

A Comprehensive Survey of Prognostics and Health Management based on Deep Learning for Autonomous Ships

André Listou Ellefsen, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang, *Senior Member, IEEE*

Abstract—The maritime industry widely expects to have autonomous and semi-autonomous ships (autoships) in the near future. In order to operate and maintain complex and integrated systems in a safe, efficient and cost-beneficial manner, autoships will require intelligent Prognostics and Health Management (PHM) systems. Deep learning (DL) is a potential area for this development, as it is rapidly finding applications in a variety of domains, including self-driving cars, smartphones, vision systems, and more recently in PHM applications. This paper introduces and reviews four well-established DL techniques recently applied to various practical PHM problems. The purpose is to support creativity and provide inspiration towards PHM based on DL (PHMDL) in autoships and the maritime industry. This paper discusses benefits, challenges, suggestions, existing problems, and future research opportunities with respect to this significant new technology.

Index Terms—Autonomous ships, deep learning, maritime industry, prognostics and health management.

ABBREVIATIONS AND ACRONYMS

AE	Autoencoder
BB-RBM	Bernoulli-Bernoulli RBM
BP-NN	Back-propagation Neural Network
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CBM	Condition Based Maintenance
CD	Contrastive Divergence
CM	Condition Monitoring
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DBN	Deep Belief Network
DL	Deep Learning
DM	Decision Maker
DSSs	Decision Support Systems
DT	Digital Twin
ETTF	Estimation Of Time To Failure
FFT	Fast Fourier Transform

FNN	Feed-forward Neural Network
GB-RBM	Gaussian-Bernoulli RBM
GG-RBM	Gaussian-Gaussian RBM
GRU-LSTM	Gated Recurrent Unit LSTM
HMI	Human Machine Interface
HMM	Hidden Markov Model
IoT	Internet Of Things
LR	Logistic Regression
LSTM	Long-Short Term Memory
MD	Mahalanobis Distance
MFCC	Mel-frequency Cepstrum Coefficient
MLP	Multilayer Perceptron
NIST	National Institute Of Standards And Technology
NN	Neural Network
PCA	Principal Component Analysis
PHM	Prognostics And Health Management
PHM08	The 1st International Conference On Prognostics And Health Management In 2008
PHMDL	Prognostics And Health Management Based On Deep Learning
PM	Preventive Maintenance
RBM	Restricted Boltzmann Machines
RCM	Reliability Centered Maintenance
ReLU	Rectified-linear Unit
RF	Random Forest
RL	Reinforcement Learning
RM	Reactive Maintenance
RNNs	Recurrent Neural Networks
RUL	Remaining Useful Life
RVM	Relevance Vector Machine
SAE	Sparse Autoencoder
SOM	Self-organizing Maps
SVM	Support Vector Machine
TDNNs	Time-delay Neural Networks
TKEO	Teager-Kaiser Energy Operation
WPT	Wavelet Packet Transform

André Listou Ellefsen, Vilmar Æsøy, and Houxiang Zhang are with the Department of Ocean Operations and Civil Engineering, as part of the Mechatronics Laboratory, Norwegian University of Science and Technology, Aalesund, 6009 Norway, (e-mail: andre.ellefsen@ntnu.no; vilmar.aesoy@ntnu.no; hozh@ntnu.no).

Sergey Ushakov is with the Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, 7491 Norway, (e-mail: sergey.ushakov@ntnu.no).

Manuscript received February 14, 2018; revised July 30, 2018.

I. INTRODUCTION

AUTONOMOUS ships operate on the surface of the water entirely by themselves. Semi-autonomous ships require specialists and technicians who operate and monitor them from an onshore location through a satellite data link [1]. The industry as well as academics widely expect that autoships, a term that encompasses both, will increase the performance

of maritime operations, improving safety and profitability of industries that use them [2]. Many projects are undertaking to create such vessels [3]. Autoships will rely on complex and integrated systems to perform their main functions, and degradation of such systems during operation poses a serious threat to operations. Thus, they will require intelligent maintenance decision support systems (DSSs), which has begun to develop.

In general, maintenance in shipping follows either a reactive maintenance (RM) or preventive maintenance (PM) approach [4]. RM introduces high risks of unscheduled downtime, while PM provides relatively high reliability, but at unnecessary costs due to predetermined maintenance intervals [5]. PM also will not detect random failures, which are in fact the most common failure pattern in the maritime industry [6]. Thus, a more predictive maintenance approach is necessary in order to identify these kinds of failures. A predictive system will considerably increase the operation performance and drastically decrease unexpected system failures [7].

During the past decade, Prognostics and Health Management (PHM) has emerged as a promising engineering discipline for predictive maintenance decision support. It has enhanced potential to detect, isolate, and identify precursor and/or incipient faults of components and sub-components, monitors and predicts the progression of the fault, and provide decision-support or automation to develop maintenance schedules and asset management procedures [8]. Indeed, recent studies have confirmed that PHM is a positive alternative to traditional Condition Based Maintenance (CBM) and has therefore gained attention in both academia and the maritime industry [8]–[10]. However, DSSs with a high degree of decision automation have continue to fail frequently in industrial applications [11]. Accordingly, intelligent PHM systems require more precise and robust data-driven algorithms than systems to date have used.

PHM systems thus far have depended on traditional data-driven diagnostics and prognostics approaches [12]–[15] and signal processing techniques [16]. With the development of internet of things (IoT) and rise with big data, the traditional approaches confront several challenges when processing the increased volumes of data. Typically, they exploit human-engineered feature extraction methods, supervised machine learning algorithms, and shallow architectures. Thus, the traditional approaches are highly application-dependent, require large quantities of labeled training data, and are simply not designed for complex and large data sets in real-world applications [17], [18].

However, during the past decade with increased processing power and great progress in graphics processors [19], DL techniques have seen rapid developments. The areas of signal and information processing [20], speech recognition [21]–[23], images [24], [25], natural language processing [26], [27], and visual tracking [28] have seen significant improvements. DL techniques consist of several layers of non-linear processing stages. They utilize supervised or unsupervised learning strategies to automatically extract feature representations from raw input data. As a result, they are able to capture com-

plicated, hierarchically statistical patterns in more complex, high dimensional and noisy real-world data [29]. For this reason, DL techniques are the most promising area of research to overcome the limitations of traditional diagnostics and prognostics approaches [30]. Nonetheless, issues remain that make it difficult to apply DL techniques to practical PHM problems.

Autoships requires intelligent PHM systems that must be capable of providing reliable diagnostics and prognostics information in varying operating environments [31]. Additionally, lack of onboard crew members and the introduction of highly automated systems necessitate an end-to-end solution. DL techniques are less application-dependent than traditional machine learning algorithms because they are able to process raw and varying sorts of input data. Consequently, human-engineered feature extraction methods are not necessary. DL techniques therefore require minimal human input in the data processing stage and can be considered an end-to-end solution. Nevertheless, DL techniques are still normally applied to perform supervised classification and/or regression tasks within the PHM domain [32]–[34]. With respect to autoships and the maritime industry generally, the lack of fault labels and run-to-failure data of components and sub-components are major issues towards successful implementation of PHM systems based on current DL techniques [35].

This paper reviews and discusses both theoretical and practical issues regarding DL techniques. The broad PHM applications and extensive literature make it impossible for one review to embrace all the work in the field. This review aims to provide a summary of the most important advances in DL techniques recently applied to PHM suitable for autoships and the maritime industry. The important advances introduced in this paper mainly took place from 2013 to 2018. The current research status and issues, benefits, challenges, and future research opportunities will be discussed. Although many DL techniques can be used for PHM purposes, the focus nonetheless is on Autoencoder (AE), Convolutional Neural Network (CNN), Deep Belief Network (DBN) and Long-Short Term Memory (LSTM). This is primarily because they are well-established and show great promise for future work.

The overall organization of the paper is as follows. Section II introduces the necessary background on PHM and DL. Section III considers the main benefits in applying PHM based on DL in autoships, as well as the most important challenges that arise in the field. Section IV reviews DL applied to PHM in other applications suitable for autoships and the maritime industry. This section elaborates strengths and weaknesses in a more theoretical and practical understanding. To the best of the authors' knowledge, the use of intelligent PHM systems based on DL techniques in autoships have not yet been studied comprehensively. Thus, section IV will provide inspiration to obtain both knowledge and understanding. Section V provides discussions regarding suitable solutions for autoships, consisting of important open questions, existing problems, and future research opportunities. Finally, Section VI concludes the survey paper.

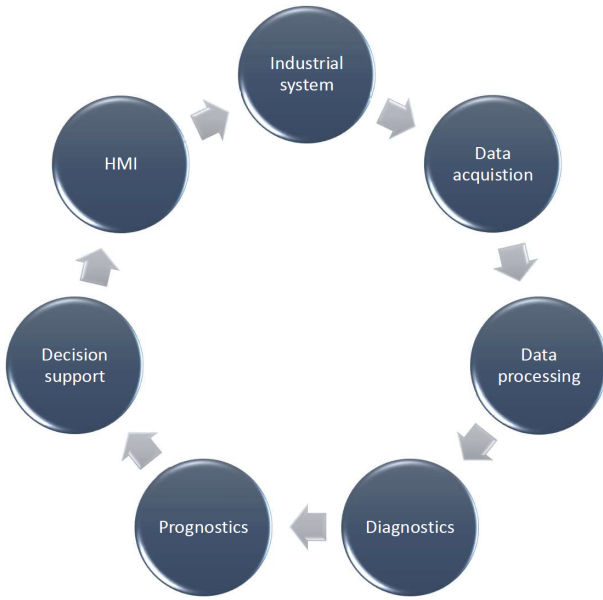


Fig. 1: CBM flowchart adopted from [44].

II. BACKGROUND: PROGNOSTICS AND HEALTH MANAGEMENT AND DEEP LEARNING

In this section, the necessary background on PHM and DL will be introduced. First, PHM is defined in general. Next, each step of PHM is explained and discussed. Finally, DL is presented with its promising aspects.

A. Prognostics and Health Management

PHM is an emerging engineering discipline that strives to decrease and ultimately eliminate inspections and time-based maintenance intervals [36]. This will be achieved through accurate condition monitoring (CM), precursor and/or incipient fault-detection, -isolation and -identification, and prediction of approaching failures. PHM amplifies and integrates the principles of both CBM and Reliability Centered Maintenance (RCM). It is designed to predict and protect the integrity of complex systems, components, and sub-components by avoiding unforeseen operational problems [37]. This creates a robust system to optimize maintenance decision making in order to increase the reliability and expected lifetime of industrial systems. Industrial systems such as the automotive industry [38], [39], the U.S Department of Defence [40], the aerospace and aviation industries [41], [42], and manufacturing systems [43] have recently integrated PHM with success.

PHM consists of seven steps initially defined from CBM [44]. Figure 1 illustrates the steps. The following subsections briefly discuss each step.

1) *Data acquisition and processing*: Data acquisition is the process of accumulating and storing raw sensor data related to the system condition. The data collected is usually categorized as CM data and event-data. CM data is the sensor

measurements associated with the system health, while event-data is the knowledge obtained from an event (e.g. what kind of failure did occur, when and where did the failure take place, who performed the maintenance procedure) [45]. Event-data provides useful information as to the performance of current features, as well as feedback in redesign or enhancement of features [44]. Thus, it is as important as CM data, although humans generally enter it manually, making it more fallible. An optimal maintenance system should automatically collect the event-data.

Data processing includes data cleaning and data analysis. Cleaning isolates potential human and/or sensor faults and eliminates data that reflects these errors. The data can be a value type, a waveform type, or a multidimensional type [44]. Waveform and multidimensional data may contain noise. Therefore, cleaning also generally includes methods like amplification, data compression, data validation, denoising, and filtering to enhance the signal-to-noise ratio [46]. Data analysis extracts condition indicators that represents incipient and/or precursor failures or faults. The main purpose of those features is to maximize diagnostics and prognostics accuracy in order to decrease false alarms. The literature has described processing techniques like wavelet transform, data denoising, and data smoothing [47], [48]. [46], [49] describes signal processing and feature extraction.

2) *Diagnostics and prognostics*: An effective PHM system includes diagnostics and prognostics approaches in order to provide ample and efficient decision support or automation. Diagnostics identify, localize, and determine the severity of an evolving fault condition [36]. It involves fault detection, fault isolation, and fault identification [50]. Fault detection, also called health/condition assessment [37], [45], compares sensor data with expected operational performance, that is, expected values of system parameters such as pressure, temperature, and vibration, to identify irregular operating conditions. Fault isolation involves pinpointing the component or sub-component that is degraded. Fault identification determines fault- type and dimension according to classes associated with specific values of measured signals [51]. Normally, this classification process uses a supervised classifier (e.g. machine learning algorithm) to classify various faults.

