction that reduces the difference between the predicted outputs based on the original input (i.e.,  $Output$  in the figure and the perturbed input (i.e.,  $Output_{Mask}$  in the figure). In the next iteration, the perturbation operator perturbs the input differently according to the updated mask. This procedure is repeated to train the mask, and described in mathematical expressions as follows:

$$M_a^* = \arg \min_{M \in [0,1]^{N \times d_x}} \lambda_e \cdot \mathcal{L}_e(M) + \lambda_a \cdot \mathcal{L}_a(M) + \lambda_c \cdot \mathcal{L}_c(M), \tag{9}$$

where  $M$  denotes a mask that describes the importance of input data elements,  $\mathcal{L}_e$ ,  $\mathcal{L}_a$ , and  $\mathcal{L}_c$  are the loss functions, and  $\lambda_e$ ,  $\lambda_a$ , and  $\lambda_c$  are hyperparameters that determining the weighting of each loss term.

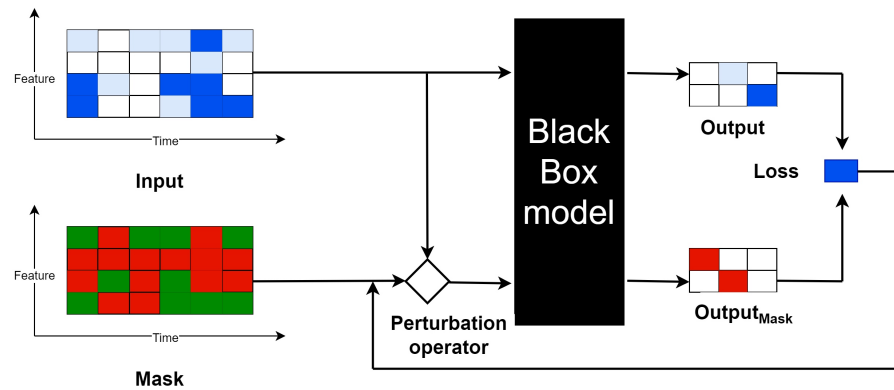


Figure 4. The flowchart of perturbation-based interpretation.

Equation (9) represents an optimization problem for finding the optimal mask  $M_a^*$  that minimizes the given objective function. The objective function is designed so that the optimal mask describes the importance of input data elements within the range of 0 and 1 while satisfying the area constraint on the ratio of the area indicated as important. In the objective function,  $\mathcal{L}_e$  measures the error of  $M$  during training,  $\mathcal{L}_a$  restricts the size of the highlighted area, and  $\mathcal{L}_c$  limits the degree of smoothing of  $M$ . First, the area regularization term  $\mathcal{L}_a$  is defined as

$$\mathcal{L}_a(M) = \|\text{vecsort}(M) - r_a\|^2, \tag{10}$$

where  $\text{vecsort}(\cdot)$  is a function that transforms a matrix into a vector and then sorts it in ascending order and  $r_a$  is a vector composed of  $(1 - a) \cdot d_X \cdot N$  zeros and  $a \cdot d_X \cdot N$  ones. In the equation,  $M$  is sorted in ascending order, and then,  $r_a$  is subtracted to adjust the area size. Therefore, if the size of the region highlighted by  $M$  deviates significantly from  $a$ , the value of  $\mathcal{L}_a(M)$  will increase. Since the objective function aims at minimizing  $\mathcal{L}_a(M)$ , the training process adjusts  $M$  to ensure that the size of the highlighted area is close to  $a$ . Consequently, this ensures that  $M$  highlights only up to the size specified by  $a$ .

The second loss term, the time smoothing term, is defined by

$$\mathcal{L}_c(M) = \sum_{i=1}^{N-1} \sum_{k=1}^{d_X} |m_{i+1,k} - m_{i,k}|, \tag{11}$$

where  $m_{i,k}$  denotes elements of  $M = (m_{i,k}) \in [0, 1]^{N \times d_X}$ . This term ensures that  $M$  is smoothly connected over time. It measures the temporal variation of the same feature  $k$  through the difference between  $m_{i+1,k}$  and  $m_{i,k}$ . During training, the model minimizes this difference, leading to a mask that is smoothly connected along the time axis. In addition, the importance of  $x_{i,k}$  is learned through  $m_{i,k}$  and  $m_{i,k}$  interacts with  $m_{i+1,k}$  in the temporal dimension. This implies that this term makes the mask reflect the temporal dependencies of data during interpretation.

The third loss term, the error loss term, is defined by

$$\mathcal{L}_e(M) = \sum_{i=i_y}^N \sum_{k=1}^{d_Y} ((V \circ \Pi_M)(\mathbf{x}))_{i,k} - [V(\mathbf{x}))_{i,k}]^2, \tag{12}$$

where  $V$  is a black-box regression model,  $\Pi_M$  is a perturbation operator, and the symbol  $\circ$  denotes the function composition operator. This term measures the difference between output of  $(V \circ \Pi_M)(\mathbf{x})$  and output of  $V(\mathbf{x})$ . Therefore, it represents a loss defined to measure the difference between the output of  $\Pi_M(\mathbf{x})$  passed through the per-

turbation operator specified by  $M$  and the output of  $\mathbf{x}$  without passing through the perturbation operator.

Through this loss term, the objective function trains  $M$  such that  $(V \circ \Pi_M)(\mathbf{x})$  exhibits minimal difference from the original output of  $\mathbf{x}$ . The perturbation operator  $\Pi$  can be defined in various ways. Here, we provide one of the representative operators based on the time average as follows:

$$\Pi(\mathbf{x}, m_{i,k}; i, k) = m_{i,k} \cdot x_{i,k} + (1 - m_{i,k}) \cdot \mu_{i,k}, \tag{13}$$

where  $\mu_{i,k}$  is the average value of  $x_{i-W,k}$  to  $x_{i+W,k}$  (i.e.,  $\mu_{i,k} = \frac{1}{2W+1} \sum_{i'=i-W}^{i+W} x_{i',k}$ ) and  $W$  is a hyperparameter that determines the distance of adjacent time points considered by the perturbation operator during perturbation. In the operator, each element of  $\mathbf{x}$  is transformed considering its adjacent elements via  $\mu$ . When  $\Pi$  transforms  $\mathbf{x}$ ,  $m_{i,k}$  is closer to 1, the more the original value  $x_{i,k}$  is preserved, and  $m_{i,k}$  is closer to 0, the more  $\mathbf{x}$  is perturbed by reflecting the adjacent average value  $\mu_{i,k}$  instead of the original  $x_{i,k}$ . This process is the part that allows the XAI method to consider time-series characteristics and temporal dependencies of data during interpretation.

In summary, Equation (9) generates the mask  $M$  that indicates the area significant for the prediction by considering the three loss terms defined in Equations (10)–(12), which obscure all areas without highlighting with minimal impact on the input-output difference and ensure smooth connections along the time axis. In this paper, we build a fault prediction model considering the time-series characteristics as described in Section 3.2. Therefore, when interpreting this fault prediction model, XAI techniques that incorporate time-series characteristics should be used to achieve more accurate interpretations.

