IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2019, accepted January 23, 2019. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.2895394

An Unsupervised Reconstruction-Based Fault Detection Algorithm for Maritime Components

ANDRÉ LISTOU ELLEFSEN, EMIL BJØRLYKHAUG, VILMAR ÆSØY, AND HOUXIANG ZHANG, (Senior Member, IEEE)

Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway

Corresponding author: André Listou Ellefsen (andre.ellefsen@ntnu.no)

This work was supported in part by the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, under Grant 90329106, and in part by the Research Council of Norway, under Grant 280703.

ABSTRACT In recent years, the reliability and safety requirements of ship systems have increased drastically. This has prompted a paradigm shift toward the development of prognostics and health management (PHM) approaches for these systems' critical maritime components. In light of harsh environmental conditions with varying operational loads, and a lack of fault labels in the maritime industry generally, any PHM solution for maritime components should include independent and intelligent fault detection algorithms that can report faults automatically. In this paper, we propose an unsupervised reconstruction-based fault detection algorithm for maritime components. The advantages of the proposed algorithm are verified on five different data sets of real operational run-to-failure data provided by a highly regarded industrial company. Each data set is subject to a fault at an unknown time step. In addition, different magnitudes of random white Gaussian noise are applied to each data set in order to create several real-life situations. The results suggest that the algorithm is highly suitable to be included as part of a pure data-driven diagnostics approach in future end-to-end PHM system solutions.

INDEX TERMS Automatic fault detection, deep learning, maritime industry, prognostics and health management, unsupervised learning.

I. INTRODUCTION

Ship systems are more complex and integrated than ever before. Thus, the degradation of critical maritime components included in these systems poses a serious threat to safe and profitable maritime operations [1]. In general, maintenance in shipping either follows a reactive maintenance (RM) or preventive maintenance (PvM) approach [2]. RM can be described as post-failure repair, and hence, it will create large and unnecessary costs when critical maritime component failures occur during operation. PvM involves predetermined maintenance intervals based on constant intervals or age-based or imperfect maintenance [3]. PvM will, of course, provide high reliability, but it involves unneeded maintenance inspections and procedures involving completely functional systems. Additionally, critical maritime components are, in fact, subject to random failure patterns due to different environmental conditions with varying operational loads [4]. Neither RM nor PvM is sufficient to identify these kinds of failures. The need for prognostics and health management (PHM) approaches which incorporate automatic fault detection and associated remaining useful life (RUL) predictions is urgent. RUL predictions aim to obtain the ideal maintenance policy through predictions of the available time until failure after a fault is detected within the component [5]. In this way, PHM approaches have the potential to prevent critical maritime component failures, and hence, considerably enhance maritime operational performance and safety [6].

Recently, deep learning (DL) has emerged as a potent data-driven area for accurate RUL predictions for component degradation [5], [7]. RUL-based DL techniques utilize raw input sensor data and are less dependent on prior domain knowledge of component mechanics. However, they depend on large, labeled run-to-failure data in the training process. Thus, the RUL predictions strongly depend on the accuracy of the fault detection algorithm, that is, the process of separating normal operating data from faulty degradation data in order to create run-to-failure labels.

The associate editor coordinating the review of this manuscript and approving it for publication was Dong Wang.

In general, traditional fault detection algorithms based on signal processing methods, such as Empirical Mode Decomposition [8] and Wavelet Transform [9], are to some extent application specific and need prior domain knowledge to distinguish normal operating data from faulty degradation data. Due to varying operational conditions, fault detection algorithms for critical maritime components should not be application specific. Additionally, with respect to the maritime industry generally, there is a lack of fault labels of critical maritime components [10]. This creates major issues towards successful implementation of fault detection algorithms that utilize a supervised classifier to separate normal operating data from faulty degradation data [11]. Thus, maritime components require independent and intelligent fault detection algorithms in order to detect and report faults automatically.

This paper investigates the possibilities for automatic fault detection within maritime components. In order to do so, an unsupervised reconstruction-based fault detection algorithm for maritime components is introduced. The algorithm can be applied to several machine learning (ML) algorithms and encoder-decoder (ED)-structured DL techniques. Thus, it will be tested on four techniques: traditional Feed-forward Neural Network with one hidden layer (1FNN), Autoencoder (AE), Variational Autoencoder (VAE), and Long-Short Term Memory (LSTM). Each technique is trained and evaluated on five different data sets of real operational run-to-failure data of the same maritime component collected from a highly regarded industrial company. Each data set is subject to a fault at an unknown time step. Additionally, different magnitudes of random white Gaussian noise are applied to each data set to create several real-life situations in order to test the robustness of the algorithm. First, the algorithm estimates an anomaly score function by calculating a reconstruction error at each time step in faulty degradation data. Then, the algorithm detects a fault automatically by estimating the time step with the highest acceleration in the anomaly score function. This study's main contributions are as follows:

- ED-structured DL techniques prove robustness towards noisy real operational input data.
- The proposed algorithm is not application specific, that is, the algorithm proves consistent high accuracy in real operational input data when subjected to varying operational conditions. Additionally, the algorithm is considered more generic than fault indications based on user-specified threshold values.
- The proposed algorithm reports faults automatically with no prior knowledge of component degradation mechanics.

The overall organization of the paper is as follows. Section II introduces recent and related work on intelligent fault detection algorithms. Section III introduces the necessary background on traditional FNN and ED-structured DL techniques. The experimental approach, results, and discussions are considered in section IV. Finally, Section V concludes and closes the paper and provides directions for future work.