Prognostics predict the progression of faults, and hence, estimate the available time before a component or sub-component loses its operational ability, namely, before a failure [52]. Because the large uncertainties involved, researchers have called prognostics “the Achilles’ heel” of PHM [53], [54]. According to [55], the technical definition of prognostics is the estimation of time to failure (ETTF). However, in line with common usage in the literature, this paper uses the technical term remaining useful life (RUL). Any RUL estimation should include associated confidence intervals, which will indicate the window in which maintenance or repair must be conducted [53]. Such intervals add assurance of continuous operation in spite of the inherent uncertainty associated with the degradation process, human errors, and flaws in both the diagnostics and prognostics approach applied in the PHM system [56]. Maintenance decisions based on prognostics information should be grounded in confidence intervals instead of a particular RUL value. The

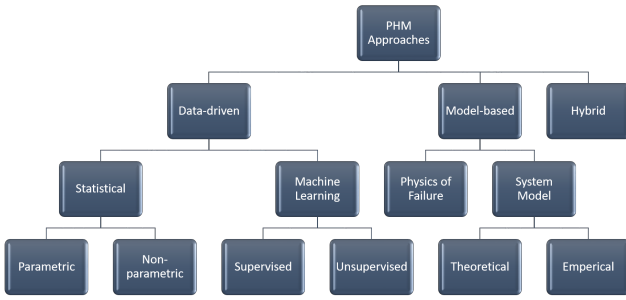


Fig. 2: A hierarchy of the three main diagnostics and prognostics approaches.

TABLE I: Recent PHM/CBM reviews based on traditional diagnostics and prognostics approaches (the years between 2006 and 2017).

Author	Refs.	Year	PHM application	Approaches
Tahan et al.	[60]	2017	Diagnostics and prognostics: Gas turbines	Data-driven Model-based Hybrid
Bailey et al.	[61]	2015	Diagnostics and prognostics: Engineering systems	Data-driven
An et al.	[62]	2015	Fatigue crack growth	Data-driven Model-based
Lee et al.	[8]	2014	Diagnostics and prognostics: Rotary machinery systems	Data-driven Model-based
Sikorska et al.	[56]	2010	Prognostics: Selection of RUL models	Data-driven Model-based
Vachtsevanos et al.	[50]	2006	Diagnostics: Book chapter	Data-driven Model-based
Vachtsevanos et al.	[53]	2006	Prognostics: Book chapter	Data-driven Model-based
Roemer et al.	[63]	2006	Prognostics: Engines	Data-driven Model-based
Jardine et al.	[44]	2006	Diagnostics and prognostics: Machinery systems	Data-driven

confidence intervals increase the reliability of the PHM system.

Successful prognostics depend on accurate diagnostics [11], [56], [57]. Diagnostics is necessary when prognostics fails and can prevent future failures of similar characteristics [44]. Even so, prognostics are considered more important than diagnostics to the ultimate goal of zero-downtime performance. This is because prognostics has the potential to prevent failures before they occur. Nevertheless, several challenges to successful implementation still exist. Thus the challenges, which [58] describes, should be addressed.

No common accepted prognostics methodology exists [43]. Figure 2 illustrates the three most common, however, which are Data-driven, Model-based, and Hybrid. The hybrid approach is a combination of data-driven and model-based approaches, aiming to utilize strengths of both approaches while avoiding their weaknesses [59]. Table I provides a summary of the most comprehensive PHM/CBM reviews regarding traditional diagnostics and prognostics approaches considered in this survey paper.

3) *Decision support and HMI*: The final object of a PHM system is to provide reliable decision support or automation

in order to enable effective maintenance scheduling. Decision support should assist a decision maker (DM), while decision automation uses software to provide entirely autonomous decisions [64]. However, according to [11], the output of today's industrial PHM systems usually constitutes decision support, as decision automation has not been integrated globally. Normally, inputs from human experts (application-dependent domain knowledge) and a DM who interprets the outputs compose the system [65]. Nevertheless, the human expert-generated input will fail when it encounters new conditions, that the knowledge base does not define. In addition, the most proficient DM has an insufficient cognitive capacity to analyze and understand large quantities of information [66]. Hence, decision support in the age of big data is subjected to uncertainties and does not always ensure good quality decisions. The literature provides several excellent reviews discussing this issue [11], [65], [66].

The advantages of a PHM system are highly connected to the decision-making based on the accumulation and understanding of diagnostics and prognostics information. Making the best decisions based on complex and large quantities of information is difficult [66]. However, advanced and deep signal processing and machine learning techniques are evolving rapidly [18]. These techniques provide automatic feature extraction and unsupervised learning procedures. Thus, such techniques minimize the human expert-generated input and have the potential to contribute to more intelligent PHM systems.

As [11] propose, the reliability of a fully autonomous (intelligent) PHM system needs to be greater than 99%. This level of reliability makes it possible for the PHM system to provide directions for maintenance procedures transferring directly from the system to the maintenance personnel, without the involvement of a DM. Another important aspect of autonomous PHM systems is that they must prove reliability in order to utilize the "black-box" approach. PHM systems with low reliability should enable user-access to the source code in order to promote understanding and trust in the system [11].

The human machine interface (HMI) is also an important perspective regarding understanding and trust since the screens and displays heavily affect how a DM or the maintenance personnel understand the PHM system. A web-based PHM system, in which the user interacts using a thin-client web browser, has several advantages. According to [67], this system is powerful to retrieve, analyze, and visualize structured data from high-dimensional databases. It can provide access to unstructured data and promote communication and decision making in distributed teams [67].

B. Deep Learning

In recent years, DL has turned into an extremely active sub-field of machine learning. DL and big data are probably the most significant trends in the fast-growing digital world [19]. According to the National Institute of Standards and Technology (NIST) [68], big data is "the large amount of data in the networked, digitized, sensor-laden, information-driven world." NIST goes on to note that big data "can overwhelm traditional

technical approaches and the growth of data is outpacing scientific and technological advances in data analytics.” Big data forces a dramatic paradigm shift towards data-driven approaches and discoveries within scientific research. [18], [19] provide exceptional reviews regarding the relationship between big data and DL. In addition, machine learning is now a major technical field of the signal processing society [69].

The expansion of big data and IoT tends to make traditional machine learning algorithms like Hidden Markov Model (HMM) [70], Support Vector Machine (SVM) [71], and Neural Network (NN) with one hidden layer [61] vague, creating several challenges. First, traditional algorithms utilize shallow architectures, with only two stages of data-dependent computation elements. This means that shallow architectures contain only a small number of non-linear processing transformations. Previous analyzes of the boolean circuit complexity theory literature [72], [73], have revealed that shallow circuits require exponentially more elements than deeper circuits [74]. According to [75], this applies also to shallow and deep architectures in machine learning algorithms when they are required to process highly non-linear and varying functions. Consider the parity function with d inputs. Gaussian SVM requires 2^d parameters, NN with one hidden layer requires d^2 parameters, while a deep architecture requires d parameters with $\log_2 d$ layers. As a result, shallow architectures are inefficient due to the increased number of computational elements (e.g. hidden units), which require many examples [75]. Consequently, Gaussian SVM and NN with one hidden layer suffer from a decreased capacity to process more complex and high-dimensional real-world data with accuracy [16], [29]. Second, most traditional machine learning algorithms use supervised learning procedures. This means they require large quantities of high-quality labeled training data. However, in real-world applications large amounts of the data are unlabeled, and according to [76], most data collected in the age of big data is heterogeneous and unstructured. Finally, traditional machine learning algorithms lack the ability to extract and organize the discriminative information from the data [77].

Over 60 years ago, Richard Bellman declared that learning complexity grows exponentially with the linear increase in the dimensionality of the data [17]. He named this phenomenon “The curse of dimensionality” [78]. During the last decades, researchers have applied human-engineered feature extraction methods to the data processing stage to reduce the dimensionality of the data so that traditional machine learning algorithms can process it [17]. As a consequence, much of the actual work in using traditional machine learning algorithms goes into the design of the features because the performance of the algorithms relies heavily on the chosen method [77]. Hence, human-engineered feature extraction methods require precise engineering and substantial domain expertise, and the applied algorithm becomes highly application-dependent [79].

Recent discoveries in neuroscience, increases in computing power and an explosion of digital data have been the central motivational factors for the emergence of DL. The discoveries in [80], [81] clarify that the neocortex allows signals to propagate through a complex hierarchy of units. In time, these units will learn to represent observations based on the regularities

they express [17]. DL focus on similar characteristics as the neocortex. Actually, DL is a three-decade-old technique and a renewal of the even older NNs [82].

Great advances and innovations have been achieved in DL since 2006 [75], [83]–[85]. At that time researchers, gathered by the Canadian Institute for Advanced Research, introduced unsupervised learning strategies that could extract features without requiring labeled training data, that is, capture statistical patterns in the observed data [79], [86]. Unsupervised DL techniques introduce hierarchical structures to automatically extract important features, from low-level input observations to high-level abstractions, using unsupervised pre-training where all layers are initialized. After precise fine-tuning, the highest level abstract features will normally be the input to a supervised classifier or regressor, minimizing the global training requirement [87].

More specifically, a DL technique is a multilayer stack of non-linear processing stages to compactly (with few parameters) represent highly non-linear and varying functions [75]. Most of the stages are subjected to supervised or unsupervised learning and compute non-linear input-output mappings. Each stage modifies its input in order to increase both the invariance and selectivity of the representation [79]. Consequently, DL techniques can often capture complex, hierarchically statistical patterns in unstructured, high dimensional, and noisy real-world data [29]. With multiple non-linear layers, DL techniques make possible extremely involved functions of its inputs that, at the same, are time-sensitive to small details and insensitive to large irrelevant variations [79].

In the past decade, DL techniques have shown fast advancements with notable impacts on signal and information processing [20], beaten records in image recognition [25] and speech recognition [22], and outperformed traditional machine learning algorithms in natural language understanding [27], and diagnostics and prognostics purposes [32], [88]. In addition, as [19] state, DL is going to play an important role in prediction tasks due to increased processing power and the advances in graphics processors. A great historical survey of DL is given in [89]. It summarizes both current work and work from the previous millennium, including the history of supervised learning and back-propagation.

III. AUTONOMOUS SHIPS: BENEFITS AND CHALLENGES IN APPLYING PHM BASED ON DL

Only three years ago, most people considered autoships as a futuristic fantasy [3]. Today, however, this preconception has changed drastically. In fact, autoships will almost certainly be in commercial use by the end of this decade [3]. The first vessels will require a few crew members, however, at least to operate in challenging maritime areas. The transition to totally human-free autoships will likely take place gradually over a period of a few decades [3].

According to [3] and [31], securing regulatory approval, support from the industry, and public acceptance for autoships requires evidence they are at least as safe as traditional ships used for similar operational tasks. As they will ultimately, have

no maintenance personnel on board ready to perform unsystematic maintenance, safety critical systems and components must be more reliable than on traditional ships.

Autoships will transfer real-time diagnostics and prognostics information to shore to permit analysis and prioritization of issues of critical systems and components. Today's maritime maintenance procedures, by contrast, typically follow an RM or PM approach [4]. RM can be described as post-failure repair of components or sub-components, while PM involves predetermined maintenance intervals based on constant intervals, age-based or imperfect maintenance [5]. Traditional ships tend to rely heavily on onboard maintenance personnel since it is less costly to conduct RM and/or PM approaches while still at sea [31].

RM would create large and unnecessary costs when critical system/component failures occur during operation of autoships. Both the process of dispatching maintenance personnel while the autoship is still at sea and the process of guiding the vessel back to shore in order to perform repairs would create random and unplanned downtime, compromising efficiency. On the other hand, the constant and experience based maintenance intervals utilized in PM could be scheduled around predetermined port of calls. This will, of course, provide high reliability, but it involves unneeded maintenance inspections and procedures of completely functional systems. It also might not prevent the random need for maintenance involved in RM, since random failures are the most common type in the maritime environment [6]. The need for predictive maintenance approaches, such as intelligent PHM systems, is clear.

Based on the background information and brief discussion in Section II, it is obvious that DL techniques have the potential to overcome the limitations of traditional machine learning algorithms applied to diagnostics and prognostics purposes. For that reason, DL techniques are highly suitable to be applied in intelligent PHM systems. The next step in this survey paper is to introduce and discuss benefits and challenges in applying Prognostics and Health Management based on deep learning (PHMDL) in autoships.

A. Benefits

- Normally, critical systems on traditional ships are over-engineered by built-in redundancy. In this way, traditional ships complete their operational tasks even if a serious functional failure occurs. This design philosophy is related to historical inaccessibility to shore [90]. However, Inmarsat and Telenor have recently launched the data transfer satellites Inmarsat-5 and Thor 7, respectively, which will provide high-speed broadband connections to ships at sea [1]. This will enable new design philosophies, including online PHMDL systems, as alternatives to the legacy redundancy policy. Real-time diagnostics and prognostics of components and sub-components in which online PHMDL systems are referred to an onboard system that links to shore will make it possible to contribute the most efficient operating conditions possible, and enable future autoships without onboard maintenance personnel [31].

- The ultimate goal of a PHMDL system is to achieve zero-downtime performance. Real-time and reliable RUL estimations, with associated confidence intervals, of different components and sub-components, will have an enormous impact on the maintenance procedure and safety concept on autoships. When the RUL of a faulty component is estimated, the maintenance procedure can be scheduled to the next appropriate port of call, or if necessary, dispatching maintenance personnel before a failure occurs when the autoship is still in operation [1]. This will significantly increase the operational performance, and at the same time, drastically decrease unexpected system failures. In addition, reliable estimations will provide trust in safe behavior in offshore activities [7].
- According to [1], the insurance company Allianz reported in 2012 that between 75% and 96% of marine accidents are a result of human errors. This is mainly a result of human exhaustion, but also because today's maritime activities require humans both to manage planned operational activities and make complicated decisions based on the overall system conditions [7]. Autoships will reduce both the number of crew members and the influence of human DMs due to increased autonomous and intelligent operational planning and decision making. In this way, autoships will have the potential to decrease human errors and the risk of injury to crew members [31]. PHMDL systems have great potential to contribute to this human error reduction since these systems are less dependent on prior knowledge and human influence.