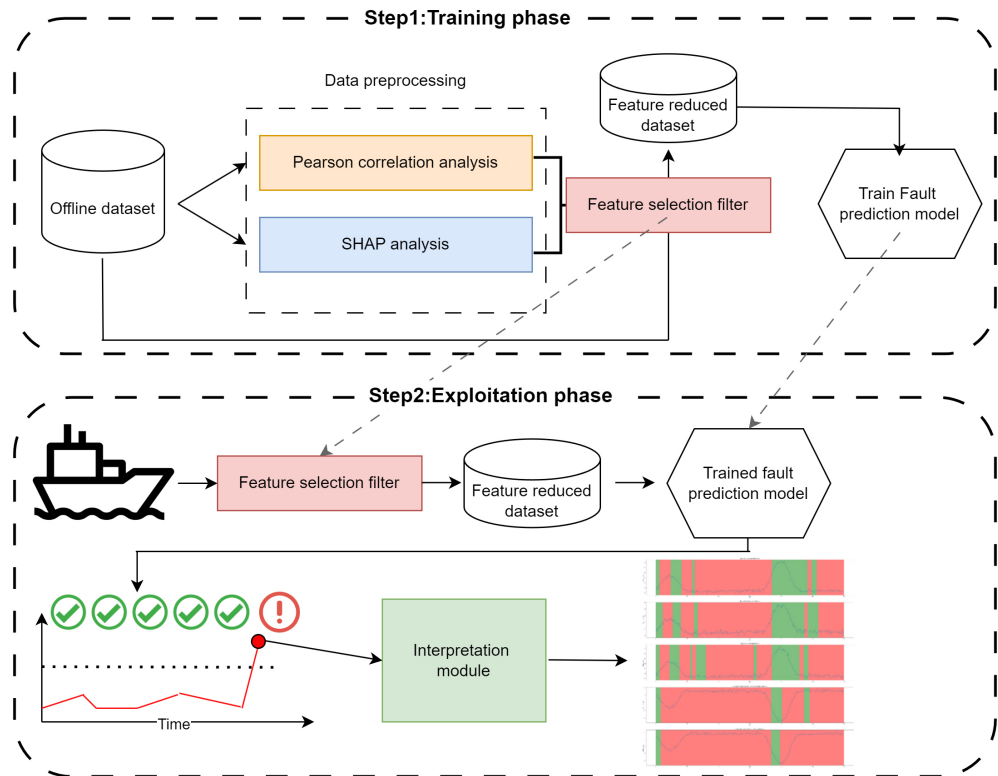
### 3.4. Proposed Time-Series Explanatory Fault Prediction Framework

We now propose a time-series explanatory fault prediction framework based on the above ingredients. It consists of two phases, a training phase and an exploitation phase. In the training phase, a fault prediction model is trained using the preprocessed dataset. In the exploitation phase, the prediction model trained in the training phase is used to predict possible faults in the future. If a fault is predicted during the exploitation phase, the temporal explanation of the predicted fault is provided via the interpretation module in the framework. We will describe the details of the phases below.

The flowchart of the entire training phase is illustrated in the upper panel of Figure 5. In the training phase, a fault prediction model is built based on LSTM as described in Section 3.2, and trained using an offline dataset before using the model in practice. In general, training the prediction model using the raw data may not only degrade the accuracy of the model due to unnecessary data inclusion but also significantly increase the training time due to the high dimensionality of the data. Therefore, the dataset should be preprocessed before being used to train the prediction model. To this end, first, two data analysis methods described in Section 3.1 are performed to extract the features that are highly correlated with the target feature. Then, a feature selection filter is constructed based on the correlation analyses. Using the filter, the dataset is processed into a feature-reduced dataset which has a smaller dimensionality compared to the original one. Then, the training samples that consist of time-series input features and a future fault occurrence are generated from the feature-reduced dataset. To predict the future fault occurrence based on the time-series input features, the prediction model is trained by using the generated training samples.

The flowchart of the exploitation phase is illustrated in the lower panel of Figure 5. Contrary to the training phase, the exploitation phase describes a fault prediction operation in practice where real-time data is collected continuously for fault prediction. To this end, the collected time-series data should be processed into feature-reduced data, since the fault prediction models were trained by using the feature-reduced dataset. The feature selection filter constructed during the training phase is used for this feature reduction. Then, the feature-reduced version of the collected time-series data is fed into the model to predict

future fault occurrence. In practice, the time-series data is collected continuously, and thus, the prediction model receives the continuous data and continuously predicts the future fault occurrence. Once the future fault occurrence is predicted in this continuous prediction process, the interpretation module derives the rationale of the fault prediction through the explanation analysis as described in Section 3.3. Specifically, the interpretation module indicates the temporal regions of the data that were significant in the model’s prediction of future failure.



**Figure 5.** The flowchart of the training phase and exploitation phase. In the result of the interpretation module, the green region of the input features represents the parts that were significant in the model’s prediction.

The main difference between the training phase and the exploitation phase is whether the fault prediction model is trained or exploited. In the training phase, the model is trained offline. On the other hand, in the exploitation phase, the trained model is implemented on the ship and used online during the ship’s operation. It is worth noting that the training phase can be performed periodically to reflect the data additionally collected during the exploitation phase.

The proposed time-series explanatory fault prediction framework can be applied to different application domains. Then, various data analysis techniques can be used to generate a feature selection filter, depending on the data characteristics of the application domain. In addition, the conditions of fault occurrence and prediction interpretation can also be varied according to the application domain. For example, the condition to perform fault prediction interpretation may be configured as more sensitive compared with the condition of fault occurrence for more conservative and robust ship management.

#### 4. Application of Explanatory Fault Prediction in Real-World Datasets

##### 4.1. Scenario of Fault Prediction

We consider a failure condition of excessive cylinder exhaust gas temperature rises. This type of failure occurs when the exhaust gas temperature within a cylinder of the main engine rises above 600 °C. In general, the exhaust gas temperature ranges between 250 °C



and 395 °C during normal operation. However, when problems occur within the engine, such as problems with the fuel injection, air intake, or cooling systems, the temperature rises, indicating a fault.

To monitor this failure condition, each cylinder within the main engine is equipped with sensors labeled 'ME(S) 1 CYL EXH GAS TEMP H', 'ME(S) 2 CYL EXH GAS TEMP H', 'ME(S) 3 CYL EXH GAS TEMP H', 'ME(S) 4 CYL EXH GAS TEMP H', and 'ME(S) 5 CYL EXH GAS TEMP H'. For clear presentation, we aim to predict the failure condition in the first cylinder by predicting the sensor values of 'ME(S) 1 CYL EXH GAS TEMP H'. Specifically, a fault prediction model in this scenario predicts the exhaust gas temperature of the first cylinder after a certain amount of time. It is worth noting that the proposed framework in this paper can be applied not only to the failure of the cylinder gas temperature but also to any type of failure.

