

Data Extraction and Deep Learning Method for Predictive Maintenance in Vessel's Engine Room

Cédric Seguin, Yohann Rioual, Jean-Philippe Diguet, Guy Gogniat

▶ **To cite this version:**

Cédric Seguin, Yohann Rioual, Jean-Philippe Diguet, Guy Gogniat. Data Extraction and Deep Learning Method for Predictive Maintenance in Vessel's Engine Room. 32nd European Safety and Reliability Conference (ESREL 2022)., Aug 2022, Dublin, Ireland. pp.1983-1990. hal-03769035

HAL Id: hal-03769035
https://hal.science/hal-03769035

Submitted on 15 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proceedings of the 32nd European Safety and Reliability Conference (ESREL 2022)
Edited by Maria Chiara Leva, Edoardo Patelli, Luca Podofillini, and Simon Wilson
©2022 ESREL2022 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-5183-4_S05-02-271-cd

Data Extraction and Deep Learning Method for Predictive Maintenance in Vessel's Engine Room

Cédric Seguin
Lab-STICC, Universite Bretagne Sud, Lorient, France. E-mail: cedric.seguin@univ-ubs.fr

Yohann Rioual
Lab-STICC, Universite Bretagne Sud, Lorient, France. E-mail: yohann.rioual@univ-ubs.fr

Jean-Philippe Diguet
CROSSING, CNRS, Adelaide, Australia. E-mail: jean-philippe.diguet@cnrs.fr

Guy Gogniat
Lab-STICC, Universite Bretagne Sud, Lorient, France. E-mail: guy.gogniat@univ-ubs.fr

Regardless of the sector of activity, industrial maintenance is a critical component for any company. It allows to not only limit breakdowns but also to prevent possible failures. Systems' maintenance improves productivity, and profitability of a factory. In maritime field, maintenance is ruled by context factors such as limited storage places, long period without confident restocking or external help from systems' experts, etc. Thus, engine officers need to rely on the best possible organization of the maintenance. Predictive maintenance would be a useful tool providing an efficient assistance by anticipating anomalies and failures. This article presents two contributions: 1) a digital twin, of a sub-system, to generate nominal and non-nominal data and, 2) the use of a machine learning based method to predict failures and provide decision making support to maintenance officers.

Keywords: Predictive Maintenance, digital twin, remaining useful life, Vessel Machinery, machine learning, deep learning

1. Introduction

Maintenance is an essential operation in the maritime domain as in all industrial sectors since it ensures systems' reliability and security. Furthermore, maintenance costs represent significant parts of the industrial budget and exploitation costs and so must be addressed with strong attention. Nowadays, the logistics sector relies on intensive and efficient exploitation of transportation resources, which means that availability of operational vessels is highly critical.

1.1. *Maintenance in maritime field*

Systematic maintenance is always considered in maritime field because parts storage areas are limited, and failures can have critical consequences on the system and especially on people. Predictive maintenance perfects maintenance organization and maximizes the time at sea while mitigating downtime at the shipyards as well as reducing maintenance costs. Predictive maintenance is therefore a necessity as it corresponds to a major industrial challenge.

The Seanatic project (Sea Analytic Connected Boat) aims to provide a maintenance assistance tool to anticipate functional breakdowns and improve the logistics of the intervention phases, limiting technical stoppages to increase days at sea. The work presented in this paper is part of this project.

1.2. *What is predictive maintenance?*

Predictive Maintenance (PdM) is based upon the philosophy that "if it is not broken, don't fix it" (Jimenez et al. (2020)). PdM consists in monitoring the system and using historical data to detect trends of equipment' behavior in order to predict when it is more likely to fail. Once failure trends are identified, and the timing of failure correctly predicted, maintenance tasks may be planned-

ahead. In general, equipment shows early signs of malfunction before failures: increases above normal temperature, rise in vibration levels, drop in performance levels, increased noise level, current and voltage changes and so on. One of the main advantages of predictive maintenance is that it allows to plan maintenance rather than just reacting to potential failures, but also to address the root cause of the failure.

1.3. *State of the art*

1.3.1. *Machine learning for prediction data*

The rise of Machine Learning (ML) has also reflected on Predictive Maintenance. Data-driven predictive maintenance is now widely used for the detection, diagnosis and prognosis of faults (Vollert and Theissler (2021)). The prognosis task to predict an asset's remaining useful life (RUL) (Si et al. (2011)), i.e. the time span from the current time point to the end of the useful life, is considered as the most complex part of predictive maintenance, yet the most beneficial one. Predicting the time of failure enables not only cost-effective scheduling of maintenance, but also related activities like spare part provisioning.

