Vibration Edge Computing in Maritime IoT

Michala, A. L.

Vourganas, I. J.

Coraddu, A.

The definitive Version of Record was published in:

ACM Transactions on Internet of Things 3(1), article 6.
https://doi.org/10.1145/3484717

Vibration Edge Computing in Maritime IoT

Anna Lito Michala [*], Ioannis Vourganas [†], Andrea Coraddu [‡]

27 October 2021

Abstract

IoT and the Cloud are among the most disruptive changes in the way we use data today. These changes have not significantly influenced practices in condition monitoring for shipping. This is partly due not only to the cost of continuous data transmission. Several vessels are already equipped with a network of sensors. However, continuous monitoring is often not utilised and onshore visibility is obscured. Edge computing is a promising solution but there is a challenge sustaining the required accuracy for predictive maintenance. We investigate the use of IoT systems and Edge computing, evaluating the impact of the proposed solution on the decision making process. Data from a sensor and the NASA-IMS open repository was used to show the effectiveness of the proposed system and to evaluate it in a realistic maritime application. The results demonstrate our real-time dynamic intelligent reduction of transmitted data volume by $\sim 10^3$ without sacrificing specificity or sensitivity in decision making. The output of the Decision Support System fully corresponds to the monitored system's actual operating condition and the output when the raw data is used instead. The results demonstrate that the proposed more efficient approach is just as effective for the decision making process.

Keywords: Edge computing, IoT, big data analytics, vibration analysis, vessel condition monitoring

1 Introduction

Edge Computing is quickly becoming the strategy of choice in many IoT applications. With IoT devices predicted to reach the space of Trillions in the next decade the existing data management strategies such as Cloud Computing will quickly run out of resources and capacity [56]. In this scenario, network resources are consumed by contextual data creating severe strain on centralised architectures [18] making traditional management approaches unusable [4]. A compelling application for Edge computing within IoT systems is presented in the maritime sector where targeted monitoring which combines several methods is needed to assess operating condition of vessels and health status of the machinery [60]. Safety and environmental regulations demand correct operating condition and maintenance [13, 12, 39, 46, 43]. This is to reduce unexpected failures with impact on safety, loss of ship, detrimental effects on the local environment, and casualties [19, 33, 38, 9]. As a consequence, for a single vessel, 10% − 15% of the operating budget is dedicated to maintenance [10, 49, 57]. Compared to other Edge computing applications the maritime sector introduces new challenges:

[*] annalito.michala@glasgow.ac.uk, ORCID:0000-0001-7821-1279, University of Glasgow, Glasgow, UK

[†] ORCID:0000-0002-3433-3757, Abertay University, Dundee,UK

[‡] ORCID:0000-0001-8891-4963, University of Strathclyde, Glasgow, UK

- **The edge environment is less dynamic and connected.** Networking constraints are heightened and a continuous link to the cloud is not feasible for prolonged periods of time when the vessel is in operation. Moreover, major maintenance actions cannot be taken during this time.

- **The edge system has higher security and reliability requirements.** Due to difficulties in maintaining the system during vessel operations the deployment the edge system should account for a higher degree of hot and cold redundancy linked to increased cost.

- **The edge has higher independence; in computational capacity and power.** The edge can be easily served by the on vessel power supply and dedicated server room space in terms of resources and storage.

- **The bandwidth for data transfer is a limiting factor.** This limitation impacts the volume of data that can be transferred from ship to shore while there is a high volume and real-time requirement to support accurate decision making.

IoT and Edge Computing is recognised as one of the main technology trends in this industry [60]. The state-of-the-art is achieved through continuous Condition Monitoring (CM), predictive maintenance (PM) and remote visibility. Usually, a centralised method is employed for PM approaches, where data is processed at shore often after manual extraction from the monitored systems (e.g. hand held equipment or USB [32]). Here, an additional challenge is introduced through incomplete datasets (i.e. lack of correctness and continuity) [2].

The shipping industry has been resilient to change [63] in adoption of IoT, Cloud and Edge Computing methods which are currently recognised as a significant opportunity [60]. In 2012, 12% of the global fleet used CM while only 2% used the data for automated maintenance management [53]. A staggering 60% of CM applications have not matured to incorporation [45]. Several barriers are often recognised including trust in the technology [8], data transfer to shore [31], Internet usage, communications costs and availability [1]. However, the growth of digital intelligence applications worldwide has increased awareness and demand in the maritime industry for collaboration and data-sharing to decrease inefficiencies and redundancy [17, 34], increasing the transmission volume. Considering the cost of real-time transmission for larger volumes of data, more inclusive efforts towards reducing unnecessary information are needed [20, 47].