II. RELATED WORK

The development of intelligent fault detection algorithms has exploded in the last two years. The majority is based on reconstruction-based fault detection by applying a reconstruction error as an anomaly score. The core idea is to train a specific machine learning (ML) algorithm, in an unsupervised manner, to reconstruct normal operating data. The ML algorithm will then provide a higher reconstruction error on unforeseen trends in faulty degradation data. Brandsæter *et al.* [12]used Auto Associative Kernel Regression (AAKR) for reconstruction and the Sequential Probability Ratio Test for anomaly detection provided. In order to determine the fault condition, a lower bound and upper bound threshold value was used. Yang *et al.* [13] used Support Vector Regression (SVR) for reconstruction and probability information based on three statistical indexes for anomaly detection. However, both AAKR and SVR are considered shallow ML algorithms which might not reconstruct high-dimensional and noisy operational data accurately. ED-structured DL techniques are well-suited to first compress and then reconstruct such operational data. The compressed version of the input supports the reconstruction process to extract information relevant to the normal operating data. In this way, ED-structured DL techniques cannot reconstruct unforeseen patterns in faulty degradation data, which results in a larger reconstruction error.

Recent studies have employed variations on the traditional AE for fault detection of rolling bearings, verified on the data set provided by Case Western Reserve University Bearing Data Center [14]. Lu *et al.* [15] demonstrated the effectiveness of a Stacked Denoising Autoencoder (SDA). The SDA showed improved accuracy for signals containing ambient noise and different working loads compared to traditional fault detection algorithms. Nevertheless, the accuracy indicated inconsistency between different working loads. Liu *et al.* [16] used a Gated Recurrent Unit-based nonlinear predictive Denoising Autoencoder (GRU-NP-DAE) provided. The proposed method showed improved accuracy compared to several state-of-the-art methods, including the SDA provided in [15]. Both the SDA and the GRU-NP-DAE trained a supervised classifier to separate normal operating data from faulty degradation data. Thus, both approaches require fault labels in the training process. Additionally, the approaches were trained under a de-noising criterion [17], that is, the input was corrupted stochastically while the target for reconstruction was kept as the original input. To make full use of both acoustic and vibratory signals, Li *et al.* [18] used a deep random forest fusion (DRFF) technique. The proposed approach combined deep feature representations and data fusion strategies to show improved performance of gearbox fault diagnostics. Nevertheless, the DRFF technique also trained a supervised classifier.

Although the above approaches have shown superior fault detection accuracy compared to traditional fault detection algorithms, they are less suitable for maritime components. First, maritime components are subjected to varying

environmental and operating conditions. Thus, a suitable fault detection algorithm should not rely on user-specified threshold values. Second, supervised classifiers require fault labels in the training process. This is a barrier given that there is a common lack of fault labels in the maritime industry. Finally, maritime components are subjected to random amounts of noise in real operational input data. Thus the de-noising criterion is not completely realistic, as in real-life situations the target for reconstruction will also contain the noise. Hence, maritime components require more independent and intelligent fault detection algorithms.

In the last two years, independent and intelligent fault detection algorithms have begun to develop. Park *et al.* [19] introduced an LSTM based Variational Autoencoder (LSTM-VAE) anomaly detector for robot-assisted feeding. The LSTM-VAE reports an anomaly when a reconstruction-based anomaly score is higher than a varying state-based threshold. The threshold changes over the estimated state of a task execution. Malhotra *et al.* [20] used an LSTM approach to reconstruct time-series data. The reconstruction error was used to compute a health index (HI) curve. Then, the HI curve was used to create run-to-failure labels in order to predict the RUL. The unsupervised reconstruction-based fault detection algorithm for maritime components which we propose in this work follows the idea of generic fault detection provided in [19]. However, the main difference is the utilization of the time step with the highest acceleration as varying fault indications. Additionally, the proposed algorithm can be further used to create run-to-failure labels in order to predict the RUL, similar to the approach in [20].

III. BACKGROUND ON ED STRUCTURED DL TECHNIQUES

This section will introduce the necessary background on the traditional FNN and the ED-structured DL techniques used in this study. First, FNN, AE, VAE, and LSTM are defined. Next, the configuration and performance evaluation of the unsupervised reconstruction models are elaborated.

A. FEED-FORWARD NEURAL NETWORK

Traditional FNNs form the basis of the ED-structured DL techniques used in this paper. FNNs aim to approximate some function f^* by mapping an input $\boldsymbol{x}$ to a target y, that is, $y = f^*(\boldsymbol{x})$. An FNN defines a mapping $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ and learns the value of the parameters $\boldsymbol{\theta}$, which consists of weights and biases, through the back-propagation algorithm [21]. FNNs are typically called networks because they are represented by combining together several layers [22]. Each unit in layer l computes its own activation value:

$$a_j^l = \sigma(z_j^l) \tag{1}$$

where σ is the activation function and the argument is the weighted sum

$$z_j^l = b_j^l + \sum_k w_{jk}^l a_k^{l-1} \tag{2}$$

of the output a_k^{l-1} from unit k in the previous layer $l-1$. b_j^l denotes the bias, while w_{jk}^l represent the weight factors. In the first hidden layer $l = 1$, the input is $a_j^0 = x_j$, where x_j, $j = 1 \ldots n$, are the inputs to the FNN. As each layer is fully connected, the weighted sum of the outputs of layer $l-1$ is over all units k.