Multiple reviews of ML-based predictive maintenance have been conducted in the recent years (Vollert et al. (2021), Lei et al. (2020), Ran et al. (2019) and many others). The domains of application of these methods are diverse: bearing (Zhang et al. (2020),Neupane and Seok (2020)), milling tool (Serin et al. (2020)) or jet engine (Che et al. (2019)). Recurrent neural network approaches are among the most popular approaches. Degradation is usually thought of as a slow process over time, and they excel because of their ability to model long-term dependencies.

1.3.2. *Digital twin in missing data cases*

The digitalization of the industry has lead to the emergence of new technologies such as Digital Twin (DT). The concept has been introduced by Glaessgen and Stargel (Glaessgen and Stargel (2012)) where a first definition was proposed. Since that, its definition continuously improves depending of the domains in which it is developed and operated (Barricelli et al. (2019), Jones et al. (2020)).

DT can be seen as the virtual representation of a process, an object, or a product. DT can be as simple as a static digital model, but it can also be more sophisticated like a dynamic model that replicates a complex behavior. DTs are widely used for conception, simulation, optimization, or maintenance in the industry.

Failure data are rarely available and yet necessary for predictive maintenance. The need for a Digital Twin arises from the fact that sensors data collected is not always adequate for RUL estimation. Thus we use a simulator for training maintenance officer as a digital twin to generate nominal and non-nominal data. We limit our experiments to a sub-system: a diesel-generator. This sub-system is critical for the vessel's operations and the maintenance team avoids letting it breaking down.

1.4. *Work positioning*

This paper presents the work aimed at building a maintenance assistance tool to anticipate functional failures and improve the logistics of the intervention phases, limiting the technical stops to increase the sea days. Two contributions are presented:

- A methodology to collect data in absence of faulty data using a digital twin solution of a diesel generator sub-system;
- The use of a deep learning approach to predict the need for maintenance of the diesel generator sub-system.

The paper is organized as follows. In Section 2, the data extraction using a digital twin is presented. The deep learning approach to estimate the RUL is discussed in Section 3. Concluding remarks and future research are discussed in Section 4.

2. Data extraction using a digital twin

The concept of digital twin (DT), in maritime field, is introduce in this first section. To address the absence of supervised data, a DT solution based on a marine training simulator for maintenance officers has been developed. We begin by

describing use-cases that enable us to build a number of representative datasets, then we explain the method used to generate unique datasets. Finally, we discuss the proposed methodology to generate supervised data and their impact on machine learning (ML) algorithms.

2.1. *Digital twin in maritime domain*

Maritime domain follows manufacturing domain in its digital revolution, as we see more and more technologies and connectivity solutions embedded in ships. For instance, engine manufacturers are a good example of system digitalization. Over the last 5 years, they have been improving engine control, monitoring and connectivity as high efficiency, low gas emissions, and safe operation of their systems are mandatory. They proposed their own support and artificial intelligence solutions with data they collected for years. Multiple proprietary solutions have been developed such as the Expert Insight Technology by Wärtsilä (2021), or the Caterpillar initiative started in 2015, with the acquisition of ESRG (Marr (2017)) that provides a software suite for the remote monitoring and diagnostics. However, datasets used for developing their technologies are not public, yet.

2.1.1. *Why using a digital twin?*

Operating data as data logs are most of the time made confidential by ship-owners, when they are not missing. Data intellectual property (IP) is a key concern in every domain of artificial intelligence. Maritime domain is not an exception, and so the data access has to be considered as a real challenge.

A second issue is related to supervised data. These specific datasets require long record campaigns to acquire a significant information. Even so, due to automatic maintenance operations, recorded data can be considered as censored. Supervised data is therefore harder to obtain in real exploitation context, and requires specific experiments on these systems.

A DT approach has been introduced to tackle the IP problem and deal with the absence of nominal and non-nominal data to train machine learning models. DT can be a good alternative to generate representative datasets, as it accurately models the system behavior. To facilitate our data generation process, a simulator to train vessels' maintenance officers has been used as DT.

2.2. *Using a simulator as a digital twin*

As mentioned in Section 2.1.1 data is a crucial part in supervised machine learning technologies, and its access can be challenging because of confidentiality and availability issues.