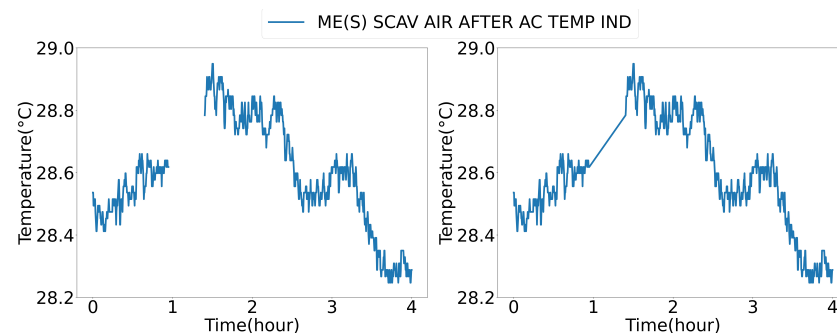
#### 4.2. Data Generation

The dataset used in this study was measured from sensor data for monitoring the condition of the main engine during the voyage of an LNGC vessel with each sample consisting of data collected at one-minute intervals over approximately seven months. As described in Section 2, it consists of 76 sensor features for ship state monitoring. For the prediction of the target condition, the target fault-related value is set as the value of the target feature, 'ME(S) 1 CYL EXH GAS TEMP H', at a certain amount of time. To show and evaluate the time-series explanation of fault prediction, we artificially simulated failure conditions by adding synthetic failure behaviors that consist of fault situations and pre-symptoms. A pre-symptom is various phenomena that occur before a fault arises, thereby indicating the possibility of the fault in the future. Therefore, the predictive interpretability for the fault prediction can be evaluated by how well the corresponding pre-symptoms are identified. These synthetic failure conditions allow us to evaluate the prediction performance and the predictive interpretability more clearly, since the reasoning behind failures in practical ships may not be clear.

Furthermore, in practice, it is difficult to collect ship operation data that includes engine failures. This is because the management agents of a ship such as companies and individuals, regularly inspect the ship's engines to prevent their failures, which incur the significant cost.

##### 4.2.1. Data Preprocessing

Typically, the dataset collected from real-world environments should be processed before being utilized for deep learning models. Here, a data preprocessing step is implemented to transform the data into a format suitable for training predictive models. Firstly, missing values within the collected data should be addressed. The data spans a relatively long measurement period. Hence, in the data, some sensor measurements within the measurement period may be missing due to sensor errors or communication errors between the ship and the marine plant. In this study, we address such missing values by linear interpolation. We provide the sample result of the missing values before and after linear interpolation in Figure 6.



**Figure 6.** The sample result of before-after linear interpolation.

In addition to the missing values, there are segments in the dataset where all sensor values abruptly drop simultaneously. These segments were observed to coincide with periods when the main engine’s RPM falls to zero, indicating a stop in operation (i.e., at anchor). Since we are considering fault prediction during operation, we removed the data during these periods as shown in Figure 7.

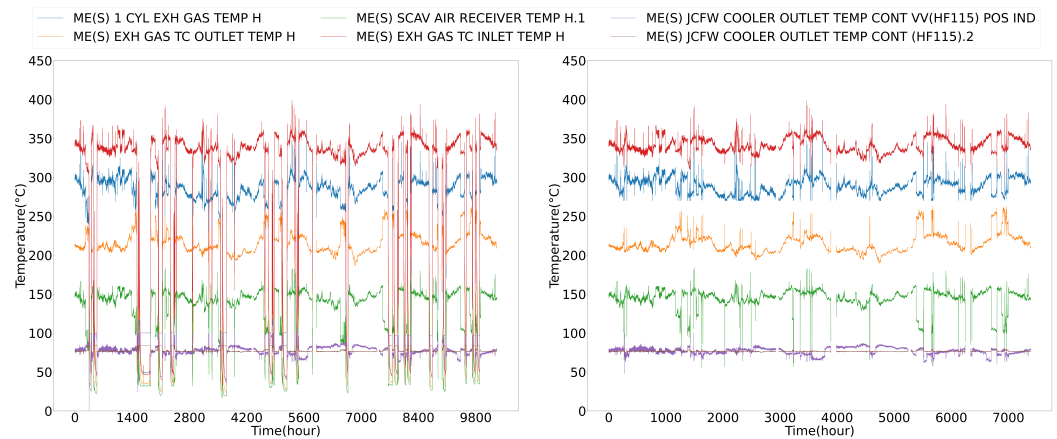


Figure 7. The sample result of the before-after removal of downtime periods.

#### 4.2.2. Synthetic Fault Behavior for Experiments

As described earlier, we artificially added failure conditions and their pre-symptoms to the data to generate training and test datasets for fault prediction, since the measured data does not include any fault data. To simulate various types of failure conditions, we generated pre-symptoms and fault appearances in diverse forms, as illustrated in Table 2. The fault behavior includes the excessive exhaust gas temperature rise above 600 °C and its pre-symptoms.

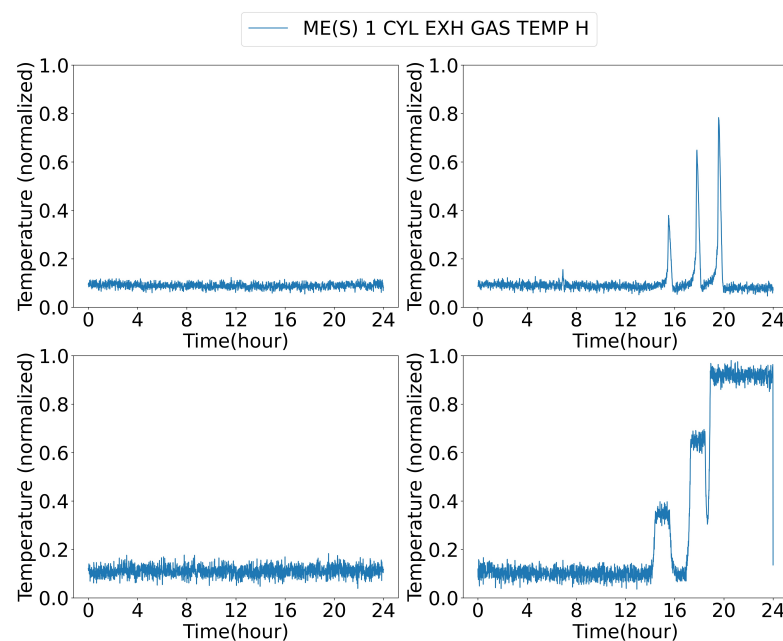
Table 2. Different failure behaviors.

Dataset	Gaussian Noise	Increasing Pattern	Combination of Pre-Symptoms	Regularity
Data 1	×	Vertical	×	✓
Data 2	✓	Vertical	×	✓
Data 3	×	Linear	×	✓
Data 4	✓	Linear	×	✓
Data 5	×	Exponential	×	✓
Data 6	✓	Exponential	×	✓
Data 7	×	Exponential	×	✓
Data 8	×	Exponential	×	×
Data 9	✓	Exponential	×	×
Data 10	×	Exponential	✓	×
Data 11	✓	Exponential	✓	×

In Table 2, there are four factors that contribute to constructing different fault behaviors with different pre-symptoms. The × symbol in the table indicates that the factor is not reflected, whereas the ✓ symbol indicates that the factor is reflected. First, to simulate scenarios where more noise is added during data measurement, Gaussian noise is applied to both target data and the entire input data. Second, when generating failure conditions and their pre-symptoms, we consider three different types of increasing patterns that represent how the temperature rises in the time domain. They describe different situations where the temperature inside the cylinder increases rapidly or gradually. Third, to account for cases where pre-symptoms appear at regular intervals and those where they do not, a condition of the combination of pre-symptoms was considered. Lastly, the factor of regularity was

included to consider scenarios where pre-symptoms occur but the fault condition does not occur. Considering the factors described above, we generate fault condition data by specifying a sequence where two pre-symptoms are followed by one fault condition for 'ME(S) 1 CYL EXH GAS TEMP H'. The pre-symptoms fall outside the normal range of 250 °C to 395 °C but do not exceed the fault criterion of 600 °C. After the pre-symptoms appear, the fault condition that exceeds 600 °C occurs. In particular, the pre-symptoms for each feature are generated according to its correlation with the target feature: For features with a positive correlation, the increasing pattern is added, while for features with a negative correlation, the decreasing pattern is added. The intervals and magnitudes of occurrence were randomly assigned. This data generation method can cover different failure patterns that may occur together with the four factors.