B. AUTOENCODER

An AE is an FNN trained to reconstruct its input through a "bottleneck" representation of latent variables (hidden units) z [23]. As seen in Figure 1, the AE consists of an encoder function $z = f_{\theta_e}(x)$ and a decoder function that produces a reconstruction $r = g_{\theta_d}(z)$. The AE objective function is as follows [23]:

$$J_{AE}(\theta_e, \theta_d) = \sum L(x, r) \tag{3}$$

The optimization of the parameters θ_e and θ_d, which consist of weights and biases, are learned concurrently in the reconstruction process and compared to the original input data in order to obtain the lowest possible reconstruction error $L(x, r)$. In this work, $L(x, r)$ is the mean squared error (MSE), and hence, the AE objective function becomes:

$$J_{AE}(\theta_e, \theta_d) = \frac{1}{m} \sum_{i=1}^{m} ||x_i - g_{\theta_d}(f_{\theta_e}(x_i))||^2 \tag{4}$$

where m is the number of units in the input layer. AEs can be stacked with several hidden layers, depending on the dimensionality of the input data, and it is trained by the back-propagation algorithm. Significantly, unsupervised pre-training might be necessary for AEs with many hidden layers.

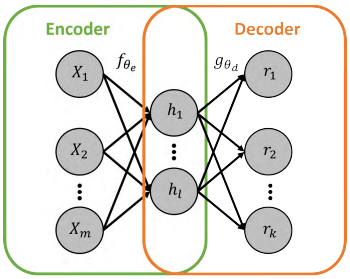

FIGURE 1. A simple illustration of an AE. *m* units in the input layer, *l* units in the hidden layer (bottleneck), and *k* units in the output layer.

C. VARIATIONAL AUTOENCODER

The VAE is a modern variation of the traditional AE, developed by Kingma and Welling [24]. Compared to the traditional AE, the VAE models the underlying probability distribution using Bayesian inference. Thus, the latent variables z are stochastic variables, and this improves generalization. As seen in Figure 2, the VAE consists of an encoder function $z = q_{\theta_e}(z|x)$ and a decoder function $r = p_{\theta_d}(x|z)$. The objective function of the VAE is to maximize the variational lower bound J_{VAE} associated with data point x [22]:

$$J_{VAE}(\theta_e, \theta_d) = -D_{KL}\big(q_{\theta_e}(z|x) \,||\, p_{\theta_d}(z)\big) + E_{q_{\theta_e}(z|x)}[\log p_{\theta_d}(x|z)] \tag{5}$$

where D_{KL} is the Kullback-Leibeler (KL) divergence. The first term provides a regularization since it measures how closely the latent variables match the encoder function (latent loss), while the second term is the reconstruction log-likelihood (generative loss). However, the reconstruction error term in Eq. 5 requires a Monte Carlo estimate of the expectation, and this is not easily differentiable [24]. A reparameterization trick of z is applied to obtain the gradients of the decoder in order to use the back-propagation algorithm. The reparameterization trick introduces a deterministic variable such that $z = \boldsymbol{\mu} + \boldsymbol{\sigma}\varepsilon$, $\varepsilon \sim \mathcal{N}(0, 1)$ [24]. Thus, the encoder now generates a vector of means and a vector of standard deviations instead of a vector of real values. As seen in Figure 2, these vectors are then used as the latent vector in the decoder. For real-valued input data, a Gaussian reconstruction distribution is used in the decoding process. Like AEs, the VAE can be stacked with several hidden layers depending on the dimensionality of the input data. Also, pretraining might be necessary with many hidden layers.

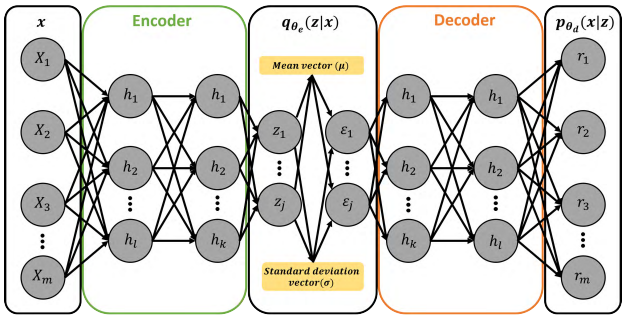

FIGURE 2. A simple illustration of a VAE. ***m*** units in the input layer, ***l*** and ***k*** units in the hidden layers of the encoder and decoder, and ***j*** units in latent vector.

D. LONG-SHORT TERM MEMORY

Today, modifications by [25]–[27] have been included in the original LSTM [28], and the literature refers to this as the ''vanilla LSTM''. This study uses ''vanilla LSTM'' with no peephole connections. As opposed to traditional Recurrent Neural Networks, the LSTM introduces a memory cell that regulates the information flow in and out of the cell. Thus, the memory cell is able to preserve its state over time, such that it learns long-term dependencies. As seen in Figure 3, the memory cell consists of three non-linear gating units that protect and regulate the cell state, $\boldsymbol{S}_t$ [29]:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{R}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \quad (6)$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{R}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \quad (7)$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{R}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \quad (8)$$

where σ is the logistic sigmoid gate activation function, $\sigma(x) = \frac{1}{1+e^{-x}}$, which provides a scaled value between 0 and 1. $\boldsymbol{W}$ is the input weight, $\boldsymbol{R}$ is the recurrent weight, and $\boldsymbol{b}$ is the bias weight. The new candidate state values, $\tilde{\boldsymbol{S}}_t$, are created

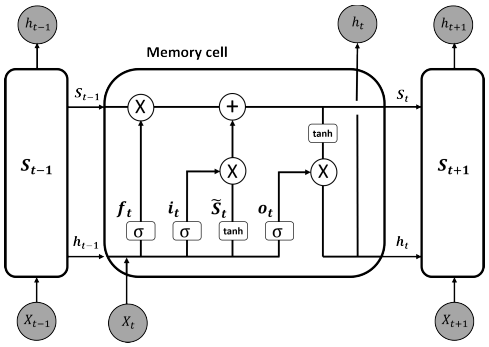

FIGURE 3. A simple illustration of an LSTM. f_t, i_t, and o_t represents the forget, input, and output gate, respectively.