To build labeled datasets, generated data are exported from an engine room simulator: K-Sim by the Norwegian's company Kongsberg (2021). This commercial solution is usually used to train future maintenance officers. Kongsberg's simulator allows us to shape scenarios composed of nominal or non-nominal data. The possibilities offered by the simulator are discussed in Laskowski et al. (2015).

K-Sim Engine models benefit from a long-term development based on actual ship specifications. Systems' behavior is accurately represented on an overall level as well as on a more specific subsystems level. Furthermore, the simulator integrates emergency situations by selecting the level of anomaly of multiple parameters.

Every system and sub-system is represented in the simulator by a page with its parts, sensors, and fluid circuits. In the diesel generator (DG) subsystem case, we can simulate a turbo-compressor, several valves, pumps and filters or coolers alongside lube-oil, fuel-oil, fresh water, or exhaust gas circuits.

Multiples variables, depicting physical values from sensors, are associated to those items. In DG's example, engine revolution per minute, fluids temperature, pressure, and throughput, or electrical power, voltage, intensity, frequency, are observable. Finally, over 150 different variables can be monitored for the DG sub-system.

As K-Sim engine main purpose is to train marine officers through different emergency situations, each system and sub-system offer a list of malfunction variables. These special variables can be set to create anomalies that impact the nominal behavior of the current system.

Thanks to the support of Kongsberg's service,

we use an additional proprietary software, called *Trend Log and Display System*, to export logs as csv files. A strategy for collecting failure data was developed based on the combination of the simulator and the data logger. Collected data are used to compose datasets to train our model.

To simplify the training and evaluation of our machine learning algorithm, five case studies has been defined in collaboration with experts from ENSM (2022), taking into account the capabilities of the simulator. These scenarios, recommended by the experts, represent use cases close to reality. The next subsection will go through the context and picked scenarios.

2.3. *Case studies to build operational context to data logs*

K-Sim Engine simulates multiple engines, from low-speed to high-speed boats. We focused on the MAN B&W 5L90MC VLCC L11-V which simulates a very large crude carrier of 305 meters, with a slow speed turbo charged diesel engine.

K-Sim engine models over a thousand of sensors from tens of systems. To limit our experiment complexity, and the amount of exported data, we focus on a sub-system: the diesel generator (DG). The DG is a diesel engine that generates electrical energy. DG is easier to apprehend than the main engine, while its physical models are quite similar and still well designed in Kongsberg's simulator. Over 90 variables are monitored from this subsystem, continuously.

To complete our approach, and to provide context to our generated data, we identified with experts, senior officers and tutors, five relevant scenarios:

- turbocharger dirty
- lube-oil filters dirty
- fuel-oil filters dirty
- high temperature fresh water cooler dirty
- low temperature fresh water cooler dirty

Each scenario offers the possibility to select a level of anomaly from 0 to 100% of the failure. This malfunction capability brings the opportunity to monitor nominal and non-nominal data. Nevertheless, multiple heterogeneous datasets are required to train ML algorithms.

2.4. *Datasets building method*

Artificial intelligence algorithms require many heterogeneous data for training its models. A simulator is not suited to generate hundreds of unique datasets. To do so, a methodology based on segments of data, exported from the simulator, has been developed. We take advantage of the malfunction variables to create different segments of data, that we combine to create unique sequences. This methodology is detailed in this sub-section.

Previous subsections 2.2 and 2.3 showed how we export data and scenarios we pick to train our models. Adjusting malfunction level allows to shape and form specific data sequences. However, we cannot simply replicate our scenario to get heterogeneous datasets. K-Sim engine is based on mathematical models, and basic replication will give the same results every time. Malfunction level sequence has to be changed from a log to another in order to get unique records. To illustrate this, in a first attempt, a malfunction level of 10% is set after 10 minutes, while in a second attempt, the same event will arrive after 20 minutes.

K-Sim engine brings the functionality to automatize the malfunction level. We can, that way, schedule various automated sequences to get different records of data. However, it will require a huge amount of time to prepare those sequences, without considering, the time needed to play and export all logs. K-Sim engine is a training simulator that makes trainees in front of a virtual engine room. While a possibility to accelerate time has been added to the simulator, it has not been developed to quickly generate thousands of datasets.