As a result, we augment the normal dataset by using 11 types of fault behavior patterns, generating multiple scenarios where each dataset represents a set of scenarios with and without faults. Each dataset consists of half normal measurements and half abnormal measurements with faults. The visualizations of Abnormal data 6 and Abnormal data 11 are shown in Figure 8.



**Figure 8.** The sample of normal scenarios and abnormal scenarios with faults in Abnormal data 6 (upper) and Abnormal data 11 (lower).

### 4.3. Training Phase

#### 4.3.1. Feature Selection Filter for Dimensionality Reduction

For training the fault prediction model, we partitioned each dataset in Table 2 into three subsets. We use 60% of measurements as a training dataset for the fault prediction model, 10% as a validation dataset to check for overfitting and to assess the model's performance during training, and 30% as a test dataset to evaluate the performance of the trained model. It is worth noting that any techniques related to model training and hyperparameter optimization can be used for training fault prediction models. These training datasets correspond to the offline dataset of the training phase depicted in Figure 5. Consequently, as described in Section 3.4, both Pearson correlation and SHAP analysis are utilized to generate a feature selection filter. We consider the failure of excessive cylinder exhaust gas temperature rises, which is identified based on the temperature measured by 'ME(S) 1 CYL EXH GAS TEMP H'. Therefore, the fault prediction model is built to predict the future sensor values of 'ME(S) 1 CYL EXH GAS TEMP H'. To analyze the correlation between

'ME(S) 1 CYL EXH GAS TEMP H' and all other features, we apply the Pearson correlation analysis to that feature as shown in Figure 9, instead of applying it to all features as in Figure 2.

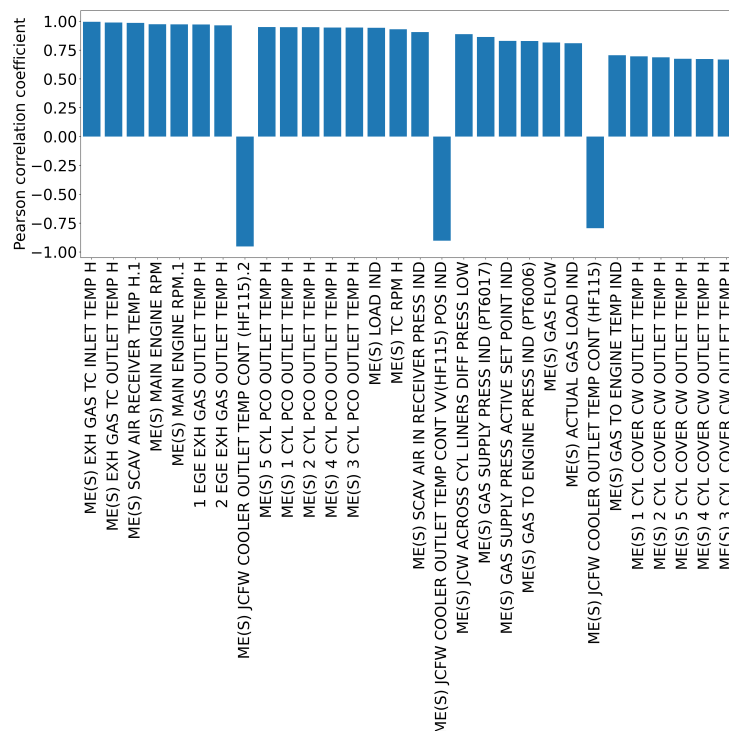


Figure 9. Pearson correlation analysis result of 'ME(S) 1 CYL EXH GAS TEMP H' (30 features with the largest absolute magnitudes).

Unlike Pearson correlation analysis, SHAP analysis is an XAI technique, requiring two steps: (1) training a predictive model for the target feature and (2) the interpretation of this model using SHAP. To this end, we build a simple DNN to predict the current 'ME(S) 1 CYL EXH GAS TEMP H' for the given current system state information. Then, the trained DNN model is analyzed using SHAP. The parameters of the utilized DNN model are presented in Table 3.

Table 3. DNN architecture.

Layer	Design Parameters
Input layer	76 nodes
1st hidden layer	64 nodes/ReLU
2nd hidden layer	32 nodes/ReLU
3rd hidden layer	16 nodes/ReLU
4th hidden layer	8 nodes/ReLU
Output layer	1 nodes

In Figure 10, the SHAP analysis results are visualized. The graph on the left displays the top 20 features, ranked in descending order of influence on the prediction of the DNN model. In this graph, samples with red, i.e., positive Shapley values, indicate a proportional relationship between the feature value and the target value (i.e., 'ME(S) 1 CYL EXH GAS TEMP H'), meaning that as the feature value increases, the target value also increases, with higher values indicating a stronger proportional relationship. Conversely, samples represented in blue, i.e., negative Shapley values, indicate an inversely proportional relationship with the target value. The larger absolute Shapley values indicate the stronger (inverse) proportional relationship. The graph on the right represents the sum of the

absolute Shapley values for each feature, providing insight into the overall importance of each feature in predicting the target value, regardless of whether the relationship is proportional or inversely proportional.

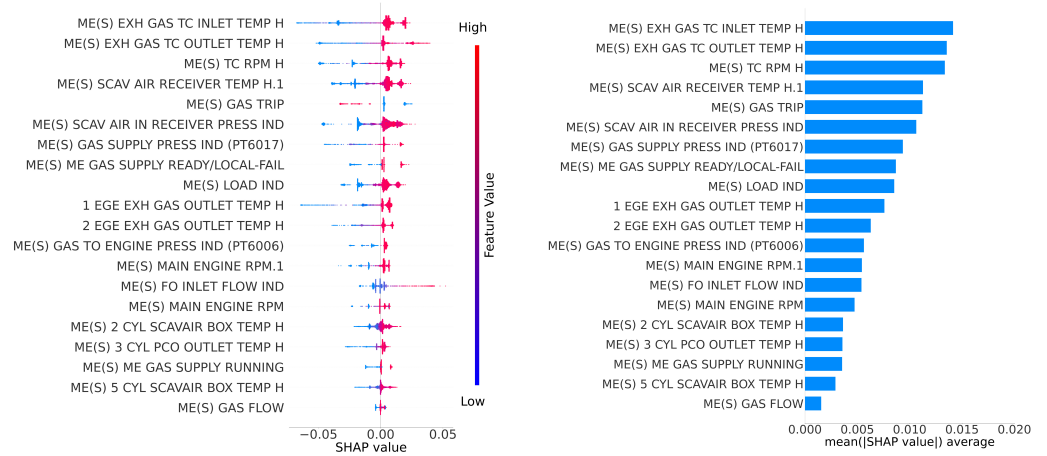


Figure 10. The SHAP analysis result of ‘ME(S) 1 CYL EXH GAS TEMP H’.