by the tanh layer:

$$\tilde{\boldsymbol{S}}_t = tanh(\boldsymbol{W}_s \boldsymbol{x}_t + \boldsymbol{R}_s \boldsymbol{h}_{t-1} + \boldsymbol{b}_s) \quad (9)$$

The previous cell state, $\boldsymbol{S}_{t-1}$, is updated into the new cell state, $\boldsymbol{S}_t$, by

$$\boldsymbol{S}_t = \boldsymbol{f}_t \otimes \boldsymbol{S}_{t-1} + \boldsymbol{i}_t \otimes \tilde{\boldsymbol{S}}_t \quad (10)$$

where $\otimes$ denotes element-wise multiplication of two vectors. First, $\boldsymbol{f}_t$ determines which historical information the memory cell should forget. Then, $\boldsymbol{i}_t$ decides what new information in $\tilde{\boldsymbol{S}}_t$ the memory cell will input and store in $\boldsymbol{S}_t$. Finally, $\boldsymbol{o}_t$ determines which parts of $\boldsymbol{S}_t$ the memory cell will output:

$$\boldsymbol{h}_t = \boldsymbol{o}_t \otimes tanh(\boldsymbol{S}_t) \quad (11)$$

Through these equations, the LSTM has the ability to remove or add information to $\boldsymbol{S}_t$, which makes it highly suitable to process time-series data. Like AEs and VAEs, the LSTM is trained by the back-propagation algorithm and can be stacked with several hidden layers depending on the dimensionality of the input data.

E. UNSUPERVISED RECONSTRUCTION MODELS

In this study, 1FNN, AE, VAE, and LSTM are structured as an ED in order to create several diverse reconstruction models for comparison. The 1FNN is the simplest model and configured by one hidden layer with 14 units in both the encoder and decoder. In other words, the 1FNN is equal to an AE with one hidden layer. The AE, VAE, and LSTM are structured as deep models and configured by three hidden layers with 17, 8, and 4 units in the encoder and three hidden layers with 4, 8, and 17 units in the decoder, respectively. Let $\boldsymbol{x}_t = [x_1 \ldots x_n]_t$ denote the vector of input sensor measurements at time step t. Each reconstruction model is trained in an unsupervised manner, such that at each time step t the input $\boldsymbol{x}_t$ is also used as the target $\boldsymbol{y}_t$ for the reconstruction, $\boldsymbol{y}_t = \boldsymbol{x}_t$. A fully connected output layer is attached to each reconstruction model to handle error calculations. The selected loss function in the output layer is the MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} ||\hat{y}_i - y_i||^2 \quad (12)$$

where n is the number of sensors, and $\hat{y}_i$ and y_i are the i_{th} predicted and target sensor measurement, respectively.

IV. EXPERIMENTAL STUDY

In the following experimental study, each reconstruction model is trained and evaluated on five different data sets of real operational run-to-failure data of the same maritime component collected from an industrial company. First, each reconstruction model is trained on normal operating data. Next, an anomaly score function is estimated for each model by calculating the MSE, Eq. 12, at each time step in faulty degradation data. Finally, a generic and intelligent fault detection algorithm is employed to detect an unknown fault automatically. All experiments are run on NVIDIA GeForce GTX 1060 6 GB and the Microsoft Windows 10 operating system. The programming language is Java 8 and the deep learning library is "deeplearning4j" (DL4J) version 1.0.0-beta2 [30].

A. DATA SETS

The five data sets used in this study are provided by a highly regarded industrial company and collected from the same maritime component. A confidentiality agreement bars us from stating the actual name of the maritime component, fault types, and sensor measurements. The data sets start with different operational loads and corresponding sensor measurements. As seen in Table 1, each data set differs in total time step length T_{total}. Data sets 1 and 4 are subjected to fault type A, while data set 2, 3, and 5 are subjected to fault type B. In each data set, the maritime component operates in normal conditions at the start, then begin to degrade at an unknown point during the time series. The degradation grows in magnitude until failure. Thus, the main objective is automatically to detect the time step where the degradation starts, that is, the fault time step f_t. In order to train the reconstruction models, the initial 25% of each data set is considered normal operating data (training data) and the remaining 75% is considered faulty degradation data (test data). Thus, the total time step lengths in the training and test data are $T_{nod} = T_{total} \cdot 0.25$ and $T_{fdd} = T_{total} \cdot 0.75$, respectively. Each data set has 14 sensor measurements.

TABLE 1. Real operational run-to-failure data sets of a maritime component.

Data set	Fault type	T_{total}	T_{nod}	T_{fdd}	w
1	A	887	222	665	19
2	B	909	227	682	19.5
3	B	1859	465	1394	39.8
4	A	2554	638	1916	54.7
5	B	3643	911	2732	78.1

B. DATA NORMALIZATION AND PREPARATION

Each sensor measurement x_n in the input and target vector, $\boldsymbol{y}_t = \boldsymbol{x}_t = [x_1 \dots x_n]_t$, is normalized with zero mean and unit variance (z-score) normalization:

$$\hat{x}_n = \frac{x_n - \mu}{\sigma} \tag{13}$$

where μ and σ is the mean and the corresponding standard deviation of the population, respectively. Additionally, maritime components are subjected to random amounts of noise in real operational input data. Thus, to increase the complexity of each training data set and create differentiated real-life maritime situations, different magnitudes of random white Gaussian noise, g, is added to each $\hat{x}_n$ at each time step t. We assume that the real world noise is random white Gaussian noise. P_{signal} and P_{noise} are the average power of the signal and the noise in the training data, respectively, and defined as follows:

$$P_{signal} = \frac{1}{T_{nod}} \sum_{t=1}^{T_{nod}} \left(\sqrt{\frac{1}{n}\left(\hat{x}_1^2 + \dots + \hat{x}_n^2\right)} \right)_t \tag{14}$$

$$P_{noise} = \frac{1}{T_{nod}} \sum_{t=1}^{T_{nod}} \left(\sqrt{\frac{1}{n}\left((\hat{x}_1 + g)^2 + \dots + (\hat{x}_n + g)^2\right)} \right)_t \tag{15}$$