To overcome this technical blockage, chunks of data for each level of malfunction, for every picked scenarios, are generated separately. The obtained subsequences of data are aggregated them together following a random pattern. A malfunction step of 10% has been arbitrarily chosen to build 10 segments of malfunction, from 10% to 100%, for each scenario. As the simulator initializes from a nominal state, when we play a malfunction segment it takes around few seconds to stabilize its data to the selected setpoint. First

hundreds data are discard to remove interference.

Finally, a Python script associates to the 11 segments generated, from 0 to 100%, a random number. This number represented how many times the specific chunk is played in the sequence to form a complete unique record of data. Figure 1 shows alongside the turbocharger speed variable, in krpm, for five different datasets automatically generated from data segments. In the last scenario (purple), nominal segment is played 8 times, 10% segment is played 10 times, 20% segment is played 1 time, and so on.

In those turbocharger datasets, the fourth one (purple) shows a fault corresponding to 10% of the clogging near the seven-hundredth sample. Since the failure occurrence, the speed of rotation is decreased as difficulties to exhaust gas appear. The next step, being 20% of failure, arrives close to the one thousand six hundredth sample. Situation gets worst because the system is getting dirtier at every stage we can see through the graph. Stages are due to the collection strategy we used with the simulator.

As we get many heterogeneous datasets, this approach presents some limitation discussed in the next subsection.

2.5. *Discussion about synthetic data*

Limits has been raised during our datasets generation process.

As much as we get unrepeatable datasets with our method, we are still far from real-life unpredictable models. We train machine learning models specific to Kongsberg's simulator models. Moreover, sensors position, and dynamic are characteristic to this ship case. ML models will need an update before being adapted to another use case. Nevertheless, these datasets brought the required data to theoretically confirm our machine learning algorithm, on representative scenarios.

The next step will be to expose our ML algorithm to real data. To do so, A 44 meters boat is dedicated to experiments and is currently being fitted with all the required instrumentation. This experiment is planned as future work.

3. Deep Learning approach for RUL estimation

Predicting the remaining useful lifetime (RUL) is difficult; we need a reliable prediction that is understandable to the maintenance officer who will choose the maintenance date. The proposed method is a two-step process: in the first step, kernel principle component analysis (KPCA) (Schölkopf et al. (1997)) is applied for non-linear features extraction and then a recurrent neural network approach called Weibull Time to Event Recurrent Neural Network (Martinsson (2016)) is used to predict the Weibull distribution of the failure risk.

3.1. *Proposed approach*

3.1.1. *KPCA*

Dimensionality affects neural networks. The more dimensions there are, the more data we will require to train our network. Furthermore, the data generated by the sensors is noisy and frequently redundant. The objective of KPCA is to reduce dimensionality and to extract non-linear features, retaining useful information and removing redundant information. KPCA is an extension of principal component analysis (PCA) using techniques of kernel methods.

3.1.2. *Gated recurrent unit (GRU)*

GRU was introduced by Cho et al. (2014) and is part of the recurrent neural network (RNN) architecture. A RNN is a type of artificial neural network which uses sequential data or time series data. They take information from prior inputs to influence the current input and output. In order to keep the information, they have two gates: a reset gate and an update gate. These gates regulate the flow of information that is required to predict the network's output.

We will describe how the activation of the j-th hidden unit at time step t is computed. In the following equations, x and h are respectively the input vector and the output vector. W and U are weight matrices which are learnt. Firstly, the reset gate r_j is computed by :

$$r_j = \sigma([W_r x]_j + [U_r h_{t-1}]_j)$$

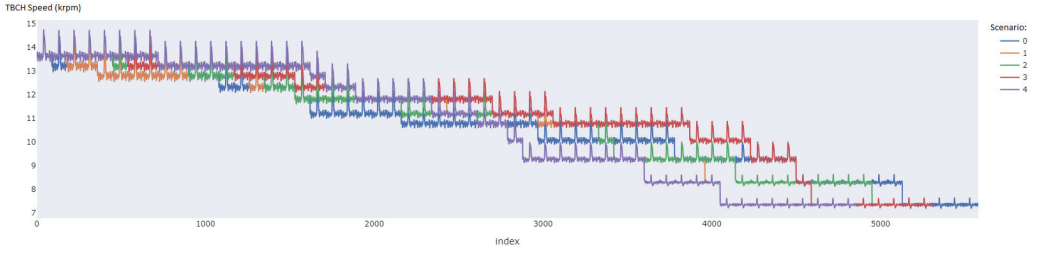


Fig. 1. Generated datasets of turbocharger speed

where σ is the sigmoid function and $[.]_j$ denotes the j-th element of a vector. Similarly, the update gate z_j is computed by:

$$z_j = \sigma([W_z x]_j + [U_z h_{t-1}]_j)$$

The actual activation of the proposed unit h_j is then computed by:

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\tilde{h}_j^t$$

where $\tilde{h}_j^t$ is generated by r_j with a $tanh$ layer. The function is:

$$\tilde{h}_t = tanh([Wx]_j + [U(r \odot h_{t-1})]_j)$$

where the operator $\odot$ denotes the Hadamard product.

In this formulation, when the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only. This effectively allows the hidden state to drop any information that is found to be irrelevant later in the future, thus, allowing a more compact representation. On the other hand, the update gate controls how much information from the previous hidden state will carry over to the current hidden state.

3.1.3. *WTTE-RNN*

The Weibull Time to Event Recurrent Neural Network is a state-of-the-art approach based on survival analysis and deep learning. This method is used to predict the parameters of a Weibull distribution corresponding to the survival function of the diesel generator sub-system based on the KPCA result. This function is used to determine

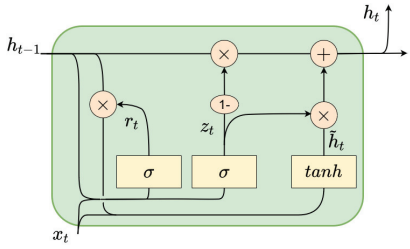


Fig. 2. Frame of GRU

when to schedule maintenance. The Weibull distribution is widely used in reliability analysis because its flexibility allows it to model various data.

The probability density function of a Weibull random variable is :

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases} \quad (1)$$

where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution.

This method can work with censored and non-censored data. Censored data is any data for which we do not know the exact event time. On the other side, we know the exact event time for non-censored data. In the context of reliability engineering we typically refer to events as "failures".

It is common to refer to data as "times" but a variety of other units of measure may also be used such as cycles, rounds, landings, etc. This depends on the type of data being collected and the units in which the life is measured. In this paper, "times" refers to the measurement period.

3.2. *Results*

There are 912 scenarios in the dataset. We train the model on five different failures with only one failure each time. 274 of the 912 scenarios are used in the test sets, while the rest are used in the training sets. The KPCA extracts only 5 components, and the neural network is made up of 10 units. We use Adam (Kingma and Ba (2014)) as the optimizer, with a learning rate of $1e-4$. We use an early stopping strategy to determinate the appropriate number of episodes to avoid overfitting. As result of learning, we obtain a loss of 7.7492 on the training dataset and 7.7833 on the validation dataset.

To determine the RUL once the model is trained, we fixed a threshold R ($0 < R < 1$). It corresponds to the reliability threshold i.e., the probability that the system will fail later than the predicted RUL. The RUL is determined using:

$$\text{RUL} = \lambda(-\ln(R))^{(1/k)} \tag{2}$$

where k and λ are the parameters of the distribution.

The goal of the threshold is to collaborate with the maintenance officer to determine the best value of R. This, however, is outside the scope of the paper.

The outcome of the RUL prediction with $R = 0.5$ for various failures is shown in Figure 3. The green dashed line represents the optimal prediction. The actual RUL is shown by the y-axis, while the expected RUL is represented by the x-axis. There are five failures, each of which corresponds to a different color. The predictions failure are centered around the optimal RUL with a threshold $R = 0.5$ and there is no distinction made by the model based on the type of failure. The performance of the model is less important than the ease with which a person can interpret its predictions.

4. Conclusion

The Seanatic project seeks to provide a maintenance support tool to predict functional breakdowns and enhance the logistics of the intervention phases, limiting technical stoppages and increasing days at sea. In this paper we present two contributions:

- For the diesel generator, a way to collect data in the absence of inaccurate data utilizing a digital twin solution;
- The use of a deep learning method to predict the remaining useful lifetime in order to schedule the maintenance of the diesel generator.

Datasets extracted, using our first contribution, were used to evaluate our deep learning algorithm. Our digital twin have generated sufficient amount of significant data to train models. Our second contribution has proven its efficiency to anticipate system's failure. This last result demonstrates how such an approach can help maintenance officers to plan their operations to extend equipment lifetime.