A feature selection filter can be generated based on the Pearson correlation analysis and SHAP analysis illustrated in Figures 9 and 10, respectively. For example, the significance of each feature is evaluated as a weighted sum of its absolute SHAP value and its Pearson correlation coefficient. Then, the weighted sums for the features are sorted in descending order to select the top  $n$  features. This method generates a feature selection filter that has a significant relationship with the target feature (‘ME(S) 1 CYL EXH GAS TEMP H’ in this application). In this application scenario, we aim to verify how the impact of each feature extracted through XAI techniques could vary in prediction factor interpretation. To ensure data diversity in the synthetic fault data, we selected the following five features with the most significant relationship, but two with the negative correlation: ‘ME(S) EXH GAS TC OUTLET TEMP H’, ‘ME(S) SCAV AIR RECEIVER TEMP H.1’, ‘ME(S) EXH GAS TC INLET TEMP H’, ‘ME(S) JCFW COOLER OUTLET TEMP CONT VV(HF115) POS IND’, and ‘ME(S) JCFW COOLER OUTLET TEMP CONT (HF115).2’. It is worth noting that there is no gold standard for selecting features and any criterion can be used according to the given situation. The statistical characteristics of these selected features and the target feature ‘ME(S) 1 CYL EXH GAS TEMP H’ are shown in Table 4.

Table 4. Statistical characteristics of the target and extracted features.

Features	Min (°C)	Max (°C)	Mean (°C)	Median (°C)	Std (°C)
ME(S) 1 CYL EXH GAS TEMP H	45.48	331.42	274.41	294.25	60.05
ME(S) EXH GAS TC OUTLET TEMP H	31.30	255.87	202.45	210.47	42.30
ME(S) SCAV AIR RECEIVER TEMP H.1	22.20	166.89	131.19	144.73	34.63
ME(S) EXH GAS TC INLET TEMP H	44.42	373.19	317.82	336.84	69.32
ME(S) JCFW COOLER OUTLET TEMP CONT VV(HF115) POS IND	0.59	100.07	80.17	79.00	6.01
ME(S) JCFW COOLER OUTLET TEMP CONT (HF115).2	70.65	88.16	77.39	76.68	2.07

#### 4.3.2. Training Fault Prediction Model with LSTM

For predicting sensor measurements, a variety of deep learning models can be used. However, we focus on predicting future sensor measurements based on time-series system state information as illustrated in Figure 11. Therefore, we use an LSTM model for its simplicity in handling time-series data for future predictions. Specifically, in this application scenario, a single future sensor measurement, ME(S) 1 CYL EXH GAS TEMP H, is predicted



using the sequence of measurements from the past. To address this fault prediction, we consider a many-to-one sensor value prediction model based on an LSTM model with the structure in Table 5. In Figure 11, the current time step indicates the time at which the prediction is made, the window size indicates the length of the input sequential data in terms of time, and the time horizon indicates the interval between the time to be predicted and the current time step.

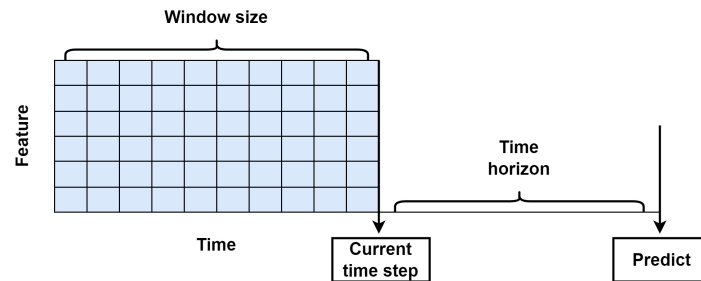


Figure 11. The illustration of fault prediction.

Table 5. LSTM architecture.

Layer	Design Parameters
Input layer	5 nodes
1st hidden layer(LSTM)	128 nodes (5 layer)/ReLU
2nd hidden layer(DNN)	640 nodes/ReLU
3rd hidden layer(DNN)	512 nodes/ReLU
4th hidden layer(DNN)	256 nodes/ReLU
5th hidden layer(DNN)	128 nodes/ReLU
6th hidden layer(DNN)	64 nodes/ReLU
7th hidden layer(DNN)	16 nodes/ReLU
8th hidden layer(DNN)	8 nodes/ReLU
Output layer(DNN)	1 nodes

In fault prediction, as the time span of the input data increases (i.e., the window size increases), the amount of information available increases, thereby enhancing prediction accuracy. On the other hand, as the time to be predicted gets further away from the current time step (i.e., the time horizon increases), the correlation between the input data and the prediction may decrease, leading to reduced prediction performance. Hence, we consider the different window sizes and time horizons across the datasets as shown in Table 6. For each dataset, the individual LSTM model is implemented according to its window size.

Table 6. Different window sizes and time horizons across datasets.

Dataset	Window Size (Hours)	Time Horizon (Hours)
Data 1	3.5	1
Data 2	12	1
Data 3	12	2
Data 4	12	2
Data 5	12	5
Data 6	12	5
Data 7	30	5
Data 8	30	5
Data 9	30	5
Data 10	30	5
Data 11	30	5

#### 4.4. Exploitation Phase

In the training phase, the feature selection filter is generated and the fault prediction model is trained as illustrated in Figure 5. Then, in the exploitation phase, they are used to transform real-time data into feature-reduced data and predict future fault conditions. Although our proposed framework is fully applicable in an environment where online data is being collected, it was not feasible to conduct experiments by directly operating a ship due to environmental constraints. Therefore, we simulate such an exploitation phase by using the test dataset. Specifically, the measurements at each time are provided to the prediction model according to the window size in Figure 11 (i.e., the prediction model cannot access future measurements). Then, the prediction model predicts the future target measurement using the data within the window size.

For every prediction, the time-series explanatory analysis of the predicted value can be performed. However, it does not have to be performed in every prediction, considering its computational cost. Therefore, we can consider a condition to perform the analysis (e.g., the model predicts the failure condition). In this application scenario, we establish two conditions: (1) situations where the actual temperature of the ship's engine sensor exceeds the failure threshold of 600 °C, and (2) situations where the predicted temperature increases rapidly. The time-series explanatory analysis is performed only if the predicted value satisfies one of the conditions. It is worth noting that we can arbitrarily design the conditions for robust fault prediction.