There is however one main challenge introduced by these new methods; the impact on predictive accuracy of the operational condition and thus maintenance management decisions. In this work we propose an Edge Computing system architecture that can achieve the required reduction without impacting on decision support outputs. To this aim, the authors present a framework that enables the fast access and transfer of data across networks and systems while generating information for easier access and analysis. The proposed framework's aim is realised through a lossy real-time intelligent volume-reduction algorithm that converts data into actionable information.

Advancing Edge computing in this application domain while considering the domain specific constrains the novelty of the proposed approach lies in the real-time, dynamic data analysis on the Edge device to pick out events and changes that are necessary to influence the decision support.
Our work is focusing on a specific niche of Edge Computing, contributing to systems for less dynamic and more constrained applications in terms of network infrastructure while maintaining very high requirements of accuracy for decision making. Compared to the state-of-the-art the contributions of this paper are:

1. *Real-time Dynamic reduction* of the transmitted volume of data by $\sim 10^3$ without impacting decision support outputs of the Edge and Cloud Server application.

2. *Outperforming* lossy compression techniques.

3. *Measurement agnostic* algorithms for IoT (Edge device) processing of data streams for ship CM.

4. *Simultaneous Edge processing* for volume reduction and event extraction (intelligence) of several data streams within the capabilities of a single IoT device to ensure continuity and correctness of data.

The rest of the paper is organised as follows. The state-of-the-art in ship CM, data compression and Edge computing approaches is presented in Section 2. Section 3 presents the use of higher computation capability IoT systems for Edge computing and transmission of extracted information while managing the recognised barriers. Further, we evaluate the impact of the proposed solution on the decision making process. We present the results of the evaluation in Section 4 and discussion in Section 5 with conclusions and future work in Sections 6 and 7 respectively.

2 Related Work

In new build vessels a variety of control/monitoring systems is available [54]. Those systems support applications such as: CM, control, online documentation, trending, maintenance check lists and training through simulation videos, weather, sea conditions, voyage data [59, 60]. The state-of-the-art in ship CM data processing falls within one of two embedded system categories [21, 60]:

1. collection and transmission,

2. immediate processing and actuation without transmission.

A wireless collection solution is proposed in [29], where an Edge gateway [60] can provide temporary services until Cloud connectivity is available. However, all data is processed on the Cloud, creating high operational cost and intermittent visibility. Additionally, the computation capabilities and memory of the embedded system are limited potentially leading to loss of data, handling error readings and system introduced errors in data stream.

An immediate processing solution, towards decentralised vibration data processing, is presented in [62]. However, Fast Fourier Transformation (FFT) is the only operation performed, time domain information is not collected, and only one type of input is possible from one particular sensor. All this restricts the applicability of the system and increases the error margin.

On the other hand, pre-processing is used in a variety of applications [30]. A closely relevant example is CM in aviation, where it is found that collected raw data do not provide any additional information [37]. It is reported that less power is required for pre-processing than transmitting the raw data, often in the form of lossless compression, and the output of the analysis is suitable for further processing [17].

Further expanding pre-processing methods, Edge computing is reported to reduce traffic by 95% [61]. In recent years Edge computing has focused on moving Cloud services closer to the Edge of the network, often on Edge Servers. In [3], the Cloud service is executed on software defined networking infrastructure to process data locally when the Cloud is unavailable. However, this method still requires the raw data to be transmitted to the Edge server; while in ship CM low network utilisation is one of the main requirements.

In another direction, Edge computing approaches aim to utilise the increased computational capacity of IoT but meet requirements of power consumption, latency and response rates [15]. Towards this direction a method that increases the computational capacity at the data collection device is presented in [25]. However, the analytics machine learning component needs to be retrained for every installation location, making the solution non-portable. Hence, a method that utilises the best of both pre-processing and Edge computing would be more appropriate aiming to reduce the volume of transmitted data without reducing higher level information density.