Then, the signal-to-noise-ratio (SNR) can be defined as:

$$SNR(\%) = \frac{P_{signal}}{P_{noise}} \cdot 100 \tag{16}$$

C. CONFIGURATION AND TRAINING

The reconstruction models are configured with joint hyper-parameters in order to make reliable comparisons. Stochastic gradient descent (SGD) is the selected optimization algorithm and adaptive moment estimation (Adam) is the learning rate method. The learning rate is $l_r = 10^{-3}$ and the $l2$ regularization value is 10^{-4}. Xavier weight initialization is applied to all layers. The rectified linear unit (ReLU) activation function is used in 1FNN, AE, and VAE. However, in the LSTM, the tanh activation function is used in order to push the input and output values between -1 and 1. The selected hyper-parameters are summarized in Table 2. During the training process of each reconstruction model, an early stopping (ES) approach is used in order to reconstruct the normal operating data as accurately as possible. The ES approach monitors the total reconstruction error of all time steps $E_{T_{nod}}$ for each epoch in the training data:

$$E_{T_{nod}} = \sum_{t=1}^{T_{nod}} \left(\frac{1}{n} \sum_{i=1}^{n} ||\hat{y}_i - y_i||^2 \right)_t \tag{17}$$

If the number of epochs with no reduction on $E_{T_{nod}}$ exceeds four, the training process is terminated. Then, the reconstruction model, in the epoch with the lowest $E_{T_{nod}}$, is saved and evaluated on the faulty degradation data.

TABLE 2. Joint hyper-parameters.

Hyper-parameter	Method/value
Optimization algorithm	SGD
l_r method	Adam
l_r	10^{-3}
$l2$ regularization	10^{-4}
Weight initialization	Xavier
Activation function	ReLU (tanh in LSTM)

D. PREDICTION OF FAULT TIME STEP IN FAULTY DEGRADATION DATA

The anomaly score function is estimated by calculating the MSE, Eq. 12, at each time step in the faulty degradation data. Then, the calculations and the corresponding time steps are saved in a score list S_l and a time step list T_l, respectively. Next, a generic and intelligent fault detection algorithm is employed in order to predict the fault time step $\hat{f}_t$. First, the algorithm creates three sliding windows of length $w = T_{fdd}/35$. Table 1 shows w for each data set. The value of 35 is used for all data sets in order to keep the same percentage level, that is $(1/35) \cdot 100 = 2.86\%$, on the faulty degradation data. In this work, the value of 35 is based on trial and error. However, w is a critical parameter and should be tuned carefully for other practical applications. The value of w will depend on the amount of noise in S_l. Second, the three windows slide across S_l for each time step in T_l. A distance equal to w is used between each sliding window. In order to remove noise in S_l, the average reconstruction score S_{avg} is calculated in the three windows. Third, the velocity v between windows 1 and 2 and between windows 2 and 3 are calculated. Finally, the acceleration a and the corresponding $\hat{f}_t$ are estimated. The sliding window operation is illustrated in Figure 4 and the proposed algorithm is elaborated in Algorithm 1. Large sensor measurements deviations compared to typical sensor measurements in normal operating data is a valid indication of a fault. The aim of the proposed algorithm is to detect the time step with the highest acceleration a_{max} in faulty degradation data. a_{max} is used as the fault criterion since this point indicates increasing v, and hence, a rapid increase in S_l. This increase in v indicates that one or several sensor measurements have started to deviate from the normal operating data rapidly. Due to latency in physical components, a_{max} is a better indication of a fault than the highest increase in v, since there is a time delay before the fault will result in large sensor measurements deviations. The proposed algorithm is considered more generic than previous fault indications based on threshold values.

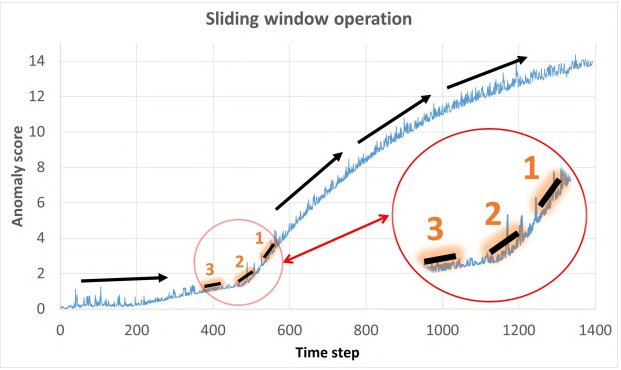

FIGURE 4. Illustration of the sliding window operation. Three windows (highlighted in orange) slide across S_l through time.