## 5. Experimental Results

### 5.1. Evaluation Metrics

In the experiments, we evaluate the performance of the proposed time-series explanatory fault prediction framework. First, to evaluate the performance of fault prediction, we use a mean squared error (MSE) on the prediction of the exhaust gas temperature as an evaluation metric. This shows how well the prediction model predicts the future target measurements. Furthermore, we should evaluate the explanatory capability of the proposed framework for the predictive model. However, it is challenging to quantify how well a framework interprets the predictive model. Besides, the actual fault data does not indicate the true factors and rationale for possible faults in the future, which implies that it is unknown whether the interpretation is correct or not. To address this issue, we design a new metric to evaluate the interpretation results of the predictions. The metric should indicate how well the interpretation results answer the question "Which features of which time period are important for prediction?" Fortunately, since we add pre-symptoms artificially to the normal dataset, we can distinguish between the periods of the symptoms of a fault and the normal periods throughout the entire dataset. If an explanatory method works well, it will indicate such periods of the pre-symptoms as the significant rationale of the fault prediction. Therefore, we can evaluate the interpretability performance by quantifying how much overlap there is between the pre-symptom periods and the periods indicated as significant by the proposed framework. We formally define the metric for feature  $i$  as

$$\text{Interpretability performance}_i = \frac{T_i^{\text{hit}}}{T_i^{\text{tot}}}, \quad (14)$$

where  $T_i^{\text{tot}}$  denotes the total intervals on feature  $i$  that are indicated as significant by the proposed framework and  $T_i^{\text{hit}}$  denotes the indicated intervals on feature  $i$  where the pre-symptom appears. As the evaluation metric in Equation (14) becomes closer to 1, the proposed framework has indicated the larger fault pre-symptom periods for feature  $i$ .

### 5.2. Performance Evaluation of Fault Prediction

As described in Section 4.3.2, we train an LSTM model on the training dataset generated in Section 4.2.2. To evaluate whether the proposed LSTM model effectively predicts faults, we train another DNN model that has an architecture in Table 7. When predicting

the future target measurement, the LSTM model uses the input features within its window considering their temporal sequence. On the other hand, the DNN architecture predicts the future target measurement without considering the temporal order of the input features. Thus, the DNN model does not consider the time-series characteristics of the data, while the LSTM model does.

**Table 7.** DNN architecture for fault prediction.

Layer	Info
Input layer	5 nodes
1st hidden layer	1280 nodes/ReLU
2nd hidden layer	512 nodes/ReLU
3rd hidden layer	256 nodes/ReLU
4th hidden layer	128 nodes/ReLU
5th hidden layer	64 nodes/ReLU
6th hidden layer	16 nodes/ReLU
7th hidden layer	8 nodes/ReLU
Output layer	1 nodes

We compare the performance metric for prediction (i.e., MSE) between the two models on the test dataset in Table 8. In the table, across all 11 datasets, the LSTM model consistently achieves lower MSE than the DNN model, indicating more accurate predictions.

This emphasizes the effectiveness of utilizing predictive models that consider time-series characteristics of sequential data.

**Table 8.** Prediction error of future target measurement.

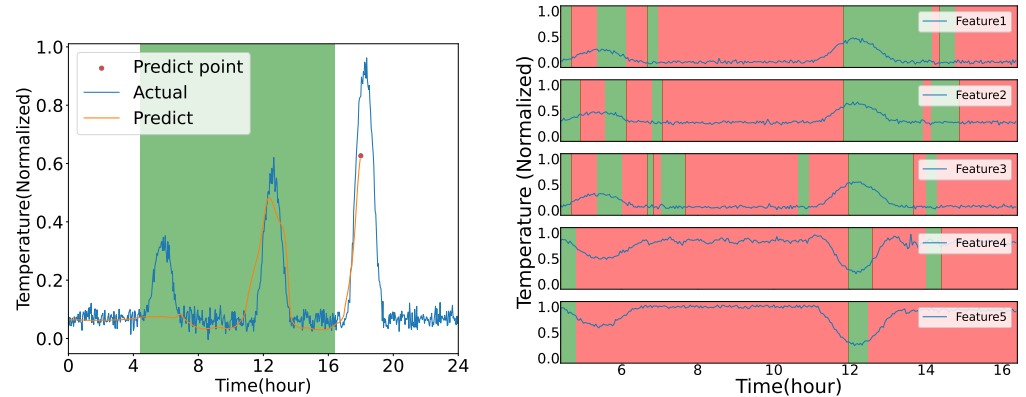
Data Name	LSTM	DNN
Data 1	0.0130	0.0245
Data 2	0.0159	0.0212
Data 3	0.0448	0.0455
Data 4	0.0409	0.0421
Data 5	0.0055	0.0060
Data 6	0.0039	0.0041
Data 7	0.1522	0.1589
Data 8	0.1405	0.1417
Data 9	0.1101	0.1108
Data 10	0.2190	0.2248
Data 11	0.1966	0.1970

### 5.3. Performance Evaluation of Time-Series Explanatory

As illustrated in Figure 5, the trained model provides predictions for future target measurements. Then, these predictions are interpreted as described in Section 3.3. The proposed time-series explanatory fault prediction framework differs from other XAI methods in that it considers adjacent time-series characteristics across input features when calculating the importance of input features. Therefore, it is particularly suitable for interpreting ship sensor data measured in sequential order. The results of interpreting the predictive model from Section 3.2 using our proposed framework are provided below.

In Figure 12, the fault prediction and its interpretation results are provided. The left panel depicts a case in which the predicted value exceeds 600 °C (i.e., the failure condition), which is a criterion for prediction interpretation. The ground truth data (blue line) illustrates one of the fault scenarios that belong to Data 4. It is randomly chosen from the test data. The predictive model is trained using the training dataset from Data 4 to predict ‘ME(S) 1 CYL EXH GAS TEMP H’. The predictions (orange line) are visualized along with the ground truth data. The intervals highlighted in green represent the period used as input features to the model (i.e., the window), and the red dot indicates the predicted future

values of 'ME(S) 1 CYL EXH GAS TEMP H' based on the input features highlighted. The right panel visualizes the interpretation results of five features by the proposed framework. The intervals indicated as significant pre-symptoms for fault prediction by the proposed framework are highlighted in green. In the right panel of the figure, the increase-then-decrease periods or decrease-then-increase periods describe the pre-symptoms for faults added in the data preparation. Then, from the results, we can see that the proposed framework successfully identified the added pre-symptoms patterns.



**Figure 12.** The visualization of the fault prediction and its interpretation results.

To evaluate the interpretability performance in deriving the rationale of predictions, we use the evaluation metric defined in Equation (14). It evaluates how well an algorithm identifies which features of which time period are important. Note that the existing XAI methods for predictive maintenance do not consider time-series explanation (i.e., they only focus on the important features and do not identify the important time period). Therefore, their time-series explanation cannot be evaluated. Here, we evaluate the proposed framework only. For the evaluation, we randomly choose five measurement scenarios from the test data for each data. The earliest fault prediction for each measurement scenario is chosen to calculate the interpretability performance defined in Section 5.1. The interpretability performance of the five predictions is averaged. The results are shown in Table 9. In Table 9, upon examining the average interpretability performance of each dataset, it can be observed that the performance is evenly distributed across all 11 datasets. The overall average of the average interpretability performance across all datasets is 0.76, indicating that the XAI effectively identifies the main causes of fault predictions.