Well established approaches for volume reduction without compromising information density are compression or information extraction. Commercial lossless compression tools include LZ77, bzip2, GZip and RAR [44]. Many compression methods focus on video and image data [50], text [44], or all combined. Lately, specialised hardware for IoT lightweight data compression has been investigated with a focus on lossless [28]. Due to fundamental differences in the data forms, signal processing approaches are more relevant to CM. To this end, a systematic review was performed to identify the most recent approaches in: (1) data stream compression, (2) lossy compression, (3) Edge computing for IoT data. To enable robust comparability the literature search targeted only one code base, Github. This decision enabled the identification of state-of-the-art algorithms in academic and industrial use while enabling reproducibility of research and experimental evaluation for the input. The search returned 38, 313 and 14 results respectively. The inclusion criteria were: (1) suitable for data stream, scientific data, or generic algorithm. The exclusion criteria were: (1) suitable for image, or text. In total 100 git repositories were screened. The screening process eliminated repositories that were not recently updated (cut off date 2015). From each search 3, 7 and 3 repositories were selected respectively, in total 13. Finally, 8 were successfully compiled on our target platform and used for review in this section and evaluation in Section 4. From these, 5 have associated academic publications, 1 has an associated book publication, and 1 is a standard. These are representative examples from each category and 5 are generally recognised compression algorithms used by commercial tools.

The lossless compression algorithms that are most relevant to data stream compression are SLDC [11], Huffman coding [23], the fast LZ77 [24] and the general purpose bzip2 [51]. Lossless commercial tool compression ratio (CR) for CM achieves $CR = 3 : 1$ [30], while lossy approaches can achieve higher reductions (e.g. $92 : 1$ without affecting the decision-making process [55], $10 : 1$ for sensor data [30]). Thus, lossy approaches are appropriate [30] for lower bandwidth consumption but do not address the challenges of continuity and correctness of data.

Recently research has intensified in lossy approaches targeting Edge computing applications. In [58], the authors expect the devices to support Cuda and provide access to a GPU which is expected to be more popular for Edge servers in the coming years. However, GPU hardware for IoT Edge devices is not yet mainstream. On the other hand, in [5], the removal of noise is identified as an opportunity for higher compression rates on IoT devices. An important consideration is that pre-processing and lossy approaches traditionally have a trade off resulting in sacrificing both specificity and sensitivity in the resulting data. Additionally, a change in operational condition (e.g. switched off), or failure (e.g. data acquisition equipment, software update) can introduce data that impact the decision support process; false negatives/positives [2].

In parallel, research has focused on moving intelligence to the Edge. Information extraction on the Edge can identify events, monitor continuity and correctness and reduce volume of transmitted data through a single computation. Many methods are reported in literature for CM and vibration data information extraction [36, 6], and depending on the recorded data a processing method is selected. In particular, correlation based methods or machine learning methods have been used for ship CM [7]. Their aim is prediction of time to failure and time to repair [52]. However, these approaches require higher power, extensive processing capabilities and are better suited to post-transmission feature extraction [14] often at an Edge Gateway or Server [3, 60].

We identify an opportunity in the merge of these approaches: compression, pre-processing and Edge computing. We propose the implementation of real-time, dynamic analysis that intelligently

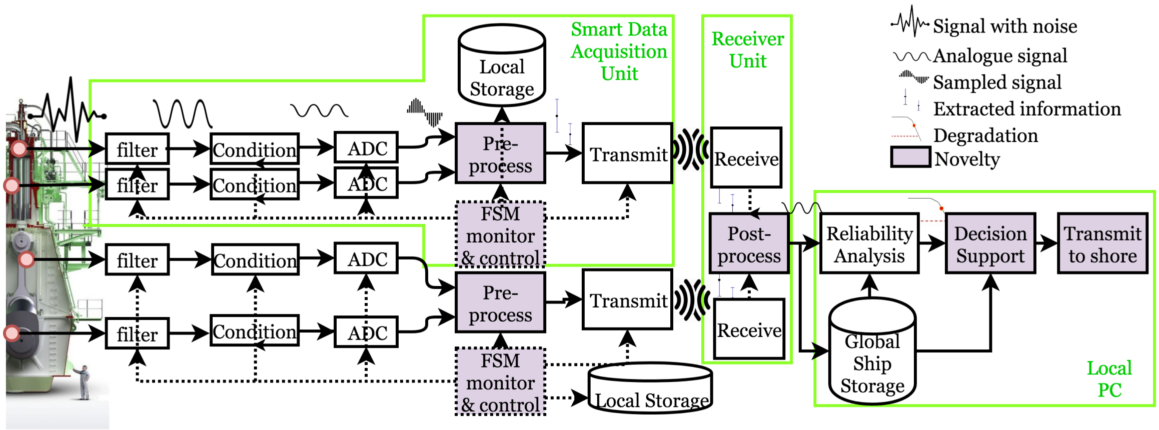


Figure 1: System Architecture - Edge Components.

compresses transmitted volume. We use trend analysis, FFT and Time Domain Analysis at the Edge of the network, consuming minimal data processing and transmission resources while providing volume reduction exceeding lossy compression methods, and ensuring data continuity and correctness. We claim that the proposed approach does not sacrifice sensitivity or specificity at the decision making level.