Algorithm 1 Algorithm for Calculating the Time Step With the Highest Acceleration in Faulty Degradation Data

Input: w, S_l, T_l, T_{fdd}
Output: $\hat{f}_t$
Initialisation :
$a_{max} \leftarrow 0$
$w \leftarrow T_{fdd}$ / 35
Creating three sliding windows of length w which slide across S_l for each time step in T_l.
A distance equal to w is used between each sliding window.
S_{avg} is calculated in each sliding window.
for $i := 0$ to T_{fdd} **do**
  $v1 \leftarrow S_{avg}1$ - $S_{avg}2$
  $v2 \leftarrow S_{avg}2$ - $S_{avg}3$
  $a \leftarrow v1$ - $v2$
  if $(a > a_{max})$ **then**
    $a_{max} \leftarrow a$
    $\hat{f}_t \leftarrow T_l[i]$ - $(w \cdot 2.5)$
    w is multiplied by 2.5 in order to find the center of the sliding-window operation.
  end if
end for
return $\hat{f}_t$

E. EXPERIMENTAL RESULTS AND DISCUSSION

The predicted fault time step $\hat{f}_t$ for each reconstruction model is shown in Table 3. In order to evaluate the results, valuable domain knowledge, provided by the industrial company, is used to determine the true fault time step f_t for each data set. In Table 3, the predicted fault time step is highlighted when $\hat{f}_t = f_t$. Four different real-life situations are created by applying 100%, 90%, 80%, and 70% SNRs to the training data in order to test the robustness of each reconstruction model. Additionally, to minimize any prediction performance bias, the training and evaluation process for each real-life situation is repeated five times for each reconstruction model. Then, the average $\hat{f}_t$ is calculated, as shown in Table 3. With reduced SNR, the input and target vector for reconstruction are corrupted stochastically, meaning $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t = \boldsymbol{y}_t$. An alternative approach is to train the reconstruction models under a de-noising criterion [17], that is, the input vector is stochastically corrupted $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t$ while the target vector is kept as the original input $\boldsymbol{y}_t = \boldsymbol{x}_t$. However, when trained in an unsupervised manner, this criterion is considered unrealistic, given the likelihood of noisy input data in real-life situations.

As seen in Table 4, $E_{T_{nod}}$ increases along with reduced SNR for the deep models, AE, VAE, and LSTM. To this extent, reduced SNR is a regularization technique that reduces overfitting. Thus, the deep models achieve robust feature extractions and are forced to generalize on the trends in the training data. Therefore, as seen in Table 3, the deep models actually improve or maintain the same prediction performance on the faulty degradation data even as the SNR on the training

TABLE 3. Predicted fault time step $\hat{f}_t$ compared to true fault time step f_t on the faulty degradation data for each reconstruction model.

Data set	Fault type	T_{fdd}	f_t	$\hat{f}_t$ SNR(%)	1FNN	AE	VAE	LSTM
1	A	665	157	100	148	151	154	158
				90	367	151	153	158
				80	412	154	152	158
				70	476	154	153	158
2	B	682	148	100	**148**	146	**148**	155
				90	152	150	147	**148**
				80	381	150	146	150
				70	463	149	146	161
3	B	1394	477	100	492	455	**477**	481
				90	480	479	479	481
				80	632	479	479	481
				70	791	481	482	481
4	A	1916	1306	100	1281	1281	1281	1281
				90	1278	1281	1281	1281
				80	1280	1282	1281	1281
				70	1282	1281	1281	1281
5	B	2732	787	100	807	752	807	732
				90	866	655	783	739
				80	932	728	796	740
				70	1043	732	800	742

TABLE 4. Total reconstruction error $E_{T_{nod}}$ on the training data for each reconstruction model.

Data set	Fault type	T_{nod}	$E_{T_{nod}}$ SNR(%)	1FNN	AE	VAE	LSTM
1	A	222	100	0.74	22.40	26.60	39.90
			90	0.41	52.04	62.52	81.89
			80	0.43	100.23	114.86	143.91
			70	0.45	166.26	187.41	225.86
2	B	227	100	0.93	18.50	47.30	60.50
			90	0.44	62.50	79.34	101.61
			80	0.42	108.37	131.51	165.66
			70	0.45	178.53	204.50	252.08
3	B	465	100	1.85	23.30	41.80	77.60
			90	0.84	103.35	115.65	153.64
			80	0.86	218.79	229.03	293.27
			70	0.93	380.95	385.17	476.72
4	A	638	100	2.55	7.20	5.90	15.40
			90	9.84	128.62	109.80	135.08
			80	13.95	301.69	268.16	325.66
			70	11.31	545.57	489.48	592.33
5	B	911	100	8.63	29.90	30.20	204.20
			90	5.07	199.88	182.16	378.73
			80	1.76	465.41	396.02	644.59
			70	1.93	784.34	714.80	1015.63

data reduces. Table 4 also shows that $E_{T_{nod}}$ decreases along with reduced SNR in data sets 1, 2, 3, and 5 for the 1FNN. Thus, the 1FNN learns the noise rather than the trends in the training data. This noise, obviously, is not part of the faulty degradation data. Therefore, as seen in Table 3, the 1FNN provides worse and less consistent prediction performance on the faulty degradation data as the SNR on the training data reduces. Nevertheless, $E_{T_{nod}}$ increases with reduced SNR in data set 4 for the 1FNN. This results in equal prediction performance on the faulty degradation data as the deep models.

The accuracy evaluations on the faulty degradation data in the four real-life situations for each reconstruction model are shown in Tables 5, 6, 7, and 8, respectively. The accuracy is defined as follows:

$$Accuracy\,(\%) = \left(1 - \frac{||\hat{f}_t - f_t||}{T_{fdd}}\right) \cdot 100 \qquad (18)$$

The 1FNN provides inconsistently average accuracy performance in the four situations. The average accuracy decreases along with reduced SNR, and hence, confirms the influences of noise. As opposed to the 1FNN, the deep models

TABLE 5. Accuracy evaluation on the faulty degradation data with 100% SNR applied to the training data for each reconstruction model.

100% SNR Data set	Accuracy (%) 1FNN	AE	VAE	LSTM
1	99.647	99.098	99.549	99.850
2	100	99.707	100	98.974
3	98.924	98.422	100	99.713
4	98.695	98.695	98.695	98.695
5	99.268	98.719	99.268	97.987
Avg. Accuracy	99.107	98.928	**99.502**	99.044

TABLE 6. Accuracy evaluation on the faulty degradation data with 90% SNR applied to the training data for each reconstruction model.