**Table 9.** Interpretability performance of fault prediction.

Data Name	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	AVG
Data 1	0.89	0.80	0.84	0.55	0.62	0.83
Data 2	0.72	0.73	0.69	0.57	0.57	0.66
Data 3	0.76	0.76	0.78	0.37	0.55	0.65
Data 4	0.74	0.70	0.67	0.54	0.94	0.70
Data 5	1.00	0.60	0.78	1.00	0.73	0.80
Data 6	0.92	1.00	0.89	0.38	0.36	0.71
Data 7	0.88	0.94	0.96	0.78	1.00	0.91
Data 8	1.00	0.83	0.81	0.60	0.63	0.76
Data 9	0.74	0.92	0.90	0.83	0.80	0.84
Data 10	0.76	0.53	0.70	1.00	0.92	0.74
Data 11	0.88	0.80	0.76	0.68	0.74	0.78

The experimental results indicate that the form of the data learned by the fault prediction model did not significantly influence the interpretability results of XAI. This suggests that even if operational data differs in form from the data considered in this study, a certain level of performance can still be expected. This finding underscores the robustness

of the model's interpretability across different data formats, implying potential utility in real-world operational scenarios.

#### 5.4. Discussions

In the preceding experiments, we demonstrate that the proposed framework achieves a high interpretability performance in detecting pre-symptoms during fault prediction. This successful experimental result indicates that the interpretation results of the proposed framework can be used to understand the fault prediction model. In particular, the proposed framework can be utilized by both marine engineers and operators to maintain ships reliably.

For marine engineers, the proposed framework can be used to develop and apply fault prediction models based on multi-channel signal data from ships. In maritime conditions, disturbances in engine cycles and the wear time and speed of components and systems change in random and nonlinear ways. The multi-channel signal data collected under these conditions are likely to contain nonlinearity and outliers, making traditional statistical analysis methods like Pearson correlation inadequate. By using the proposed framework, it is possible to generate a more stable feature reduction filter that has a smaller information size but higher performance. In addition, when building a multi-channel signal-based fault prediction model for ships, the interpretation module allows engineers to verify that the model is trained as intended. Furthermore, for operators on ships, the interpretation module provides visualization of the prediction causes, offering evidence that the fault prediction model is making predictions in reasonable ways, thereby enhancing reliability. In addition, operators can accurately identify the types of signals and the time intervals that caused future faults, enabling precise preventive measures.

## 6. Conclusions

In this paper, we proposed a time-series explanatory fault prediction framework for a marine engine. The framework constitutes feature reduction via statistical correlation analysis and XAI, the fault prediction model, and the interpretation module to derive the main cause of prediction. The proposed framework not only predicts faults that could occur in the future but also provides the rationale for the predictions. Thus, it can be used to enhance the reliability and robustness of fault prediction. Through extensive experiments, we demonstrated that the proposed framework can effectively indicate the pre-symptoms that occur before faults. This clearly shows that the proposed framework can accurately highlight the significant regions of the input data in the feature-time domain, while the existing XAI methods in the field of predictive maintenance mainly focus on determining the impact of each feature without considering the time-domain characteristics. Consequently, the proposed framework can provide time-series explanatory insights for fault prediction, which helps in effective maintenance. As a future work, we plan to evaluate the performance of the proposed framework using actual ship failure data instead of synthetic fault behaviors. In addition, the proposed framework can be improved to consider a variety of maritime conditions. For example, extreme maritime conditions, such as strong ocean currents and stormy waters, can accentuate the random and nonlinear nature of disturbances in engine cycles and the wear time and speed of components and systems.

**Author Contributions:** Conceptualization, H.J.-G., S.K., J.-H.Y. and H.-S.L.; methodology, H.J.-G., S.K. and H.-S.L.; software, H.J.-G., S.-H.P. and J.-U.K.; validation, J.-H.Y. and H.-S.L.; formal analysis, J.-H.Y. and H.-S.L.; investigation, H.J.-G. and Y.-S.P.; resources, S.K., J.-H.Y. and H.-S.L.; data curation, S.K., J.-H.Y. and H.-S.L.; writing—original draft preparation, H.J.-G., Y.-S.P., S.-H.P. and J.-U.K.; writing—review and editing, S.K. and H.-S.L.; visualization, H.J.-G., S.-H.P. and J.-U.K.; supervision, S.K. and H.-S.L.; project administration, J.-H.Y. and H.-S.L.; funding acquisition, S.K., J.-H.Y. and H.-S.L. All authors have read and agreed to the published version of the manuscript.



**Funding:** This work was supported in part by the Ministry of Science and ICT (MSIT), Republic of Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01816) supervised by the Institute for Information and Communications Technology, Planning and Evaluation (IITP) (50%) and in part by the MSIT, Republic of Korea, through the ICT Challenge and Advanced Network of HRD (ICAN) program (IITP-2024-RS-2022-00156345), under the supervision of IITP (50%).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing is not applicable to this article due to privacy and security issues.

**Conflicts of Interest:** Author Jung-Hee Yang was employed by the company Smart Ship Solution Department, Hanwha Ocean Co. Ltd., Republic of Korea. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Nomenclature

$\mathcal{D}$	Dataset
$N$	Total number of samples (total length of the time-series data)
$i$	Index of samples (also considered as the time order of the data)
$k$	Feature of samples
$d_X$	Dimension of input data
$d_Y$	Dimension of target data
$\mathbf{x}$	Input time-series data
$x_{i,k}$	Sample of $\mathbf{x}$ at the $i$ -th index and $k$ -th feature
$\mathbf{y}$	Target fault-related vector
$P$	Pearson correlation coefficient
$M$	Mask for learning key predictors
$V$	Black-box model to be interpreted
$m_{i,k}$	Sample of $M$ at the $i$ -th index and $k$ -th feature
$\lambda_e, \lambda_a, \lambda_c$	Hyperparameters determining the weights of each loss term
$\mathcal{L}_e$	Error loss term considered in $M$ learning
$\mathcal{L}_a$	Area constraint loss term considered in $M$ learning
$\mathcal{L}_c$	Mask smoothing term considered in $M$ learning
$a$	Area constraint hyperparameter
$r_a$	Term consisting of $(1 - a)$ zeros and $a$ ones for the area constraint
$\Pi$	Perturbation operator
$W$	Hyperparameter determining the length of the surrounding time series considered when the perturbation operator is applied
$\mu_{i,k}$	Mean value from $x_{i-W,k}$ to $x_{i+W,k}$

## Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
CNN	Convolutional neural network
DNN	Deep neural network
FMEA	Failure modes and effects analysis
FTA	Fault tree analysis
LNGC	Liquefied natural gas carrier
LSTM	Long short-term memory
ML	Machine learning
NCR	Normal continuous rating

RNN	Recurrent neural network
RPM	Revolutions per minute
SHAP	Shapley additive explanations
VAE	Variational autoencoder
XAI	Explainable AI