A 44 meters boat is dedicated to experiments and is currently being fitted with all the required instrumentation. Future perspectives will be the integration of our approach on a training ship.

Acknowledgement

The research presented in this paper is carried out in the context of the SEANATIC Project (N°2082C0023). This project is supported by the Future Investments Program (PIA) operated by ADEME (the French Environment and Energy Management Agency). Teachers from the École Nationale Supérieure Maritime (ENSM) contributed to our work bringing their technical and mechanical expertise to our project.

References

Barricelli, B. R., E. Casiraghi, and D. Fogli (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access 7*, 167653–167671.

Che, C., H. Wang, Q. Fu, and X. Ni (2019). Combining multiple deep learning algorithms for prognostic and health management of aircraft. *Aerospace Science and Technology 94*, 105423.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

ENSM (2022). École nationale supérieure maritime. `https://www.supmaritime.fr/`.

Glaessgen, E. H. and D. Stargel (2012). The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference-*

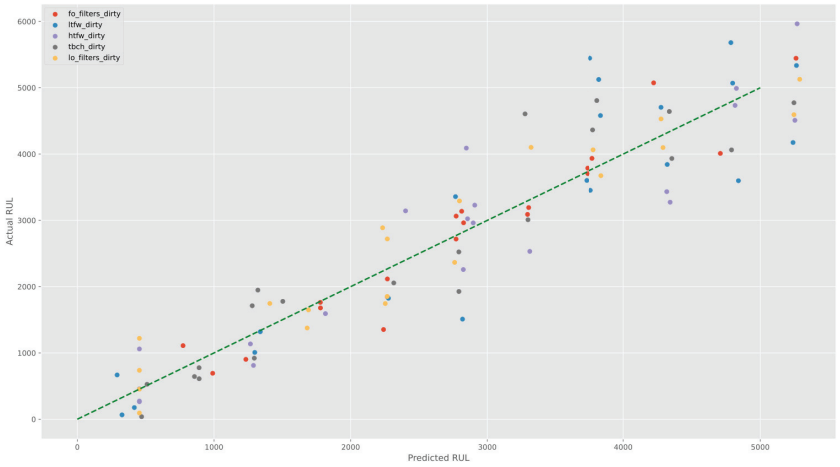


Fig. 3. Predictions of the proposed approach on a test dataset

Special Session on the Digital Twin, Number NF1676L-13293.

Jimenez, V. J., N. Bouhmala, and A. H. Gausdal (2020). Developing a predictive maintenance model for vessel machinery. *Journal of Ocean Engineering and Science 5*(4), 358–386.

Jones, D., C. Snider, A. Nassehi, J. Yon, and B. Hicks (2020). Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology 29*, 36–52.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization.

Kongsberg (2021). K-sim engine simulator. `https://www.kongsberg.com/`.

Laskowski, R., L. Chybowski, and K. Gawdzińska (2015). An engine room simulator as a tool for environmental education of marine engineers. In *New Contributions in Information Systems and Technologies*, pp. 311–322. Springer.

Lei, Y., B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap.

Marr, B. (2017). Iot and big data at caterpillar: How predictive maintenance saves millions of dollars. *20*, 2020.

Martinsson, E. (2016). WTTE-RNN : Weibull Time To Event Recurrent Neural Network. Master's thesis, Chalmers University of Technology & University of Gothenburg.

Neupane, D. and J. Seok (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access 8*, 93155–93178.

Ran, Y., X. Zhou, P. Lin, Y. Wen, and R. Deng (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS XX*.

Schölkopf, B., A. Smola, and K.-R. Müller (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pp. 583–588. Springer.

Serin, G., B. Sener, A. M. Ozbayoglu, and H. O. Unver (2020). Review of tool condition monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology 109*(3), 953–974.

Si, X.-S., W. Wang, C.-H. Hu, and D.-H. Zhou (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research 213*(1), 1–14.

Vollert, S., M. Atzmueller, and A. Theissler (2021). Interpretable Machine Learning: A brief survey from the predictive maintenance perspective. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2021-Septe*.

Vollert, S. and A. Theissler (2021). Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*.

Wärtsilä (2021). Expert insight – enable predictive maintenance. `https://www.wartsila.com/`.

Zhang, S., S. Zhang, B. Wang, and T. G. Habetler (2020). Deep learning algorithms for bearing fault diagnostics—a comprehensive review. *IEEE Access 8*, 29857–29881.