B. Challenges

- Autoships requires adaptation and integration within the functioning of a business of an organization, and hence, significant changes in the organizational culture [4]. The introduction of autoships also involves confidence and trust in "black-box" systems, such as a PHMDL system. These systems are intelligent in that they transfer directions for future maintenance procedures directly from the autoship to the maintenance team on shore. In order to act as a fully autonomous and intelligent system, the PHMDL system must adapt to the varying operational and environmental conditions that occur in the harsh maritime environment [35].
- A further concern is the continuous flow of data to shore. Autoships depend on heavily integrated and complex systems to deliver their main functions. As a result, the associated flow of sensor data becomes massive, high-dimensional, heterogeneous, and unstructured. The PHMDL system will have to provide automatic pre-processing and dimensionality reduction schemes. This massive flow of data also presents a cybersecurity challenge, as hackers would threaten safe maritime operations [1].
- A great challenge in the maritime industry is the lack of run-to-failure data of components and sub-components [35]. Traditional ships are often application-designed and unique, or batch-produced in two to ten

vessel series [4]. These short series creates a slow accumulation of relevant failure data compared to, for instance, the aviation industry that produces hundreds of the same aircraft in series [4]. In addition, traditional ships are typically equipped with components from several different manufacturers [91]. The resulting diversity of uncoordinated monitoring systems increases the complexity of the failure data. With respect to the introduction of autoships and PHMDL systems, it would be advantageous to build extensive databases regarding run-to-failure data of critical and relevant components and sub-components. This could be realized if stakeholders agreed to cooperate to share data.

C. Summary

Reliable and real-time diagnostics and prognostics in autoships have the potential to improve efficiency, maintenance procedures, and safety aspects. Based on the above-mentioned challenges, such as varying operational and environmental conditions and massive data flows, DL techniques will be superior to the combination of human-engineered feature extraction methods and traditional machine learning algorithms. This is because DL techniques utilize unsupervised learning procedures to automatically extract key features and reduce the dimensionality of raw unlabeled input data. Accordingly, DL techniques do not require human-engineered feature extraction methods, such as Mel-frequency Cepstrum Coefficient (MFCC) or wavelet transform, in the data processing stage. This means that the diagnostics and prognostics accuracy of a PHMDL system is less application-dependent. For that reason, PHMDL systems will have the potential to perform diagnostics and prognostics under different environmental and operational conditions. However, DL techniques are usually used to perform supervised classification and/or regression tasks. For that reason, available run-to-failure databases would be advantageous. The next section reviews recent PHMDL applications. This is to fully elaborate strengths and weaknesses in a more theoretical and practical understanding.

IV. APPLICATIONS OF DEEP LEARNING TO PROGNOSTICS AND HEALTH MANAGEMENT

In recent years, DL has emerged as an innovative and encouraging research field for PHM [30]. This section introduces and reviews well-established DL techniques like Autoencoder (AE), Convolutional Neural Network (CNN), Deep Belief Network (DBN) and Long-Short Term Memory (LSTM) based on applications to PHM in the recent five years. This information will support the need for creativity and inspiration in producing PHMDL possibilities for autoships.

A. Deep Belief Network

1) *Introduction:* In 2006, Hinton et al. [83], introduced a greedy layer-wise unsupervised learning algorithm. This was the first valid algorithm for training fully-connected deep architectures, and hence, marked the starting point for notable progress in DL. The algorithm was originally introduced

for DBNs and improved previous optimization problems of training deep architectures by initializing the weights in a region near a good local minimum [75]. The algorithm makes it possible to automatically learn internal representations of data. These internal representations are high-level abstractions of the input and allow a network to produce complex input-output mappings directly from data [87]. In this way, the algorithm is, in theory, not dependent on human-engineered features in the data processing stage.

The fundamental ideas of the algorithm are as follow [75], [87];

- 1) Pre-train one layer at a time in a greedy way. In other words, layer n is kept fixed while the $n - th$ layer is trained using the output of n as the input.
- 2) Perform unsupervised learning at each layer in order to maintain information from the input.
- 3) Fine-tune the whole network with respect to the global training requirement.

DBNs consists of several layers of Restricted Boltzmann Machines (RBMs) [92], and normally some additional layers to conduct e.g. classification or regression tasks.

RBMs [29], [75], [86], [93], [94] are probabilistic generative models that learn a joint probability distribution from unlabeled training data. They are a special type of Markov random fields, typically with Bernoulli or Gaussian stochastic visible units, v , in a single input layer and Bernoulli stochastic hidden units, h , in a single hidden layer. Normally, as shown in Figure 3, the visible and hidden units are fully connected with bias vectors, b and c , respectively, and weight matrix, w . In addition, units in the same layer have zero connections. Consequently, RBMs can be defined as symmetrical bipartite graphs. The hidden layer in the first RBM will serve as the input layer for the second RBM.

The Bernoulli-Bernoulli RBM (BB-RBM) is the binary version of RBMs. It is an energy-based model with the joint probability distribution specified by its energy function [93]:

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (1)$$

The energy function is given by:

$$E(v, h) = - \sum_{i=1}^V b_i v_i - \sum_{j=1}^H c_j h_j - \sum_{i=1}^V \sum_{j=1}^H w_{ij} v_i h_j \quad (2)$$

where w_{ij} represents the weight between the binary states of visible unit v_i and hidden unit h_j , b_i and c_j denotes the bias terms, while V and H indicates the numbers of visible and hidden units, respectively. The partition function, Z , is given by summing all possible combinations of visible and hidden vectors. It ensures that the distribution is normalized:

$$Z = \sum_v \sum_h e^{-E(v, h)} \quad (3)$$

Due to the fact that RBMs are symmetrical bipartite graphs, the conditional probabilities $p(v|h)$ and $p(h|v)$ are factorial, and can be efficiently calculated as (see full derivation in [86],

[93]):

$$P(v_i = 1|h) = \sigma\left(b_i + \sum_{j=1}^H w_{ij}h_j\right) \quad (4)$$

$$P(h_j = 1|v) = \sigma\left(c_j + \sum_{i=1}^V w_{ij}v_i\right) \quad (5)$$

where σ is the activation function. The logistic sigmoid function $\frac{1}{1+e^{-x}}$ is a usual choice [29].

For real-value data applications, Gaussian-Bernoulli RBM (GB-RBM) is normally used as the initial RBM to convert real-valued stochastic variables to binary stochastic variables [95], [96]. The second RBM can then be a BB-RBM with a rectified-linear unit (ReLU) [97] transformation as the activation function for further processing. The energy function for GB-RBM is given by [93]:

$$E(v, h) = \sum_{i=1}^V \frac{(v_i - b_i)^2}{2\gamma_i^2} - \sum_{j=1}^H c_j h_j - \sum_{i=1}^V \sum_{j=1}^H \frac{v_i}{\gamma_i} h_j w_{ij} \quad (6)$$

where γ_i is the standard deviation of visible unit v_i . The corresponding conditional probabilities are expressed by:

$$P(v_i = x|h) = \frac{1}{\gamma_i \sqrt{2\pi}} \exp\left(-\frac{(x - b_i - \gamma_i \sum_{j=1}^H h_j w_{ij})^2}{2\gamma_i^2}\right) \quad (7)$$

$$P(h_j = 1|v) = \sigma\left(c_j + \sum_{i=1}^V w_{ij} \frac{v_i}{\gamma_i}\right) \quad (8)$$

where x is a real number. In practice, to make the model implementation of GB-RBM more simple, the input data should be normalized to have zero mean and unit variance [93]. It should be noted that a study conducted in 2010 has shown that noisy ReLUs works better than Bernoulli stochastic units in RBMs hidden layer [98].

The contrastive divergence (CD) [99] update rule is used to train RBMs:

$$\Delta w_{ij} = \epsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right) \quad (9)$$

where ϵ is the learning rate and $\langle * \rangle$ denotes expectations under the distribution. The first expectation is with respect to the data distribution and samples visible units based on hidden units (Equation 7). The second expectation has to do with the reconstructed input data distribution, generated by Gibbs sampling, which samples hidden units based on visible units (Equation 8). The reconstruction part of RBM training makes it a generative model since it guesses the probability distribution of the original input. The weights between the input layer and the hidden layer are then updated using Equation 9. This process will repeat until the parameters converge, that is, the hidden layer is able to approximate the input layer. Thus, RBMs model data distribution using hidden units without the use of label knowledge. After the RBM training process, the parameters are presented to the DBN. In the end, the whole DBN architecture is fine-tuned using supervised back-propagation with a much smaller data set of labeled training

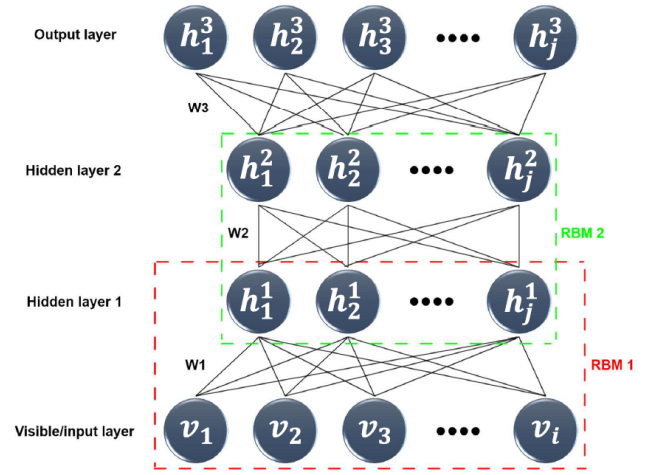


Fig. 3: A simple DBN representation with two hidden layers. Each visible and hidden unit are essentially nodes where calculations take place.

data [100]. It should be noted that the training process of RBMs is crucial in applying DBNs successfully to practical problems. [93] includes a practical training guide by the machine learning group at the University of Toronto.

2) *Recent applications to PHM*: DBNs are capable of providing automatic feature extraction from unlabeled training data and of performing supervised classification or regression tasks by adding one or more additional layers. These properties are well suited for PHM systems. The paragraphs below review applications of DBNs to PHM in the years between 2013 and 2017.

Regardless of the well-proven applicability of traditional data-driven diagnostic approaches, CM through multiple sensors remains one of the major difficulties to be addressed in the areas of classification and health diagnostics [14]. The reason for this is that the complexity of the classification model increases with multiple sensors and heterogeneity of sensor signals, and hence, the data becomes highly dimensional. Tamilselvan et al. [32] proposed a novel DBN approach for use in multi-sensor health diagnostics state classification. The proposed approach was demonstrated with the publicly available data set from the competition held at the 1st international conference on Prognostics and Health Management in 2008 (PHM08) [101]. The data set was produced by the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), provided by NASA [102]. In addition, two case studies were conducted for further demonstration. The DBN provided better classification performance compared with four traditional data-driven diagnostic algorithms; SVM, back-propagation Neural Network (BP-NN), self-organizing maps (SOM), and Mahalanobis distance (MD). However, in this study, labeled training data was used for different health states. Thus, this study did not investigate DBNs full potential for automatic feature extraction of unlabeled training data.

Tran et al. [103] also utilized DBN as the diagnostics

approach. The proposed approach was validated with signals from a two-stage reciprocating air compressor under different valve conditions. The DBN was used to classify faults and showed superior performance compared to traditional data-driven diagnostic algorithms, such as Relevance Vector Machine (RVM) and BP-NN. However, in this study, the DBN approach was only used for classification, and hence, it did not examine the automatic feature extraction of unlabeled training data aspect. On the other hand, the Teager-Kaiser energy operation (TKEO) and wavelet transform were used as the feature extraction methods.

Nevertheless, the automatic feature extraction aspect was heavily explored in [100] and [104]. Yang Fu et al. [100] demonstrated that the performance of the traditional data-driven diagnostics algorithms SVM, Multilayer perceptron (MLP) and k-means, strongly depends on the human-engineered feature method selected. Three kinds of features are included in this comparison: raw vibration data with normalization, MFCC, and wavelet method. In this study, the DBN consistently presented wonderful classification performance in all three features. This shows that DBN is a promising automatic feature extraction tool to be used on raw signals without too much data preparation. [104] is a similar study. Li et al. utilized a DBN as a statistical feature learning tool for bearing and gearbox systems in time, frequency, and time-frequency domains. The proposed approach indicated better classification results compared to SVM and a single layer of GB-RBM.

Various traditional data-driven prognostics approaches have been proposed for different applications. Normally, they involve human-engineered feature extraction methods in combination with a single traditional machine learning algorithm. As a consequence, these traditional approaches can hardly maintain good generalization performance and adapt to different prognostics applications. However, Zhang et al. [105] proposed a multiobjective DBN ensemble (MODBNE) method. MODBNE applies a multiobjective evolutionary ensemble learning framework combined with the DBN training process. In this way, the proposed method is able to create multiple DBNs of varying accuracy and diversity, which in fact are two conflicting objectives. The evolved DBNs are then combined to perform RUL estimations. The proposed method was evaluated by the publicly available C-MAPSS data set, the turbofan engine degradation simulation data set [106] produced by the C-MAPSS and provided by NASA. The big difference between the PHM08 data set and the C-MAPSS data set is that only the latter provides true RUL targets. The proposed approach was compared with several traditional data-driven algorithms.