90% SNR Data set	Accuracy (%) 1FNN	AE	VAE	LSTM
1	68.421	99.098	99.398	99.850
2	99.413	99.707	99.853	100
3	99.785	99.857	99.857	99.713
4	98.434	98.695	98.695	98.695
5	97.108	95.168	99.854	98.243
Avg. Accuracy	92.632	98.505	**99.531**	99.300

TABLE 7. Accuracy evaluation on the faulty degradation data with 80% SNR applied to the training data for each reconstruction model.

80% SNR Data set	Accuracy (%) 1FNN	AE	VAE	LSTM
1	61.654	99.549	99.248	99.850
2	65.839	99.707	99.707	99.707
3	88.881	99.856	99.857	99.713
4	98.643	98.695	98.695	98.695
5	94.693	97.840	99.671	98.280
Avg. Accuracy	81.941	99.130	**99.435**	99.249

TABLE 8. Accuracy evaluation on the faulty degradation data with 70% SNR applied to the training data for each reconstruction model.

70% SNR Data set	Accuracy (%) 1FNN	AE	VAE	LSTM
1	52.030	99.549	99.399	99.850
2	53.812	99.853	99.707	98.094
3	77.475	99.713	99.641	99.713
4	98.747	98.695	98.695	98.695
5	90.629	97.987	99.524	98.353
Avg. Accuracy	74.539	99.159	**99.393**	98.941

TABLE 9. Average training time per epoch TT_{avg} for each reconstruction model.

Data set	TT_{avg} (seconds) 1FNN	AE	VAE	LSTM
1	1.0	2.8	1.6	48.9
2	0.9	2.8	1.7	47.8
3	2.1	5.4	3.2	100.0
4	3.2	8.3	4.5	143.0
5	4.4	11.7	6.9	201.9

provide consistently average accuracy performance in all situations. Thus, the deep models confirm robustness towards noisy real operational input data. The VAE proves to be the most reliable ED-structured reconstruction model since it provides a slightly better overall accuracy performance than the AE and LSTM. In addition to the accuracy, the average training time per epoch TT_{avg} needs to be considered for each reconstruction model. Table 9 shows TT_{avg} in the five data sets. Both AE and VAE provides satisfactory training

time. Compared to the AE and VAE, the LSTM provides extremely slow training time in all data sets. This is due to the internal cell structure of the LSTM, which results in a high amount of trainable parameters when structured as an ED. Thus, an ED structured LSTM is not recommended when it is trained in an unsupervised reconstruction-based manner. The total amount of trainable parameters for each reconstruction model is shown in Table 10.

TABLE 10. Total amount of trainable parameters for each reconstruction model.

Reconstruction model	Parameters
1FNN	420
AE	955
VAE	1010
LSTM	5796

As previously mentioned, each data set starts with different operational conditions and corresponding sensor measurements. The performance of the VAE range between 98.695% and 100% accuracy throughout the five data sets in the four different real-life situations. Thus, the proposed algorithm proves high independence towards varying operational conditions, which are expected in the maritime environment. Overall, the algorithm has proven to be highly suitable to automatically detect faults within maritime components. By combining the algorithm with fault isolation based on valuable human domain knowledge, it establishes performance strong enough to be included as a pure data-driven diagnostics approach in future end-to-end PHM system solutions where the a_{max} value could be used as the fault indicator.

V. CONCLUSION AND FUTURE WORK

This paper has investigated the possibilities for automatic fault detection within maritime components. Due to different environmental conditions with varying operational loads, and the common lack of fault labels in the maritime industry, maritime components require application-independent and intelligent fault detection algorithms in order to detect and report faults automatically. Therefore, an unsupervised reconstruction-based fault detection algorithm has been proposed in this paper. The algorithm has been applied to four different ED structured reconstruction models. The experiments were performed on five different data sets of real operational run-to-failure data of the same maritime component collected from a highly regarded industrial company. Each data set was subjected to a fault at an unknown time step. Different magnitudes of random white Gaussian noise have been applied to each data set in order to create four real-life situations. First, each reconstruction model is trained on normal operating data in an unsupervised manner. Then, the algorithm estimates an anomaly score function by calculating a reconstruction error at each time step in faulty degradation data. Finally, the algorithm detects a fault automatically by estimating the time step with the highest acceleration in the anomaly score function. The acceleration is chosen as the fault indicator due to latency in physical components. Thus, there is an expected time delay before a fault will result in large sensor measurements deviations. By this approach, the algorithm is considered more generic compared to previous user-specified threshold values. Additionally, the algorithm is independent of any prior domain knowledge of component degradation mechanics.

The algorithm achieved an average accuracy between 99.393% and 99.531% when compared to the true fault time step based on valuable human domain knowledge. Overall, the algorithm has both proven to be robust towards noisy real operational input data and independent of varying operational conditions. Thus, the algorithm, in combination with fault isolation based on valuable human domain knowledge, is highly suitable to be included as a pure data-driven diagnostics approach in future end-to-end PHM system solutions for maritime applications. In such a system, the value of the highest acceleration will be used as the fault indicator. Additionally, the corresponding time step to the fault indicator can be further used to create run-to-failure labels for any data-driven prognostics algorithm automatically. Future work will address these issues.

ACKNOWLEDGMENT

The authors would like to thank Digital Twins For Vessel Life Cycle Service (DigiTwin).