3 Methodology

3.1 System Architecture

The system architecture is presented in Fig. 1. Two hardware components are demonstrated, the IoT data-acquisition/Edge-computing component and the receiver/Edge-Server unit. The main processing capacity is provided by a Beaglebone Black (BBB) commodity hardware platform. The receiver unit comprises the physical antenna connected directly to the main PC onboard the vessel along with the developed Decission Support software.

We have previously developed a Decision Support System (DSS) presented in [40] which is deployed on the Edge server. The interface of the DSS can be deployed on the Cloud to provide remote visibility. The DSS translates the extracted information to predicted failures, maintenance actions for engineers onboard the vessel, and direct cost impact (benefit identification). This addresses both benefit identification and reduced training overheads as discussed in [42, 40]. The DSS is integrated in the system presented in this paper.

The software stack (Fig. 2) is separated in the IoT/Edge-processing component, the monitoring and control component, the post-processing component and the reliability, DSS and transmission to shore component. The components are distributed between IoT and Edge Server as described in Fig. 2. This architecture allows a sensor-server type of architecture bringing some of the Cloud intelligence to the very IoT device that captures the data. This distributing of the data processing to various locations is possible due to the increased computational capacity but also due to the ease of access to mains. The proposed system architecture achieves significant reduction of network traffic over the wireless channel but also enables the IoT device to identify higher order events through multiple data sources.

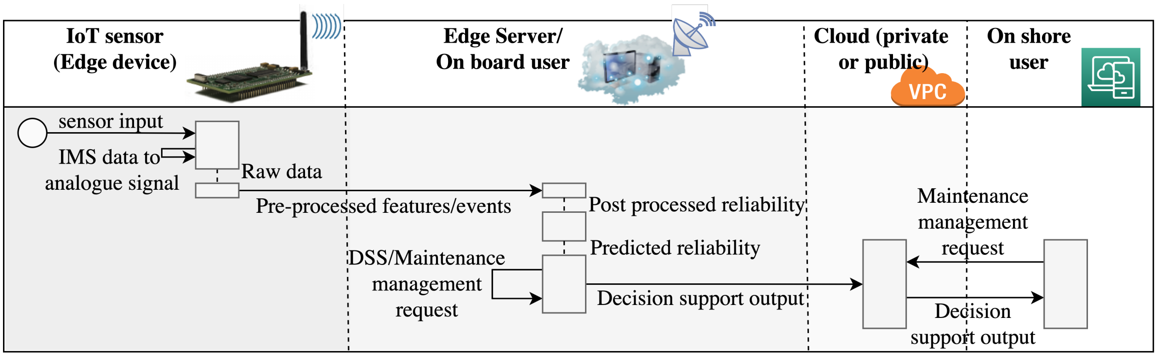


Figure 2: Data flow through the system from virtual and physical sensors on the IoT Edge device to DSS output generated at the Edge and Cloud servers supporting onboard and on shore user applications.

The Edge-processing component is an information extraction algorithm that aims to capture the statistical profiles of the attached measurements and identify events such as normal operation, alarm conditions, IoT device malfunction and internal errors. On the other hand the post-processing tool is tasked with informing the further processing units of any events that would compromise continuity or correctness of the sourced data so that the DSS tool is not affected by data that should be ignored.

3.2 Edge Computing algorithms for Ship CM

A new approach for Edge and post processing is proposed and applied for vibration data as time and frequency spectrum information is extracted along with events in a single pass over the incoming data stream (Fig. 2). The Edge-processing and post-processing algorithms are presented in Algorithms 1 and 2 respectively.

The Edge processing algorithm reads a 10 minute window of incoming data points from each stream. The window is then supplied to the FFT function which translates the incoming data to their corresponding frequency spectrum series. The frequencies are shorted according to amplitude from highest to lowest. Only the top 1% of the sorted frequencies (i.e. those with the highest amplitudes) are retained for further transmission. The remaining are discarded. This percentage was selected in the interest of minimising the transmitted data volume.

Further, the time series is profiled and statistical features extracted and retained for transmission. These properties are the amplitude $peak-to-peak$, $mean$ (μ), $standard\ deviation$ (σ), $root\ mean\ square$ (RMS) and $variance$ (σ^2). Finally, the raw time series is stored in local memory. Local memory is overwritten in a circular buffer approach when full. Extraction of the raw time series is possible manually if necessary.