Deutsch et al. [107] introduced a deep architecture for RUL estimations of rotating components using vibration sensors. The proposed approach combines the automatic feature learning ability of DBN, and the predictive power of feed-forward Neural Network (FNN). The approach is termed DBN-FNN and has the opportunity to either utilize processed vibration features or extract features from the vibration data to estimate RUL. The RUL estimation includes confidence boundaries obtained by the re-sampling technique jackknife. The proposed approach overcomes the limitations of traditional data-

driven approaches by performing automatic feature extraction and RUL estimations without human interference or prior knowledge. Thus, the DBN-FNN approach confirms potential towards the application of autoshops.

To enable accurate RUL estimations, feature extraction is a vital step. Liao et al. [108] proposed an enhanced single layer RBM with a novel regularization term to automatically generate features that are suitable for RUL estimations. The main advantage of the regularization term is that it tries to maximize the trend of the output features. Consequently, it has the potential to make better representations of the degradation patterns in the system. The proposed approach is compared with traditional RBM and principal component analysis (PCA). This method has the opportunity to be extended to a DBN by stacking multiple enhanced RBMs. However, the proposed approach is based on a Gaussian-Gaussian RBM (GG-RBM). According to [75], DBNs containing only Gaussian units will only be able to model Gaussian data. In addition, the mean-field propagation through a Gaussian unit gives rise to a purely linear transformation. Hence, the internal representations would be completely linear. In other words, Gaussian transformations do not work well on RBMs' hidden layers.

Jiang et al. [109] proposed a deep architecture involving a DBN and a non-linear kernel-based parallel evolutionary SVM. The objective was to predict evolution states of complex systems in classification tasks. The goal of the algorithm is to predict class labels of test data without any label information. In two case studies, the proposed approach outperformed both SVM and the traditional DBN.

DBNs have also been successfully and heavily applied in time series forecasting [110]–[112].

B. Autoencoder

1) *Introduction:* The greedy layer-wise unsupervised learning algorithm introduced by Hinton et al. [83] and further analyzed by Bengio et al. [75], can be applied not only to RBMs but also to AEs. An original AE [29], [75], [77], [86], [94] is an FNN, normally with one hidden layer, trained to reproduce its input to its output by forcing the computations to flow through a “bottleneck” representation [74], namely, dimensionality reduction. The hidden layer, h , describes a code used to represent the input, x . The network consists of two parts: an encoder function $h = f_{\theta_e}(x)$ and a decoder function that produces a reconstruction $r = g_{\theta_d}(h)$. If the AE learns the identity function, $g_{\theta_d}(f_{\theta_e}(x)) = x$, it will not be effective to extract meaningful features [113]. However, modern variations of the original AE are normally restricted to only copy input that is similar to the training data. Consequently, the AE is forced to prioritize which characteristics of the input it should copy. Thus, it often learns useful features of the data, and at the same time, filters useless information [94]. In addition, since the input vector is transformed into a lower dimension, the efficiency of the learning process can be increased [20]. Figure 4 shows a simple AE. It should be noted that AE is also called autoassociator in the literature.

The visible units, x , in the input layer, the hidden units, h , in the hidden layer, and the reconstruction units, r , in the

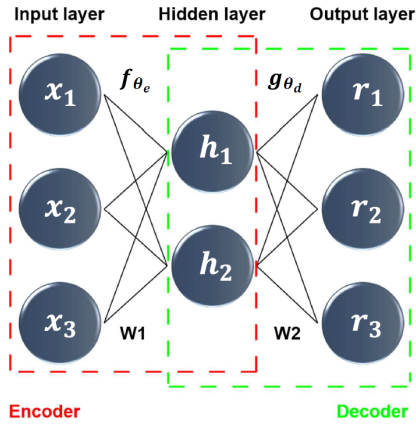


Fig. 4: Simple structure of an autoencoder. Three nodes in the input and output layer and two nodes in the hidden layer (bottleneck).

output layer are connected with weight matrices, w^1 and w^2 . The hidden layer and the output layer have bias vectors b and c , respectively. As opposed to the parameterization of RBMs (single weight matrix), the AE framework permits a different matrix in the encoder, $\theta_e = \{w^1, b\}$, and in the decoder, $\theta_d = \{w^2, c\}$. Nevertheless, in practice it is common to use tied weights, $w^2 = (w^1)^T$, [77]. This provides the parameterizations identical and serves as a regularizer since it constrains the parameter space [29]. θ_e and θ_d are learned concurrently on the task of reconstruction and compared to the original input in order to obtain the lowest possible reconstruction error $L(x, r)$ [77]:

$$J_{AE}(\theta_e, \theta_d) = \sum L(x, g_{\theta_d}(f_{\theta_e}(x))) \quad (10)$$

where L is a loss function such as the squared error $L(x, r) = \|x - r\|^2$. Basic AE training consists in finding values of the weights and biases in order to minimize $L(x, r)$. The most normal encoder and decoder function are affine (feed-forward) mappings, optionally followed by a non-linearity [77]:

$$f_{\theta_e}(x) = \sigma_f(b_j + \sum_i w_{ji}^1 x_i) \quad (11)$$

$$g_{\theta_d}(h) = \sigma_g(c_i + \sum_j w_{ij}^2 h_j) \quad (12)$$

where σ_f and σ_g are the encoder and decoder activation functions. It should be noted that the choice of activation and loss function depends on the input domain range and character. AEs can be stacked, like the RBM, to form a deep architecture. Thus, the training procedure is equivalent to the one introduced for DBNs [83], but using AEs rather than RBMs. [74] presents a comparative study regarding AEs and RBMs. This study suggests that DBNs have a slight edge over stacked AEs. According to [86], this is probably because CD is closer to the log-likelihood gradient than the reconstruction error gradient. There exist several modern variations of the

original AE in the literature. In the following subsections, the Denoising Autoencoder (DAE) and the Sparse Autoencoder (SAE) will be introduced in relation to recent applications to PHM.

2) *Denoising Autoencoder*: Vincent et al. [114], [115] proposed the DAE in 2008. This extension of the original AE was designed to learn more robust representations in a deep architecture. DAEs are trained with corrupted data, \tilde{x} , by adding noise into the training data through the stochastic corruption process, $\tilde{x} \sim q(\tilde{x}|x)$. The robustness is achieved when the DAE reconstructs the clean version of the training data through the training process. The objective function for optimization in the DAE is given by:

$$J_{DAE}(\theta_e, \theta_d) = \sum IE_{q(\tilde{x}|x)}[L(x, g_{\theta_d}(f_{\theta_e}(\tilde{x})))] \quad (13)$$

where $IE_{q(\tilde{x}|x)}[*]$ represents the average value over \tilde{x} drawn from the stochastic corruption process $\tilde{x} \sim q(\tilde{x}|x)$ [77]. The major difference between AE and DAE is that, r is a deterministic function of \tilde{x} rather than of x . Hence, DAE must undo the corruption instead of simply copy the input [94]. DAEs can also be stacked to form a deep architecture. The greedy layer-wise training strategy is identical to the strategy for the original AE and RBM. It should be noted that the stochastic input corruption process is only applied in the training procedure in order to learn more robust and valuable representations [116]. Thereafter, the reconstructed clean version is used as the input to the next layer. Various corruption processes like additive Gaussian noise, salt and pepper noise, and masking noise can be considered [115].

3) *Sparse Autoencoder*: In 2006, Ranzato et al. [117] proposed the learning algorithm for sparse representations. SAE is also an extension of the original AE that aims to use sparse representations in order to produce a simple understanding of the input data by extracting the hidden structure of the data [20]. The training criterion involves a sparsity penalty term, $\Omega(h)$, on the hidden layer, h , in addition to the reconstruction error $L(x, r)$ [94]:

$$J_{SAE}(\theta_e, \theta_d) = \sum L(x, g_{\theta_d}(f_{\theta_e}(x))) + \Omega(h) \quad (14)$$

The sparsity penalty term $\Omega(h)$ is added to the objective function of the original AE (Equation 10) in order to constrain the learned features. It controls the number of active neurons in the hidden layer, h . A neuron is considered active if the output is close to 1, and inactive otherwise [113]. The sparsity penalty term is defined as:

$$\Omega(h) = \beta \sum_{j=1}^H KL(\rho|\rho_j) \quad (15)$$

where β controls the weight, H is the number of neurons in the hidden layer and $KL[*]$ is the Kullback-Leibler divergence [118]:

$$KL(\rho|\rho_j) = \rho \log \frac{\rho}{\rho_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \rho_j} \quad (16)$$

where ρ is a hyperparameter (typically close to zero, e.g. $\rho = 0.05$ [77]) and ρ_j is the average activation of the hidden unit j . As seen from Equation 16, the sparsity penalty term is zero if $\rho_j = \rho$. Thus, the sparsity penalty term will penalize ρ_j if it deviates considerably from ρ . In other words, it promotes partial activations of each hidden unit as specified by ρ [29]. By only activating a few hidden nodes at the same time, the system robustness is improved. According to [94], SAEs are typically used to learn features for classification tasks due to its enhanced performance. After stacking several SAEs to form a deep architecture, the greedy layer-wise training procedure is also here identical as for the DAE, original AE, and RBM.

4) *Recent Applications to PHM*: AEs are, as with DBN, capable of providing automatic feature extraction from unlabeled training data, and in addition, performing supervised classification or regression tasks by adding one or more additional layers. The modern versions of AE, DAE, and SAE seem particularly promising for PHM applications and autoships. DAE is robust to noise and SAE has the potential to increase the robustness of the system and the performance of classification tasks. In the paragraphs below, applications of AEs, DAEs, and SAEs to PHM are reviewed in the years between 2014 and 2017.

Feature extraction is a crucial part of a PHM system because it determines the performance of both diagnostics and prognostics. Lu et al. [119] proposed a stacked AE, containing two hidden layers, as the feature extraction method for rolling bearing fault diagnostics. The results indicated that the second hidden layer provided more precise and identifiable features than the first hidden layer and the raw features in the visible layer. Thus, a stacked AE is a promising tool to extract features from bearing signal data.

Typically, in large industrial systems, the data is derived from several platforms that could potentially involve different data types. Based on this, Ma et al. [120] proposed an architecture with multiple input modalities applied to fault diagnostics. The proposed approach is using RBMs to obtain a unified representation for both images and structured data. Then, the unified representation is the input to a stacked AE in order to reconstruct the images and the structured data to obtain abstract features and remove useless information. In the final layer, a supervised linear classifier is added to classify the learned features and fine-tune the whole network. Comparing the proposed approach with BP-NN showed lower misjudgment rate for both normal and fault conditions.

Jia et al. [121] proposed a novel intelligent fault diagnostics method for rotary machinery in order to overcome the limitations of traditional diagnostic approaches. The main limitations highlighted in this study are shallow architectures and the requirement of application-dependent human-engineered feature extraction methods in the data processing stage. To overcome these limitations, the proposed method utilized a stacked AE to adaptively extract fault characteristics (features) from measured signals in the frequency domain, and automatically classify machinery health conditions. The proposed method was validated using rolling element bearing- and planetary gearbox data sets, and finally, compared with the traditional BP-NN. The results indicated that the proposed

method overcomes the above-mentioned limitations.

Xia et al. [116] also addresses the limitations of traditional diagnostics approaches, specifically the need for prior knowledge of features and the requirement of large quantities of labeled condition data as the main limitations. In addition, most traditional approaches need to be rebuilt or retrained in order to diagnose new conditions. This procedure is both computationally expensive and time-consuming. To overcome these limitations, the proposed method in this study utilized a stacked DAE with a softmax regression classifier in the output layer. The results indicated that the proposed approach is robust to noise, capable of automatically learning representative features from unlabeled data, and achieves high performance in fault classification. In addition, the proposed method is capable of classifying new conditions by fine-tuning the trained architecture applying small amounts of labeled data from that new condition. This proves suitability towards autoships which are subjected to varying environmental and operating conditions. The proposed method was verified with a standard data set of bearing faults and compared to SVM and k-nearest neighbor algorithm.

Thirukovalluru et al. [122] also pointed out the importance of the feature extraction process in diagnostics systems. This study compares the classification performance of traditionally human-engineered features and stacked denoising SAE generated features. The human-engineered features extraction methods used in this analysis are Fast Fourier Transform (FFT) and Wavelet Packet Transform (WPT), and SVM and Random Forest (RF) are used as the classifiers. The stacked denoising SAE is a variation of the original AE that both utilize the strengths from DAE and SAE, namely, the input corruption process and the sparsity penalty term. The results of the experiments showed that the stacked denoising SAE generated features achieved higher classification performance than the human-engineered features methods at least once. The results were validated using five different data sets: air compressor monitoring, drill bit monitoring, steel plate monitoring, and two data sets of bearing fault-monitoring data.

High-quality labeled training data and expert knowledge are not easily obtained regarding induction motors due to environmental interference and inherent motor structure complexity. For that reason, Sun et al. [113] also proposed a stacked denoising SAE in order to improve induction motor fault classification by reducing the dependency of labeled data and expert knowledge. The input corruption process enhances the robustness of the automatically extracted features and the stability of the proposed architecture. The extracted features are then used to train a classifier. Both SVM and logistic regression (LR) are considered as the classifiers. The “dropout” technique [123] is also introduced in this study. This is a regularization technique invented in 2014, and it was integrated into the whole architecture to reduce overfitting in the training process. For verification, the effectiveness of proposed architecture was compared with three different BP-NNs.