REFERENCES

[1] A. L. Ellefsen, V. Æsøy, S. Ushakov, and H. Zhang, "A comprehensive survey of prognostics and health management based on deep learning for autonomous ships," *IEEE Trans. Rel.*, to be published.
[2] K. E. Knutsen, G. Manno, and B. J. Vartdal, "Beyond condition monitoring in the maritime industry," *DNV GL Strategic Res. Inovation Position Paper*, 2014. [Online]. Available: https://www.researchgate.net/publication/263583976_Beyond_condition_monitoring_in_the_maritime_industry
[3] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics—A review of current paradigms and practices," *Int. J. Adv. Manuf. Technol.*, vol. 28, nos. 9–10, pp. 1012–1024, 2006.
[4] T. M. Allen, "Us navy analysis of submarine maintenance data and the development of age and reliability profiles," U.S. Navy SUBMEPP, Kittery, ME, USA, Tech. Rep., 2001.
[5] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Rel. Eng. Syst. Saf.*, vol. 183, pp. 240–251, Mar. 2019.
[6] B.-M. Batalden, P. Leikanger, and P. Wide, "Towards autonomous maritime operations," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Mar. 2017, pp. 1–6.
[7] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.
[8] E. Delechelle, J. Lemoine, and O. Niang, "Empirical mode decomposition: An analytical approach for sifting process," *IEEE Signal Process. Lett.*, vol. 12, no. 11, pp. 764–767, Nov. 2005.
[9] J. Gilles, "Empirical wavelet transform," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3999–4010, Aug. 2013.
[10] A. S. Zymaris, Ø. Å. Alnes, K. E. Knutsen, and N. M. P. Kakalis, "Towards a model-based condition assessment of complex marine machinery systems using systems engineering," in *Proc. 3rd Eur. Conf. Prognostics Health Manage. Soc.*, Bilbao, Spain, 2016, pp. 1–15.
[11] G. Wu, "Fault detection method for ship equipment based on BP neural network," in *Proc. Int. Conf. Robots Intell. Syst. (ICRIS)*, May 2018, pp. 556–559.
[12] A. Brandsæter, G. Manno, E. Vanem, and I. K. Glad, "An application of sensor-based anomaly detection in the maritime industry," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2016, pp. 1–8.

[13] C. Yang, J. Liu, Y. Zeng, and G. Xie, "Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model," *Renew. Energy*, vol. 133, pp. 433–441, Apr. 2019.
[14] *Case Western Reserve University Bearing Data Center*. Accessed: Dec. 4, 2018. [Online]. Available: https://csegroups.case.edu/bearingdatacenter/pages/download-data-file
[15] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.
[16] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," *ISA Trans.*, vol. 77, pp. 167–178, Jun. 2018.
[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[18] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals," *Mech. Syst. Signal Process.*, vols. 76–77, pp. 283–293, Aug. 2016.
[19] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.
[20] P. Malhotra *et al.*, "Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder," in *Proc. Workshop Mach. Learn. Prognostics Health Manage.*, 2016, pp. 1–10.
[21] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.
[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[23] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[24] D. P. Kingma and M. Welling. (2013). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114
[25] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, vol. 2. Sep. 1999, pp. 850–855.
[26] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.
[27] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3. Jul. 2000, pp. 189–194.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[29] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[30] *Eclipse Deeplearning4j Development Team, Deeplearning4j: Open-Source Distributed Deep Learning for the JVM Apache Software Foundation License 2.0*, 2018. [Online]. Available: http://deeplearning4j.org

ANDRÉ LISTOU ELLEFSEN received the master's degree in subsea technology from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2016. He is currently pursuing the Ph.D. degree with NTNU, Åalesund, Norway, as part of the Mechatronics Laboratory, Department of Ocean Operations and Civil Engineering. His current research interests include artificial intelligence, deep learning, decision support, predictive maintenance, and digital twins.

EMIL BJØRLYKHAUG is currently pursuing the Ph.D. degree with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology. His current research interests include robotic technologies for automating fish processing, tools that may facilitate industrial robots in performing more complex tasks, deep learning, and computer vision.

VILMAR ÆSØY graduated from the Norwegian University of Science and Technology (NTNU), in 1989, and continued his research on natural gas fueled marine engines at NTNU/MARINTEK, until 1997. He received the Ph.D. degree for his research on natural gas ignition and combustion through experimental investigations and numerical simulations, in 1996. From 1989 to 1997, he was engaged in several large R&D projects developing gas fueled engines and fuel injection systems for the diesel engine manufacturers, Wärtsilä, and Bergen Diesel, Roll-Royce. From 1998 to 2002, he was a R&D Manager for Rolls-Royce Marine Deck Machinery. Since 2002, he has been employed in teaching with the Aalesund University College, where he is also developing and teaching courses in marine product and systems design on bachelor's and master's level. In 2010, he received the green ship machinery professorship. His current research interest includes energy and environmental technology, with focus on combustion engines and the need for more environmental friendly and energy efficient systems.

HOUXIANG ZHANG (M'04–SM'12) received the Ph.D. degree in mechanical and electronic engineering, in 2003, and the Habilitation degree in informatics from the University of Hamburg, in 2011. Since 2004, he has been a Postdoctoral Fellow and a Senior Researcher with the Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, Institute of Technical Aspects of Multimodal Systems, University of Hamburg, Germany. He was with the Aalesund University College, in 2016. In 2011, he joined the Norwegian University of Science and Technology, Norway, where he is currently a Professor on mechatronics. His current research interests include biological robots and modular robotics, especially on biological locomotion control, and virtual prototyping in demanding marine operation. He has applied for and coordinated more than 20 projects supported by the Norwegian Research Council, German Research Council, and industry. In these areas, he has published over 160 journal and conference papers as author or co-author. He received four best paper awards and four finalist awards for Best Conference Paper at the International Conference on Robotics and Automation.