## References

- Hountalas, D.T. Prediction of marine diesel engine performance under fault conditions. *Appl. Therm. Eng.* **2000**, *20*, 1753–1783. [[CrossRef](#)]
- Wan, Z.; Su, Y.; Li, Z.; Zhang, X.; Zhang, Q.; Chen, J. Analysis of the impact of Suez Canal blockage on the global shipping network. *Ocean. Coast. Manag.* **2023**, *245*, 106868. [[CrossRef](#)]
- Özkanlısoy, Ö.; Akkartal, E. The effect of Suez Canal blockage on supply chains. *Dokuz Eylül Üniversitesi Denizcilik Fakültesi Derg.* **2022**, *14*, 51–79. [[CrossRef](#)]
- Park, J.; Oh, J. Analysis of collected data and establishment of an abnormal data detection algorithm using principal component analysis and K-nearest neighbors for predictive maintenance of ship propulsion engine. *Processes* **2022**, *10*, 2392. [[CrossRef](#)]
- Ellefsen, A.L.; Æsøy, V.; Ushakov, S.; Zhang, H. A comprehensive survey of prognostics and health management based on deep learning for autonomous ships. *IEEE Trans. Reliab.* **2019**, *68*, 720–740. [[CrossRef](#)]
- Bennetot, A.; Donadello, I.; Qadi, A.E.; Dragoni, M.; Frossard, T.; Wagner, B.; Saranti, A.; Tulli, S.; Trocan, M.; Chatila, R.; et al. A practical guide on explainable AI techniques applied on biomedical use case applications. *arXiv* **2021**, arXiv:2111.14260.
- Jimenez, V.J.; Bouhmala, N.; Gausdal, A.H. Developing a predictive maintenance model for vessel machinery. *J. Ocean. Eng. Sci.* **2020**, *5*, 358–386. [[CrossRef](#)]
- Yan, R.; Wang, S. Ship detention prediction using anomaly detection in port state control: Model and explanation. *Electron. Res. Arch.* **2022**, *30*, 3679–3691. [[CrossRef](#)]
- de Brito Duarte, R.; Correia, F.; Arriaga, P.; Paiva, A. AI trust: Can Explainable AI enhance warranted trust? *Hum. Behav. Emerg. Technol.* **2023**, *2023*, 4637678. [[CrossRef](#)]
- Glomsrud, J.A.; Ødegårdstuen, A.; Clair, A.L.S.; Smogeli, Ø. Trustworthy versus explainable AI in autonomous vessels. In Proceedings of the International Seminar on Safety and Security of Autonomous Vessels (ISSAV) and European STAMP Workshop and Conference (ESWC), Espoo, Finland, 17–18 September 2019; Volume 37.
- Nath, K.; Meng, X.; Smith, D.J.; Karniadakis, G.E. Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines. *Sci. Rep.* **2023**, *13*, 13683. [[CrossRef](#)] [[PubMed](#)]
- Zocco, F.; Wang, H.C.; Van, M. Digital twins for marine operations: A brief review on their implementation. *arXiv* **2023**, arXiv:2301.09574.
- Youssef, A.; Noura, H.; Amrani, A.E.; Adel, E.M.E.; Ouladsine, M. A survey on data-driven fault diagnostic techniques for marine diesel engines. *arXiv* **2024**, arXiv:2404.10363.
- Fedorishin, D.; Forte, L.; Schneider, P.; Setlur, S.; Govindaraju, V. Fine-grained engine fault sound event detection using multimodal signals. In Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 1186–1190.
- Ellefsen, A.L.; Han, P.; Cheng, X.; Holmeset, F.T.; Æsøy, V.; Zhang, H. Online fault detection in autonomous ferries: Using fault-type independent spectral anomaly detection. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 8216–8225. [[CrossRef](#)]
- Velasco-Gallego, C.; De Maya, B.N.; Molina, C.M.; Lazakis, I.; Mateo, N.C. Recent advancements in data-driven methodologies for the fault diagnosis and prognosis of marine systems: A systematic review. *Ocean. Eng.* **2023**, *284*, 115277. [[CrossRef](#)]
- Velasco-Gallego, C.; Lazakis, I. RADIS: A real-time anomaly detection intelligent system for fault diagnosis of marine machinery. *Expert Syst. Appl.* **2022**, *204*, 117634. [[CrossRef](#)]
- Marins, M.A.; Barros, B.D.; Santos, I.H.; Barrionuevo, D.C.; Vargas, R.E.; Prego, T.d.M.; de Lima, A.A.; de Campos, M.L.; da Silva, E.A.; Netto, S.L. Fault detection and classification in oil wells and production/service lines using random forest. *J. Pet. Sci. Eng.* **2021**, *197*, 107879. [[CrossRef](#)]
- Tan, Y.; Zhang, J.; Tian, H.; Jiang, D.; Guo, L.; Wang, G.; Lin, Y. Multi-label classification for simultaneous fault diagnosis of marine machinery: A comparative study. *Ocean. Eng.* **2021**, *239*, 109723. [[CrossRef](#)]
- Xu, N.; Yang, L.; Lazzaretto, A.; Masi, M.; Shen, Z.; Fu, Y.; Wang, J. Fault location in a marine low speed two stroke diesel engine using the characteristic curves method. *Electron. Res. Arch.* **2023**, *31*, 3915–3942. [[CrossRef](#)]
- Karatuç, Ç.; Tadros, M.; Ventura, M.; Soares, C.G. Strategy for ship energy efficiency based on optimization model and data-driven approach. *Ocean. Eng.* **2023**, *279*, 114397. [[CrossRef](#)]
- Ji, Z.; Gan, H.; Liu, B. A deep learning-based fault warning model for exhaust temperature prediction and fault warning of marine diesel engine. *J. Mar. Sci. Eng.* **2023**, *11*, 1509. [[CrossRef](#)]
- Han, P.; Ellefsen, A.L.; Li, G.; Æsøy, V.; Zhang, H. Fault prognostics using LSTM networks: Application to marine diesel engine. *IEEE Sens. J.* **2021**, *21*, 25986–25994. [[CrossRef](#)]
- Sun, T.; Chen, Y.; Zhou, Y. Fault prediction of marine diesel engine based on time series and support vector machine. In Proceedings of the 2020 International Conference on Intelligent Design (ICID), Xi'an, China, 11–13 December 2020; pp. 75–81.

25. Tong, Z.; Sun, Y.; She, J.; Zhu, Y.; Zhao, Z. Identification of typical fault states of marine diesel engines based on optimized BP neural network. *Highlights Sci. Eng. Technol.* **2022**, *7*, 10–18. [[CrossRef](#)]
26. Lazakis, I.; Raptodimos, Y.; Varelas, T. Predicting ship machinery system condition through analytical reliability tools and artificial neural networks. *Ocean. Eng.* **2018**, *152*, 404–415. [[CrossRef](#)]
27. Qi, Z.; Qi, Y.; Hu, G. Research on fault prediction for marine diesel engines. *J. Comput. Commun.* **2020**, *8*, 36–44. [[CrossRef](#)]
28. Hong, C.W.; Lee, C.; Lee, K.; Ko, M.S.; Kim, D.E.; Hur, K. Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors* **2020**, *20*, 6626. [[CrossRef](#)] [[PubMed](#)]
29. Armstrong, R.A. Should Pearson’s correlation coefficient be avoided? *Ophthalmic Physiol. Opt.* **2019**, *39*, 316–327. [[CrossRef](#)] [[PubMed](#)]
30. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [[CrossRef](#)]
31. Marcílio, W.E.; Eler, D.M. From explanations to feature selection: Assessing SHAP values as feature selection mechanism. In Proceedings of the 2020 33rd SIBGRAPI conference on Graphics, Patterns and Images (SIBGRAPI), Virtual, 7–10 November 2020; pp. 340–347.
32. Yuan, Y.; Shao, C.; Cao, Z.; He, Z.; Zhu, C.; Wang, Y.; Jang, V. Bus dynamic travel time prediction: Using a deep feature extraction framework based on RNN and DNN. *Electronics* **2020**, *9*, 1876. [[CrossRef](#)]
33. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [[CrossRef](#)]
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.