C. Long-Short Term Memory

1) *Introduction*: Recurrent Neural Networks (RNNs) [26],

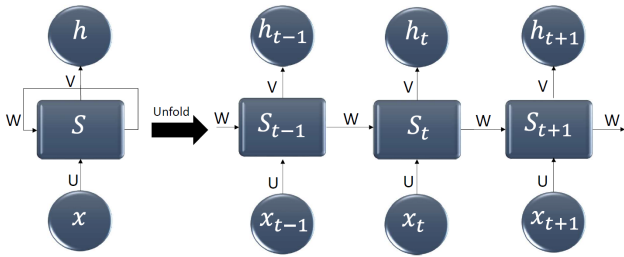


Fig. 5: RNN unfolded in time, adopted from [79]. The same weight matrices (U,V,W) are used at each time step.

[79], [94] are a group of neural networks used for tasks that involve sequential data. The popularity of RNNs emerged with the idea of connecting past information to the current task. In order to do so, traditional RNNs share the same weights (U,V,W) across several time steps, and this is the main difference compared to FNN. Weight sharing is important because a specific piece of information can occur at several positions within the sequential data [94]. RNNs are usually trained with the back-propagation algorithm to calculate the derivative of a total error with respect to all states, S_t , and all the parameters [79]. Figure 5 illustrates a simple model of this.

However, during the early 1990s, [124], [125] discovered a vanishing and exploding gradient problem. That is, when the shared (fixed) weight, W , is multiplied by itself several times, depend on magnitude, the product, W^t , will either vanish or explode [94]. Consequently, when the gap between previous relevant information and the present task becomes large, the information will be lost, and hence, the traditional RNN have difficulties of learning long-term dependencies.

One of the most popular approaches to reduce the difficulty of learning long-term dependencies is the LSTM. The original LSTM is a special kind of RNN that was first introduced by [126]. The initial idea of the LSTM architecture is to introduce a memory cell. This memory cell contains non-linear gating units in order to regulate the information flow in and out of the cell. By this, the memory cell is able to maintain its state over long durations, and the weights are conditioned on the context and not fixed. Thus, the time scale of integration can vary dynamically [94]. The literature provides several modifications and variations of the original LSTM; see [127] for a thorough review. Regarding recent applications to PHM, the vanilla LSTM with no peephole connections, originally described by [128], [129], is the most common choice. For that reason, the paragraphs below will discuss vanilla LSTM (hereinafter referred to as LSTM).

By introducing the memory cell, LSTMs are explicitly designed to learn long-term dependencies. Inside the memory cell, as illustrated in Figure 6, three non-linear gating units protect and regulate the cell state, S_t . The gating units introduce a sigmoid layer, σ , in order to obtain an output value between 0 and 1, input weights W , recurrent weights R , and bias weights b . The paragraphs below are based on a comprehensive blog post regarding the understanding of LSTM networks, [130].

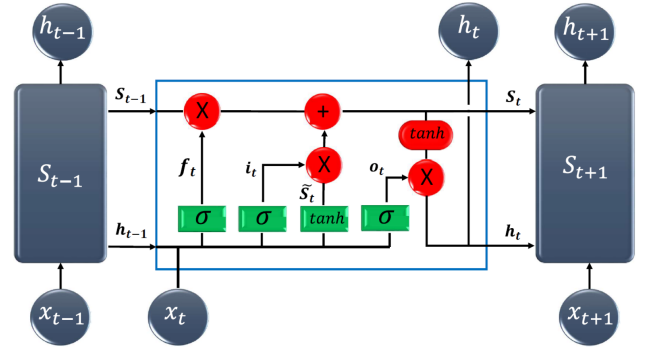


Fig. 6: Vanilla LSTM, adopted from [130]. The blue rectangle represents the memory cell.

The first gating unit is called the forget layer, and is defined as:

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f) \quad (17)$$

The forget layer determines which historical information the memory cell removes from the cell state. In this layer, an output value of 0 means to completely remove it, while an output value of 1 means to completely keep it. The second gating unit consists of two parts. The first part is called the input layer:

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \quad (18)$$

The input layer decides which values the memory cell will updated. The second part, is a tanh layer who creates a vector of new candidate state values, \tilde{S}_t :

$$\tilde{S}_t = \tanh(W_s x_t + R_s h_{t-1} + b_s) \quad (19)$$

In this way, the second gating unit determines what new information the memory cell is going to store in the cell state. Obviously, the next step is to update the previous cell state, S_{t-1} , into the new cell state, S_t :

$$S_t = f_t \otimes S_{t-1} + i_t \otimes \tilde{S}_t \quad (20)$$

where, \otimes , denotes element-wise multiplication of two vectors. First, the previous state is multiplied by the output from forget layer, and then the new candidate state values are added, scaled by the output from the input layer, that is, how much each new candidate state value will be updated. The third and final gating unit decides the output. This also consists of two parts. The first part is the output layer:

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \quad (21)$$

The output layer determines which parts of the cell state the memory cell is going to output. Then, the second part will create a filtered version of the cell state in order to push the values between -1 and 1, and finally multiply it by the scaled output value from the output layer:

$$h_t = o_t \otimes \tanh(S_t) \quad (22)$$

Through this procedure, LSTMs have the ability to remove or add information to the cell state. The traditional RNN lack this ability, and hence, it will completely override cell states.

TABLE II: The C-MAPSS data set [106].

Data set	FD001	FD002	FD003	FD004
Timeseries training set	100	260	100	249
Timeseries test set	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

2) *Recent Applications to PHM*: LSTMs are highly capable of learning long-term dependencies and specially designed for sequential data. With respect to PHM applications, sequential data is a standard format of the input data, e.g temperature and vibration measurements [131]. For that reason, LSTMs are good candidates for RUL estimations because LSTMs might reveal hidden information in the data, as well as the past dependencies that may influence future events. The paragraphs below review applications of LSTM to PHM in the recent three years.

Chen et al. [132] applied LSTM in mechanical state predictions. The proposed method was divided into two steps. The first step, which had two parts, involved feature extraction methods to obtain the mechanical state characteristics. The empirical mode decomposition method was used to decompose bearing data into stationary signals. Then, the intrinsic mode function energy entropy was calculated based on the decomposition. This calculation was to characterize the mechanical state. The second step applied the LSTM network in order to make predictions. The results indicated that the mean square error index of the LSTM network was lower compared to SVM.

Accurate and reliable RUL estimations play a vital role for a PHM system. Traditional data-driven approaches, such as HMMs and traditional RNNs, encounter difficulties when modeling sequential data. Both approaches have issues with long-term dependencies. In addition, CNNs do not fully account for sequence information because of their segmented input. Based on this, Zheng et al. [33] proposed an LSTM approach to provide RUL estimations. The proposed architecture consists of multiple layers of LSTMs combined with multiple layers of FNN. The LSTM layers are good for temporal modeling and can reveal hidden patterns in the sequential input data. The FNN layers are then applied in order to map the LSTM features and predict the RUL. Multiple layers are used to discover the complex relationship within the sensor data. This study used both the C-MAPSS data set [106] and the PHM08 data set [102]. The results indicated that the proposed architecture outperformed the above-mentioned approaches. The proposed method was trained in a supervised manner by engineering and utilizing piece-wise linear RUL targets, as recommended by [133], for the training data sets.

A similar and more comprehensive study on the C-MAPSS data set [106] was conducted in [134]. In this work Wu. et al also proposed an LSTM approach for RUL estimations. The C-MAPSS data set consists of four subsets as shown in Table II. Both subset FD002 and FD004 involves several operating conditions. Therefore, a dynamic difference method was proposed in order to extract new features from inter-frame dynamic changes before the training procedure. These

changes contain valuable degradation information, and hence, enable the LSTM approach to better control the underlying physical processes. The proposed method indicated improved performance compared to traditional RNN and gated recurrent unit LSTM (GRU-LSTM) [135].

Yuan et al. [136] proposed another LSTM approach to providing RUL estimations. However, the motivational factor in this study was to utilize the LSTM approach to build a common model for several different faults. In addition, the proposed approach was able to get RUL estimations and the probability of each fault at the same time. This feature makes it easy to design confidence intervals. The proposed approach was compared with traditional RNN and two variations of LSTM: GRU-LSTM and AdaBoost-LSTM. However, in all cases, the LSTM showed enhanced performance. The comparison used the C-MAPSS data set [106]. In this study, an SVM was used as an anomaly detector to create labels.

The majority of recent PHM applications based on DL have been focusing on either automatic feature extraction, classification, or regression. For that reason, Liao et al. [131] proposed a novel end-to-end deep architecture by stacking LSTM, FNN, and survival analysis. In this way, the proposed method integrates feature extraction and prediction as a single optimization task. This study utilized the LSTM as the first feature extraction layer. The reason for this is that the LSTM layer is able to handle the raw sequential input, and potentially discover past information that may influence future events. The extracted features will then be the input for the FNN layer, which makes it possible to further improve learning of the feature representation. Finally, the survival model learns the features and predicts the failure probability to indicate health conditions. Stochastic gradient descent is used as the learning optimization method for all the parameters. The proposed method showed promising results and was validated by a small data set of fleet mining haul trucks and by a large open source data set.

It should be noted that the proposed methods in the above LSTM-studies are based on supervised learning. In other words, trained on labeled training data. Nevertheless, in real-world applications, e.g. the maritime industry, high-quality labeled training data is hard to acquire, and large amounts of the data are unlabeled. For that reason, the first feature extraction layer, in any architecture, would have the advantage of utilizing unsupervised learning strategies, like the above mentioned DBNs or AEs. Gensler et al. [137] proposed an interesting approach that combines AE and LSTM. Specifically, the proposed approach combines automatic feature extraction from unlabelled training data with the temporal context utilization of the LSTM. The main idea is that after pre-training the AE in an unsupervised manner, the network architecture will be cut after the encoding side (bottleneck), and then the learned encoding will act as the input for the LSTM. Finally, the AE-LSTM architecture will be fine-tuned, where only the LSTM weights are trained, to produce the desired output. The proposed approach showed enhanced prediction performance compared to MLP, LSTM, and DBN. Although this study is targeting solar power forecasting, the proposed method has the potential to provide inspiration towards future intelligent PHM

systems applied to autoships.

Malhotra et al. [138] provides another interesting approach. Prognostic approaches for RUL estimations are generally based on the assumption that the health degradation curve follows a specific shape, e.g. exponential or linear. In this study it was observed that such assumptions are not well-suited for real-world applications. In addition, most prognostic approaches are application-dependent, meaning they are not robust towards new conditions. Based on these observed limitations, this study proposed an unsupervised approach, by utilizing an LSTM encoder-decoder, to obtain a health index (HI) for a multi-sensor time-series data system. The HI is then used to learn a regression model for RUL estimations. Briefly explained, the LSTM encoder learns a representation of the input time-series. Then, the LSTM decoder applies this representation to reconstruct the time-series using the current hidden state and the predicted value of the previous time-step. See [27] for a deeper understanding of the LSTM encoder-decoder method. The study used the C-MAPSS data set [106] and a milling machine data set, as well as a case study on real-world data, for validation. Surprisingly, the proposed approach showed improved performance compared to approaches that rely on assumptions about health degradation.

D. Convolutional Neural Network

1) *Introduction:* CNNs [20], [79], [94], [139] are designed for processing multiple arrays of 1D, 2D, or 3D grid-structured topology data. Examples of 1D, 2D, and 3D grid are: time-series data taking samples at systematic time intervals, pixels in image data, and video or volumetric images, respectively. CNNs have been inspired by earlier work on time-delay neural networks (TDNNs) [140]. Primarily, TDNNs are one-dimensional CNNs applied to time-series and use shared weights in a temporal dimension in order to reduce learning computation requirements. In addition to shared weights, convolution, pooling, and multi-layer architectures are the important ideas of CNNs. In fact, CNNs are the first truly DL technique to successfully train multiple layers [17].

A CNN can briefly be defined as a neural network that uses the mathematical operation convolution instead of general matrix multiplication in at least one of its layers [94]. With respect to mathematical understanding, convolution is an operation to combine two functions of a real-valued argument by measuring the overlap of two functions when one proceeds over the other. The discrete 1D convolution operation can be defined as:

$$\mathbf{S}(t) = (\mathbf{I} \cdot \mathbf{K})(t) = \sum_a \mathbf{I}(a)\mathbf{K}(t - a) \quad (23)$$

where t is the discretized time index, \mathbf{I} is the input, \mathbf{K} is a kernel (filter), and a is the finite number of array elements. The output, $\mathbf{S}(t)$, is usually referred to as the feature map. Expanding Equation 23, the discrete 2D convolution operation can be defined as:

$$\mathbf{S}(i, j) = (\mathbf{I} \cdot \mathbf{K})(i, j) = \sum_m \sum_n \mathbf{I}(m, n)\mathbf{K}(i - m, j - n) \quad (24)$$

where i and j are the discretized time indexes, and m and n are the finite number of array elements in each of the two dimensions. However, in the context of CNNs in practice, the standard discrete convolutional operations are moderately different. The reason for this is that the operation consists of several convolutions in parallel in order to extract many types of features at several locations in the input data. In addition, the input is normally a grid of vector-valued observations, and not only a grid of real values. Full derivations of the above equations, as well as practical variations of the standard discrete convolutional operation, appear in [94].

The convolution operation exploits three prime features of CNNs in the learning process [20], [94]. First, CNNs have sparse interactions. This is realized by making the kernel smaller than the input, and hence, CNNs needs to store fewer parameters compared with traditional FNN. This is because traditional FNN uses general matrix multiplication between layers, that is, every output unit interacts with every input unit. The sparse interaction feature reduces computational and memory requirements, as well as increasing statistical efficiency. Second, shared weights between several functions in the architecture further reduce the memory requirement and the complexity of the network. Finally, shared weights result in equivariance in the layers. That is, the output will change according to the input.

Researchers generally use one of two sets of terminology for describing the conceptual structure of CNNs. The first is the complex layer terminology [94], which this survey paper employs. The second is the simple layer terminology where every processing step is considered to be a separate layer. [17], [20] further describes simple layer terminology.

Complex layer terminology understands each layer in a CNN as having three processing steps, as illustrated in Figure 7. The first step, $C1$, executes several convolution operations in parallel in order to produce feature maps with linear activations. These activations are then processed by non-linear activation functions, σ , in the second step. The sigmoid activation function or the ReLU are common choices. In the third step (usually called sub-sampling), a pooling function, $S1$, is used to further adjust the output by calculating summary statistics of the nearby outputs. Regular choices are max-pooling [141] and average-pooling. Max-pooling calculates the maximum output in a rectangular neighborhood, while average-pooling calculates the average output. All pooling functions further reduce the dimensionality of the representations, and at the same time, generates an invariance to small translations of the input, namely, if the input to the pooling function changes, most of the pooled outputs do not change [79], [94]. This layer procedure repeats itself in the next layers. Finally, the outputs from the last layer are rasterized and presented as a single input vector to a traditional FNN in order to perform functions such as classification or regression.

The training procedure of CNNs is introduced in [139]. It is similar to standard back-propagation training performed on FNN, but the reduced number of parameters and shared weights in CNNs improve the training efficiency. In addition, CNNs are capable of handling raw input data, and hence, pre-processing is rare. This means that CNNs are less dependent

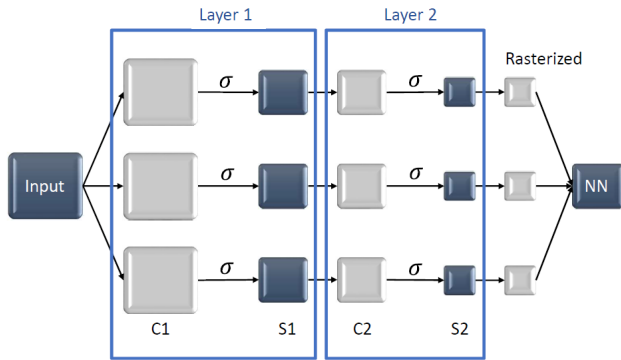


Fig. 7: Conceptual structure of CNN.

on prior knowledge and human-engineered feature extraction methods which appears to be suitable towards future intelligent PHM systems applied to autoships. The convolutional operations also enable CNNs to process inputs with varying spatial extents [94].

2) *Recent Applications to PHM*: According to [94], CNNs have been most successful on 2D and 3D grid-structured topology data, like object recognition [142] and face recognition [143], respectively. Nevertheless, recent applications to PHM have applied 1D grid-structured topology for sequential data and shown enhanced performance compared with traditional machine learning algorithms. For that reason, CNN applications to PHM in the years between 2016 and 2018 will be reviewed in the following paragraphs.

In order for CNNs to act as an effective feature extraction method for raw industrial signals in PHM applications, the successful applications of CNNs to 2D and 3D grid topology data have to be modified. The reason for this is that raw industrial signals are usually 1D time-series with hidden information behind strong intervals and deep correlations amid various time points. Thus, to fully exploit all the information in the signal, it is necessary to consider the relationship between signals in diverse locations. Liu et al. [144] proposed a novel dislocated time-series CNN (DTS-CNN) diagnostics approach. The proposed approach utilizes a dislocated layer as the initial layer in order to extract the relation between periodic vibration signals with varying intervals. After the initial dislocated layer, there are two layers with convolution- and max-pooling steps, then a fully connected softmax classifier is used for classification. For verification, an electric machine fault simulator with different operating conditions was used in two experiments. The DTS-CNN showed improved performance, compared to traditional CNN and wavelet packet SVM, due to its robustness under different non-stationary operating conditions. In addition, the proposed approach was capable of automatically extracting features from raw input data, and hence, less dependent on human-engineered feature extraction methods and prior knowledge.

Jing et al. [145] proposed a CNN approach to provide automatic feature extraction and fault diagnosis. The motivational factor in this study was three limitations of traditional

diagnostic algorithms: First, their inability to handle raw input data. This requires human-engineered feature extraction methods, which means the diagnostics accuracy heavily depends on domain expertise and prior knowledge. Second, feature extraction methods are application-dependent. In this way, the diagnostics results are sensitive to changes in the mechanical system. Third and finally, traditional diagnostics algorithms lack the ability to mine new features. In this study, CNN was applied to automatically learn features from raw vibration data in the time domain, frequency domain, and time-frequency domain, and then conduct diagnostics of gearboxes. 1D segments are collected from the raw vibration data as the input for the CNN. This study uses one layer with convolution and pooling steps because this approach showed higher accuracy and more stable results than other configurations. In the end, a fully connected softmax layer was used for classification. The proposed approach was validated with a publicly available data set for gearboxes. The comparison uses manual feature extraction methods from each domain and traditional diagnostics approaches such as FNN, SVM, and RF. The results indicated that the proposed approach outperformed the comparative methods.

Traditional prognostics algorithms are usually based on shallow architectures. Consequently, they lack the ability to capture more complex relationships between the sensory data and RUL estimations. In addition, traditional algorithms do not have the ability to automatically learn salient features. Based on these restrictions, Babu et al. [88] proposed a novel CNN-based regression approach for RUL estimations from multi-variate time-series sensor signals. However, applying CNN to multiple channels of time-series signals has two main challenges that apply to the processing steps: they need to be applied along temporal dimensions, and they need to be shared among multiple sensors. To cope with these challenges, this study adopted a sliding window strategy in order to create segmented collections of the time-series signals. Each segment is then fed into two layers with convolution- and average-pooling steps. The first convolution step is two-dimensional, while the second and the two pooling steps are one-dimensional. These processing steps automatically capture salient features of the sensor signals at different time scales, and hence, the features extracted are task dependent and not human-engineered. Finally, all salient features are systematically unified and mapped into the RUL estimation of a traditional FNN regressor. Both the PHM08 data set [101] and the C-MAPSS data set [106] were used to validate the results. The proposed approach showed enhanced performance compared to MLP, SVM, and RVM. It should be noted that a piece-wise linear RUL target function has been used in this study. This means a supervised training procedure with target run-to-failure data. However, Zheng et al. [33] claim that LSTM outperformed the CNN approach in their study because the RUL estimations in the CNN approach are built based on independent sliding windows. Sliding windows are in fact time-dependent with respect to RUL estimations, and hence, the CNN approach does not fully consider sequence information.

In 2018, Li et al. [34] proposed a new CNN approach

that has shown improved RUL prediction performance on the C-MAPSS data set [106] compared to both the CNN approach in [88] and the LSTM approach in [33]. Similar to [88], this CNN approach also prepares the input data in two dimensions (sequence length \times selected features) and utilize a time-window strategy. However, in contrast to [88], this CNN approach performs all the convolution steps in one dimension. In this way, the CNN is able to learn high-level representations of each raw feature from the very start rather than learning the spatial relationship of several features and then extracting information. Additionally, the proposed approach employs the advanced regularization technique “dropout” [123] and the adaptive learning rate method “Adam” [146], both invented in 2014.

E. Discussion and Summary

Table III provides a structured summary of the recent PHM applications based on DL reviewed in this survey paper. This representative set of applications has been selected to provide insight into the current state-of-the-art and to encourage important directions for future research towards intelligent PHM systems suitable for autoships. DBNs, AEs, DAEs, SAE, LSTMs, and CNNs have been explicitly chosen as the DL techniques primarily because they are well-established and show great promise for future developments.

With respect to autoships and the maritime industry more generally, future intelligent PHM systems will have to adapt to highly varying operational and environmental conditions, as the maritime environment is harsh and uncertain. Additionally, such systems needs to provide automatic pre-processing and dimensionality reduction schemes in order to effectively process massive data flows of high-dimensional and unstructured data. Based on the reviews this paper has discussed, DBNs, AEs, DAEs, and SAEs seem like promising ways to address these challenges since these DL techniques provide an unsupervised learning procedure as an initial pre-training step. This procedure will automatically capture abstract, important statistical structures and reduce dimensionality of raw unlabeled input data. As a result, DL techniques minimize the need for human-crafted feature extraction methods in the data processing stage, reduce the need for large amounts of high-quality labeled training data, and have the potential to be applied to new conditions by fine-tuning the trained final architecture using a much smaller labeled data set from that new condition. [104], [107], [113], [116], [137] intensively investigates the effectiveness of these qualities.

Another important concern is the fact that sensor data is going to be the most common data type format for future intelligent PHM systems used in autoships. LSTMs are highly capable of learning long-term dependencies that may influence future events, they are specially designed for sequential data, and might discover hidden data information. [33], [134] suggests the strengths of the LSTM. Furthermore, the CNN approach in [34] seems to be highly suitable for sequential data. Actually, it outperformed both an equivalent CNN approach in [88] and the LSTM approach in [33] on the C-MAPSS data set [106].

TABLE III: Recent PHM applications based on DL (the years between 2013 and 2018).

DL technique	Author. Reference	Year	PHM application	
DBN	Deutsch et al. [107]	2017	Automatic feature extraction and failure prognostics: Rotating components	
	Li et al. [104]	2016	Automatic feature extraction and fault diagnostics: Rotary machinery	
	Zhang et al. [105]	2016	Automatic feature extraction and failure prognostics: C-MAPSS data set [106]	
	Liao et al. [108]	2016	Automatic feature extraction and failure prognostics: Rotating systems	
	Jiang et al. [109]	2016	Automatic feature extraction and Time-series prediction: Complex systems	
	Yang Fu et al. [100]	2015	Automatic feature extraction: Cutting state monitoring	
	Tamilselvan et al. [32]	2013	Fault diagnostics: PHM08 data set [101]	
DAE	Tran et al. [103]	2013	Electric power transformer Fault diagnostics: Reciprocating compressor valves	
	Xia et al. [116]	2017	Automatic feature extraction and fault diagnostics: Motor bearings	
	Sun et al. [113]	2016	Automatic feature extraction and fault diagnostics: Induction motor	
	Thirukovalluru et al. [122]	2016	Automatic feature extraction and fault diagnostics: Air compressor monitoring Drill bit monitoring Steel plate monitoring	
	Lu et al. [119]	2015	Bearing fault monitoring Automatic feature extraction and fault diagnostics: Rolling bearing data	
	Jia et al. [121]	2015	Automatic feature extraction and fault diagnostics: Rolling element bearing	
	Ma et al. [120]	2014	Planetary gearbox Automatic feature extraction and fault diagnostics: Power transformers Circuit breakers	
	LSTM	Wu et al. [134]	2018	Failure prognostics: C-MAPSS data set [106]
		Chen et al. [132]	2017	Failure prognostics: Bearings
		Zheng et al. [33]	2017	Failure prognostics: PHM08 data set [101]
Yuan et al. [136]		2016	C-MAPSS data set [106] Failure prognostics: C-MAPSS data set [106]	
Liao et al. [131]		2016	Feature extraction and failure prognostics: Mining haul trucks	
Gensler et al [137]		2016	Automatic feature extraction and failure prognostics: Solar power plants	
AE/LSTM	Malhotra et al. [138]	2016	Automatic feature extraction and failure prognostics: C-MAPSS data set [106] Milling machine	
CNN	Li et al. [34]	2018	Automatic feature extraction and failure prognostics: C-MAPSS data set [106]	
	Liu et al. [144]	2017	Automatic feature extraction and fault diagnostics: Electric machine fault simulator	
	Jing et al. [145]	2017	Automatic feature extraction and fault diagnostics: Gearbox	
	Babu et al. [88]	2016	Automatic feature extraction and failure prognostics: PHM08 data set [101] C-MAPSS data set [106]	

V. DISCUSSION

The previous sections discuss the benefits and challenges of applying PHMDL in autoships in addition to specific introductions and reviews of well-established DL techniques with respect to recent applications to PHM. These sections are intended to enlighten the reader with both opportunities and a more theoretical and practical understanding on how PHMDL can be applied to autoships. This section gives more general discussions concerning suitable deep architectures in order to address the challenges and exploit the benefits. Finally, existing problems and future research opportunities are introduced.

A. Suitable Deep Architectures

The majority of the reviewed studies in Section IV, either utilize DBN, AEs, LSTM, or CNN as the selected DL technique. However, different DL techniques can be stacked in order to further exploit the advantages and reduce the drawbacks of each DL technique. Gensler et al. [137] combines AE and LSTM to both make use of the automatic feature extraction from unlabeled input data and the temporal context utilization of LSTM. This approach has served as inspiration for this paper's proposed deep architecture for intelligent PHM systems for autoships.

PHMDL systems in autoships demands automatic pre-processing and dimensionality reduction schemes due to varying operational and environmental conditions and massive data flows of high-dimensional and unstructured data. Thus, the unsupervised DL techniques, DBN and the modern variations of the original AE, DAE, and SAE, have great potential to be applied as the first layer of a deep architecture. More specifically, the stacked denoising SAE [113], [122] is an encouraging opportunity. The reason for this is that both the stochastic corruption process, $\tilde{x} \sim q(\tilde{x}|x)$, in DAEs and the sparsity penalty term, $\Omega(h)$, in SAEs appears particularly suitable for autoships and the maritime environment. This combination will give the first layer the potential to provide robustness towards noisy sensor data and to control the number of active hidden neurons. This will increase both the robustness of the system and the efficiency of the automatic feature extraction process. The increased efficiency reflects the reduced number of active neurons in the hidden layers, as it costs energy to activate neurons and to send signals between them. Actually, human brains seem to minimize such computational costs in a similar manner [89].

The lack of onboard crew members in autoships creates higher demands for scheduling maintenance procedures to the next appropriate port of call [31]. Consequently, PHMDL systems have to provide reliable RUL estimations of relevant components and sub-components. Additionally, sensor data will be the most common data type format for PHMDL systems in autoships. Thus, LSTM is a quite promising candidate to act as the following layer or layers of a deep architecture. LSTM is especially designed for sensor data and was capable of revealing hidden information and learning long-term dependencies within sensor data with multiple operating and fault conditions in [33]. This proves the potential to enhance RUL estimations. Finally, a traditional FNN layer

can be applied in order to map all extracted features and provide RUL estimations. However, it should be noted that any RUL estimation should include associated confidence intervals in order to provide reliable and trustworthy outputs. This is necessary to help autoships to optimize maintenance planning.

The well-proven regularization technique, "dropout" [123], extends the idea of the DAE. "Dropout" randomly drop units during training, and hence, regularize a network by adding noise to its hidden units. In this way, the network learns to make generalized representations of the input data, which enhances the feature extraction ability. "Dropout" can be applied to all hidden layers in both LSTMs and FNNs. Therefore, it should be considered as part of the proposed deep architecture. It should be noted that "dropout" should only be applied to the non-recurrent connections in LSTMs.

B. Existing Problems and Future Research Opportunities

Even if unsupervised DL techniques are applied in the first layer, the deep architecture will still require a reduced amount of labeled training data in order to perform supervised classification or regression in the final layer. The labeled training data is necessary to fine-tune the whole architecture with respect to the final classification or regression task. As a consequence, the supervised learning procedure assumes that input events are independent of earlier output events [89]. As a result, the DL techniques reviewed in this survey paper do not involve learning to act in totally unknown environments. We assume that this feature will be extremely useful in future intelligent PHM systems applied to autoships due to the lack of fault labels and run-to-failure data of components and sub-components [35].

Additionally, according to [147], current DL techniques are insufficient for fixed network architectures. This is because recently introduced tri-traversal theory proves that DL techniques will need to adapt at three levels of organization (T3-structure), that is, be equipped with intelligent procedures that rapidly adjust their network architectures, in order to deal with the complexity of the maritime environment as well as other contexts.

Solving this problem through a combination of DL and reinforcement learning (RL) where there is no supervised teacher is an exciting research objective. Briefly explained, RL enables learning from feedback received through interactions with an external environment [18]. In the application of autoships, we propose that this environment could involve several environmental and operating conditions. Assuming some typical values of the sensor data in each condition, RL is able to search for possible inputs and outputs in order to maximize a reward [148], e.g. the performance of the deep architecture. Based on the observed rewards, RL is able to obtain the optimal, or nearly optimal, deep architecture structure for each condition. Consequently, RL adaptively adjusts the hyper-parameters in the deep architecture, e.g. learning rate, number of hidden layers and nodes, "dropout" rate, etc. Its policy requires RL to decide whether to use the hyper-parameters that gave the highest reward last time for a specific condition or to try out different hyper-parameters in hope of providing even better performance [148].

Another interesting research objective is the digital twin (DT). Today, the concept of the DT is one of the most-sought research objectives in the maritime industry [149]. A DT will consist of a series of simulation models that are continuously updated to mirror their real-life twins [150], e.g. an autoship. In this way, a DT differs from a generic model because it is specific to its physical counterpart. The DT model must also reflect changes involving the physical autoship. A DT will include a system and data information model, simulation models and data analytics, and dependability and performance models [35]. The DT concept allows new design paradigms where different stakeholders will be able to contribute to the creation of a DT with specific models and evaluate in advance how the autoship will operate in different scenarios [90]. In this way, the DT is able to collect highly important prior knowledge, in addition to the knowledge acquired through sensors etc. in the current state. The new design paradigms enable DTs to build extensive databases regarding run-to-failure data of critical and relevant components and sub-components. This will be highly beneficial and necessary for successful implementations of future intelligent PHM systems on autoships.

According to [82], DL has the property that it gets better as it receives more data. This property also applies to DTs because, over time, they will be increasingly more detailed and continuously updated with sensor data. This quality accelerates developments towards data-driven approaches suitable for industrial big data, such as DL techniques.

VI. CONCLUSION

Autoships are a provident and rapidly expanding research field. In order to operate and maintain complex and integrated systems in a safe, efficient, and cost-beneficial manner, autoships are expected to include intelligent PHMDL systems. PHMDL have the potential to reduce built-in redundancy and to provide more robust and reliable maintenance scheduling.

In this survey paper, well-established DL techniques have been introduced and reviewed with respect to recent PHM applications. In these reviews, DL techniques have demonstrated that they are a superior alternative to human-crafted feature extraction methods combined with traditional machine learning algorithms in many practical PHM problems. Hence, DL techniques are highly suitable to be applied to diagnostics and prognostics tasks in future intelligent PHM systems in the age of big data. The main intention of this survey paper is to support creativity and inspiration in explorations of PHMDL possibilities in the maritime industry, particularly autoships. To guide future researchers, this paper introduced and discussed the benefits and challenges of implementation of PHMDL in autoships, and the maritime industry in general. In highly varying operational and environmental conditions and massive data flow, DL techniques will be advantageous due to unsupervised learning procedures that automatically extract high-level abstract features and, at the same time, reduce the dimensionality of raw unlabeled input data. In this way, PHMDL is less application-dependent than traditional approaches, and hence, has the potential to operate in different conditions.

This paper has also provided more general discussions, concerning suitable deep architectures, existing problems, and future research opportunities. It appears that DL, RL, and DT all have the potential to push the development of autoships and intelligent PHM systems to the next level.

ACKNOWLEDGMENT

This work was supported by the Norwegian University of Science and Technology within the Department of Ocean Operations and Civil Engineering under project no. 90329106. The authors would like to thank Digital Twins For Vessel Life Cycle Service (DigiTwin) and the Research Council of Norway, grant no. 280703.

REFERENCES

- [1] O. Levander, "Autonomous ships on the high seas," *IEEE Spectrum*, vol. 54, no. 2, pp. 26–31, February 2017.
- [2] L. Kretschmann, H.-C. Burmeister, and C. Jahn, "Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier," *Research in Transportation Business & Management*, 2017.
- [3] E. Jokioinen, "Remote and autonomous ships - the next steps: Introduction," *Rolls-Royce, Buckingham Gate, London: The Advanced Autonomous Waterborne Applications (AAWA)*, pp. 4–14, 2016.
- [4] K. E. Knutsen, G. Manno, and B. J. Vartdal, "Beyond condition monitoring in the maritime industry," *DNV GL Strategic Research & Innovation Position Paper*, 2014.
- [5] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics a review of current paradigms and practices," *The International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9, pp. 1012–1024, 2006.
- [6] T. M. Allen, "Us navy analysis of submarine maintenance data and the development of age and reliability profiles," *Department of the Navy SUBMEPP*, 2001.
- [7] B.-M. Batalden, P. Leikanger, and P. Wide, "Towards autonomous maritime operations," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, 2017, pp. 1–6.
- [8] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems: reviews, methodology and applications," *Mechanical Systems and Signal Processing*, vol. 42, no. 1, pp. 314–334, 2014.
- [9] A. Brandsaeter, G. Manno, E. Vanem, and I. K. Glad, "An application of sensor-based anomaly detection in the maritime industry," in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2016, pp. 1–8.
- [10] X. Xu, S. Fan, H. Huang, H. Zhu, and Q. Wen, "Research on phm technology application of ship maintenance program optimization," in *2013 IEEE Conference on Prognostics and Health Management (PHM)*, June 2013, pp. 1–6.
- [11] D. McDonnell, N. Balfe, S. Al-Dahidi, and G. O'Donnell, "Designing for human-centred decision support systems in phm," in *European Conference of the Prognostics and Health Management Society*. IEEE, 2014.
- [12] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.
- [13] P. Baraldi, F. D. Maio, D. Genini, and E. Zio, "Comparison of data-driven reconstruction methods for fault detection," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 852–860, 2015.

- [14] O. Geramifard, J.-X. Xu, C. K. Pang, J. Zhou, and X. Li, "Data-driven approaches in health condition monitoring: a comparative study," in *2010 8th IEEE International Conference on Control and Automation (ICCA)*. IEEE, 2010, pp. 1618–1622.
- [15] K. Goebel, B. Saha, and A. Saxena, "A comparison of three data-driven techniques for prognostics," in *62nd meeting of the society for machinery failure prevention technology (mfpt)*, 2008, pp. 119–131.
- [16] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.
- [17] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning - a new frontier in artificial intelligence research [research frontier]," *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010.
- [18] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, 2016.
- [19] X. W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.
- [20] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, no. Supplement C, pp. 11–26, 2017.
- [21] Z. H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, May 2015.
- [22] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [23] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.
- [24] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8595–8598.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [26] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1017–1024.
- [27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.
- [28] G. Wu, W. Lu, G. Gao, C. Zhao, and J. Liu, "Regional deep learning model for visual tracking," *Neurocomputing*, vol. 175, no. Part A, pp. 310–323, 2016.
- [29] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.
- [30] G. Zhao, G. Zhang, Q. Ge, and X. Liu, "Research advances in fault diagnosis and prognostic based on deep learning," in *Prognostics and System Health Management Conference (PHM-Chengdu)*. IEEE, 2016, pp. 1–6.
- [31] R. Jalonen, R. Tuominen, and M. Wahlström, "Remote and autonomous ships - the next steps: Safety and security in autonomous shipping - challenges for research and development," *Rolls-Royce Buckingham Gate, London: The Advanced Autonomous Waterborne Applications (AWA)*, pp. 56–73, 2016.
- [32] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, no. Supplement C, pp. 124–135, 2013.
- [33] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2017, pp. 88–95.
- [34] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.
- [35] A. S. Zymaris, Ø. Å. Alnes, K. E. Knutsen, and N. M. Kakalis, "Towards a model-based condition assessment of complex marine machinery systems using systems engineering," in *PHM Europe, Bilbao, Spain*, 2016, Conference Proceedings.
- [36] P. W. Kalgren, C. S. Byington, M. J. Roemer, and M. J. Watson, "Defining phm, a lexical evolution of maintenance and logistics," in *2006 IEEE Autotestcon*, Sept 2006, pp. 353–358.
- [37] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Pra, and N. Zerhouni, "Prognostics and health management of pemfc state of the art and remaining challenges," *International Journal of Hydrogen Energy*, vol. 38, no. 35, pp. 15 307–15 317, 2013.
- [38] M. A. Taie, M. Diab, and M. ElHelw, "Remote prognosis, diagnosis and maintenance for automotive architecture based on least squares support vector machine and multiple classifiers," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, Oct 2012, pp. 128–134.
- [39] A. Ismail and W. Jung, "Recent development of automotive prognostics," *Korean Reliability Society*, pp. 147–153, 2015.
- [40] P. L. Dussault, "Creating a closed loop environment for condition based maintenance plus (cmb+) and prognostics health management," in *2007 IEEE Autotestcon*, Sept 2007, pp. 327–331.
- [41] F. Camci, G. S. Valentine, and K. Navarra, "Methodologies for integration of phm systems with maintenance data," in *2007 IEEE Aerospace Conference*, March 2007, pp. 1–9.
- [42] K. M. Janasak and R. R. Beshears, "Diagnostics to prognostics - a product availability technology evolution," in *2007 Annual Reliability and Maintainability Symposium*, Jan 2007, pp. 113–118.
- [43] J. Lee, M. Ghaffari, and S. Elmelgy, "Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 111–122, 2011.
- [44] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.
- [45] W. Elghazel, J. Bahi, C. Guyeux, M. Hakem, K. Medjaher, and N. Zerhouni, "Dependability of wireless sensor networks for industrial prognostics and health management," *Computers in Industry*, vol. 68, no. Supplement C, pp. 1–15, 2015.
- [46] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, "Signal processing and database management systems," in *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc., 2007, ch. 4, pp. 95–171. [Online]. Available: <http://dx.doi.org/10.1002/9780470117842.ch4>
- [47] Z. K. Peng and F. L. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography," *Mechanical Systems and Signal Processing*, vol. 18, no. 2, pp. 199–221, 2004.
- [48] G. Niu and B.-S. Yang, "Intelligent condition monitoring and prognostics system based on data-fusion strategy," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8831–8840, 2010.
- [49] Y. Lei, "Signal processing and feature extraction," in *Intelligent*

Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery. Butterworth-Heinemann, 2017, ch. 2, pp. 17–66.

- [50] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, "Fault diagnosis," in *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc., 2007, ch. 5, pp. 172–283.
- [51] E. Zio and G. Gola, "Neuro-fuzzy pattern classification for fault diagnosis in nuclear components," *Annals of Nuclear Energy*, vol. 33, no. 5, pp. 415–426, 2006.
- [52] Y. Lei, "Introduction and background," in *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, Y. Lei, Ed. Butterworth-Heinemann, 2017, ch. 1, pp. 1–16.
- [53] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, "Fault prognosis," in *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc., 2007, ch. 6, pp. 284–354.
- [54] T. Khawaja, G. Vachtsevanos, and B. Wu, "Reasoning about uncertainty in prognosis: a confidence prediction neural network approach," in *NAFIPS 2005. Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 2005, pp. 7–12.
- [55] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot, "A data-driven failure prognostics method based on mixture of gaussians hidden markov models," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 491–503, June 2012.
- [56] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1803–1836, 2011.
- [57] G. W. Vogl, B. A. Weiss, and M. Helu, "A review of diagnostic and prognostic capabilities and best practices for manufacturing," *Journal of Intelligent Manufacturing*, 2016.
- [58] B. Sun, S. Zeng, R. Kang, and M. G. Pecht, "Benefits and challenges of system prognostics," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 323–335, 2012.
- [59] J. Liu, W. Wang, F. Ma, Y. B. Yang, and C. S. Yang, "A data-model-fusion prognostic framework for dynamic system state forecasting," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 4, pp. 814–823, 2012.
- [60] M. Tahan, E. Tsoutsanis, M. Muhammad, and Z. A. Abdul Karim, "Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review," *Applied Energy*, vol. 198, pp. 122–144, 2017.
- [61] T. Sutharssan, S. Stoyanov, C. Bailey, and C. Yin, "Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms," *The Journal of Engineering*, vol. 1, no. 1, 2015.
- [62] D. An, N. H. Kim, and J.-H. Choi, "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliability Engineering & System Safety*, vol. 133, pp. 223–236, 2015.
- [63] M. J. Roemer, C. S. Byington, G. J. Kacprzynski, and G. Vachtsevanos, "An overview of selected prognostic technologies with application to engine health management," *ASME Paper No. GT2006-90677*, 2006.
- [64] D. J. Power and R. Sharda, "Decision support systems," in *Springer Handbook of Automation*, S. Y. Nof, Ed. Springer Berlin Heidelberg, 2009, ch. 87, pp. 1539–1548.
- [65] N. Iyer, K. Goebel, and P. Bonissone, "Phm decision support under uncertainty," in *Annual conference of the prognostics and health management society*, 2016, Conference Proceedings.
- [66] —, "Framework for post-prognostic decision support," in *2006 IEEE Aerospace Conference*. IEEE, 2006, pp. 10–pp.
- [67] D. J. Power, "Web-based and model-driven decision support systems: concepts and issues," *AMCIS Proceedings*, p. 387, 2000.
- [68] "Nist big data interoperability framework: Volume 1, definitions," 2015. [Online]. Available: <http://dx.doi.org/10.6028/NIST.SP.1500-1>
- [69] L. Deng, "Expanding the scope of signal processing [from the editor]," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 2–4, May 2008.
- [70] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.
- [71] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [72] J. Hästad, *Computational Limitations of Small-depth Circuits*. Cambridge, MA, USA: MIT Press, 1987.
- [73] E. Allender, "Circuit complexity before the dawn of the new millennium," in *Foundations of Software Technology and Theoretical Computer Science: 16th Conference Hyderabad, India, December 18–20, 1996 Proceedings*, V. Chandru and V. Vinay, Eds. Springer Berlin Heidelberg, 1996, pp. 1–18.
- [74] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 473–480.
- [75] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 153–160.
- [76] C. Petrou and M. Paraskevas, "Signal processing techniques restructure the big data era," in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*. ACM, 2016, p. 52.
- [77] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [78] R. Bellman and R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
- [79] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [80] T. S. Lee and D. Mumford, "Hierarchical bayesian inference in the visual cortex," *Journal of the Optical Society of America A*, vol. 20, no. 7, pp. 1434–1448, 2003.
- [81] T. S. Lee, D. Mumford, R. Romero, and V. A. F. Lamme, "The role of the primary visual cortex in higher level vision," *Vision Research*, vol. 38, no. 15, pp. 2429–2454, 1998.
- [82] N. Jones, "Computer science: The learning machines," *Nature*, vol. 505, pp. 146–148, 01 2014.
- [83] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [84] G. E. Hinton, "What kind of a graphical model is the brain?" in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 1765–1775.
- [85] M. aurelio Ranzato, C. Poultney, S. Chopra, and Y. L. Cun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems 19*, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 1137–1144.
- [86] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [87] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," pp. 153–160, 2009/04/15 2009.
- [88] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*. Springer, 2016, pp. 214–228.
- [89] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [90] O. Niculita, O. Nwora, and Z. Skaf, "Towards design of prognostics and health management solutions for maritime assets," *Procedia CIRP*, vol. 59, pp. 122–132, 2017.

- [91] G. Manno, K. E. Knutsen, and B. J. Vartdal, "An importance measure approach to system level condition monitoring of ship machinery systems," in *11th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, CM 2014 and MFPT 2014, At Manchester*, 2014, Conference Proceedings.
- [92] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Morgan-Kaufmann, 1992, pp. 912–919.
- [93] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012, pp. 599–619.
- [94] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [95] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [96] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [97] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudk, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323.
- [98] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
- [99] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [100] Y. Fu, Y. Zhang, H. Qiao, D. Li, H. Zhou, and J. Leopold, "Analysis of feature extracting ability for cutting state monitoring using deep belief networks," *Procedia CIRP*, vol. 31, no. Supplement C, pp. 29–34, 2015.
- [101] A. Saxena and K. Goebel, "Phm08 challenge data set," *NASA Ames Prognostics Data Repository (<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>)*, NASA Ames Research Center, Moffett Field, CA, 2008.
- [102] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*, Oct 2008, pp. 1–9.
- [103] V. T. Tran, F. Althobiani, and A. Ball, "An approach to fault diagnosis of reciprocating compressor valves using teagerkaiser energy operator and deep belief networks," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.
- [104] C. Li, R.-V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera, "Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning," *Sensors*, vol. 16, no. 6, p. 895, 2016.
- [105] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.
- [106] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository (<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>)*, NASA Ames Research Center, Moffett Field, CA, 2008.
- [107] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–10, 2017.
- [108] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.
- [109] P. Jiang, C. Chen, and X. Liu, "Time series prediction for evolutions of complex systems: A deep learning approach," in *2016 IEEE International Conference on Control and Robotics Engineering (ICCRE)*. IEEE, 2016, pp. 1–6.
- [110] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*. IEEE, 2014, pp. 1–6.
- [111] R. Hrasko, A. G. C. Pacheco, and R. A. Krohling, "Time series prediction using restricted boltzmann machines and backpropagation," *Procedia Computer Science*, vol. 55, no. Supplement C, pp. 990–999, 2015.
- [112] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted boltzmann machines," *Neurocomputing*, vol. 137, no. Supplement C, pp. 47–56, 2014.
- [113] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, no. Supplement C, pp. 171–178, 2016.
- [114] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.
- [115] 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, pp. 3371–3408, Dec. 2010.
- [116] M. Xia, T. Li, L. Liu, L. Xu, and C. W. d. Silva, "Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder," *IET Science, Measurement & Technology*, vol. 11, no. 6, pp. 687–695, 2017.
- [117] M. Ranzato, C. Poultney, S. Chopra, and Y. Lecun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems 19*, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 1137–1144.
- [118] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951.
- [119] W. Lu, X. Wang, C. Yang, and T. Zhang, "A novel feature extraction method using deep neural network for rolling bearing fault diagnosis," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, May 2015, pp. 2427–2431.
- [120] Y. Ma, Z. Guo, J. Su, Y. Chen, X. Du, Y. Yang, C. Li, Y. Lin, and Y. Geng, "Deep learning for fault diagnosis based on multi-sourced heterogeneous data," in *2014 International Conference on Power System Technology*, Oct 2014, pp. 740–745.
- [121] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72–73, no. Supplement C, pp. 303–315, 2016.
- [122] R. Thirukovalluru, S. Dixit, R. K. Sevakula, N. K. Verma, and A. Salour, "Generating feature sets for fault diagnosis using denoising stacked auto-encoder," in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2016, pp. 1–7.
- [123] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
- [124] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," *Diploma thesis, Technische Universität München*, vol. 91, 1991.

- [125] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar 1994.
- [126] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [127] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–11, 2017.
- [128] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.
- [129] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, July 2005, pp. 2047–2052 vol. 4.
- [130] C. Olah, "Understanding lstm networks. 2015," URL <http://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png>, 2015.
- [131] L. Liao and H.-i. Ahn, "Combining deep learning and survival analysis for asset health management," *International Journal of Prognostics and Health Management*, vol. 16, 2016.
- [132] Z. Chen, Y. Liu, and S. Liu, "Mechanical state prediction based on lstm neural network," in *Control Conference (CCC), 2017 36th Chinese*. IEEE, 2017, pp. 3876–3881.
- [133] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*. IEEE, 2008, pp. 1–6.
- [134] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla lstm neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 2018.
- [135] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [136] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *IEEE International Conference on Aircraft Utility Systems (AUS)*. IEEE, 2016, pp. 135–140.
- [137] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting approach using autoencoder and lstm neural networks," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 002 858–002 865.
- [138] P. Malhotra, V. Tv, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder," in *2016 Workshop on Machine Learning for Prognostic and Health Management*, 2016, Conference Proceedings.
- [139] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
- [140] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar 1989.
- [141] Y. T. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *IEEE 1988 International Conference on Neural Networks*, July 1988, pp. 71–78 vol.2.
- [142] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [143] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1701–1708.
- [144] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, 2017.
- [145] L. Jing, M. Zhao, P. Li, and X. Xu, "A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox," *Measurement*, vol. 111, no. Supplement C, pp. 1 – 10, 2017.
- [146] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [147] D. Nikoli, "Why deep neural nets cannot ever match biological intelligence and what to do about it?" *International Journal of Automation and Computing*, 2017.
- [148] S. Marsland, "Reinforcement learning," in *Machine learning: an algorithmic perspective*, R. Herbrich and T. Graepel, Eds. Chapman & Hall/CRC, 2015, ch. 11, pp. 231–247.
- [149] K. B. Ludvigsen, L. K. Jamt, N. Husteli, and J. Smogeli, "Digital twins for design, testing and verification throughout a vessel's life cycle," in *15th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT), Lecce, Italy, 2016*, Conference Proceedings, pp. 448–457.
- [150] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.



André Listou Ellefsen received his Master 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, Alesund, 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, prognostics and health management, and digital twins.



Prof. Dr. Vilmar Æsøy graduated from NTNU in 1989, and continued his research on natural gas fueled marine engines at NTNU/MARINTEK to 1997. In 1996 he received his PhD degree for his research on natural gas ignition and combustion through experimental investigations and numerical simulations. During the research period 1989–1997 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 worked as R&D manager for Rolls-Royce Marine Deck Machinery. Since 2002 he has been employed in teaching at Aalesund University College, developing and teaching courses in marine product and systems design on bachelor and master level. From January 2010 he received the green ship machinery professorship. His special research interest is within the field of energy and environmental technology, with focus on combustion engines and the need for more environmental friendly and energy efficient systems.



Prof. Dr. Sergey Ushakov received his PhD degree in 2012 from the Department of Marine Technology at NTNU for the work on measurement and characterization of particulate matter emissions from marine diesel engines. Before re-joining the Department in 2016 as professor in marine machinery, for several years worked in MARINTEK (currently SINTEF Ocean) within the fields of marine diesel engine emission characterization and emission reduction technologies covering both volatile and non-volatile exhaust emissions. During this work he was involved

in a number of bigger and smaller research projects where accumulated substantial experience with experimental work both in laboratory and on board of different vessels. The current research focus is environmentally friendly shipping as well as improvement of marine diesel engines efficiency especially emphasizing the experimental part of this work.



Prof. Dr. Houxiang Zhang (M04-SM12) received Ph.D. degree on Mechanical and Electronic Engineering in 2003. From 2004, he worked as Post-doctoral fellow, senior researcher at the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Germany. In Feb. 2011, he finished the Habilitation on Informatics at University of Hamburg. Dr. Zhang joined the NTNU (before 2016, Aalesund University College), Norway in April 2011

where he is a Professor on Mechatronics. Dr. Zhang has engaged into two main research areas: 1) Biological robots and modular robotics, especially on biological locomotion control, 2) Virtual prototyping in demanding marine operation. He has applied for and coordinated more than 20 projects supported by Norwegian Research Council (NFR), German Research Council (DFG), and industry. In these areas, he has published over 160 journal and conference papers as author or co-author. Dr. Zhang has received four best paper awards, and four finalist awards for best conference paper at International conference on Robotics and Automation.