As this paper focuses on the reduction of the transmitted volume the event extraction steps of the algorithm were omitted for simplicity and clarity. However an approach similar to the post-processing unit is used to perform these actions on the Edge node.

The post processing algorithm is utilising an open source tool that classifies vibration signals and maps them to reliability values [48]. A reliability value of 100% identifies an excellent operating condition for optimal performance, while a reliability of 0% identifies a system that has failed. The tool uses $k-means$ clustering to identify the proximity to normal or abnormal operation. Additionally, incoming events are used by the implemented post-processing algorithm to identify

```
Pre-process ADC output:
    foreach $window \in windows_n$ do
        while $sample_{length} \neq window_{length}$ do
            Read data in normal sampling rate;
        end
        Calculate $FFT_{new}$ ;
        $top_{1\%new}$ = Extract ($FFT_{new}$);
        calculate $pk - pk_{new}$, $\mu_{new}$, $\sigma_{new}$, $RMS_{new}$, $\sigma^2_{new}$;
        packet = Package( $timestamp$, $window_{length}$, $top_{1\%new}$,$pk - pk_{new}$, $\mu_{new}$, $\sigma_{new}$,
          $RMS_{new}$);
        Transmit (packet);
    end
return
```

Algorithm 1: Pre-processing vibration data inputs.

when predictions should not be updated; for example, if the IoT device has an internal error any data received is directly discarded and the prediction of the last valid input is retained.

```
Post-process Incoming traffic:
    Enable spectrum analyser extension for reliability;
    Provide Higher and Lower operational characteristics from manufacturer;
    foreach $packet \in Received$ do
        $class_k = k - means(packet)$;
        $\lambda = d(class_k, class_{threshold})$;
        $Reliability = e^{-\lambda * window_{length}}$;
        foreach $month := 0 \rightarrow 5$ do
            prediction = Predict(Reliability,prediction);
        end
    end
return
```

Algorithm 2: Post-processing transmitted packets; where $d()$ and $k - means()$ are functions provided by the reliability extension of the spectrum analyser and $Predict()$ is provided by the reliability analysis tool.

3.3 The IMS Dataset

The IMS dataset from NASA's open repository [22] was used as input data source. Data of three run-to-failure experiments of bearings are presented in the set (Table 1). The dataset is collected from four bearings mounted in series on an endurance test rig. The failure causes as identified by the researchers recording the dataset are also presented in the table. Other failure causes include wear and tear, lubrication and minor defect which are all mild conditions and often early warnings of the failure causes and are presented in the table. The set covers periods longer than the manufacturer's guideline for bearing life-time. Both x and y-axis vibration is recorded in the 1st experiment of the dataset (8 readings from Table 2) but only x-axis vibration is recorded in experiments 2 and 3.

The assumptions at the beginning of every experiment are: (i) the bearings were in perfect condition at the beginning of every experiment (dataset description) (ii) reliability of 100%. The

Table 1: IMS dataset experiments and recorded failures used for validation of the DSS output.

Experiment	Duration	Failure
1st experiment	34.5 days	inner race defect in bearing 3 roller element defect in bearing 4
2nd experiment	7 days	outer race failure in bearing 1
3rd experiment	44 days	outer race failure in bearing 3

Table 2: Measured parameters on bearing vibration.

No.	Measurement	No.	Measurement
1	Bearing 1 x-axis	5	Bearing 1 y-axis
2	Bearing 2 x-axis	6	Bearing 2 y-axis
3	Bearing 3 x-axis	7	Bearing 3 y-axis
4	Bearing 4 x-axis	8	Bearing 4 y-axis

original data is recorded in csv format and for the total of the 3 experiments span 86 days (approximately 2.8 months). The collective dataset is 6.1 GB. Transmission of the raw data would require approximately 76 MB of data to be wirelessly transmitted from one Edge computing unit per day.

3.4 Implementation

The FFTW open source C library generated the frequency spectrum data [16]. An open source tool was used to generate reliability values from these properties [48]. The reliability values were inputted to the reliability analysis tool [41] which returns a set of five predicted reliability values over each current calculated reliability value. Then the DSS tool received those as input[40]. The evaluation focuses on DSS results.

3.5 Experimental Setup

To conduct the experiment we required an environment similar to the engine room but also a significant data load from several sensors. The virtual sensors allow controlled inputs and low cost high data volume generation from an established vibration dataset. The experiments have been set up in the Marine Engineering Laboratory within the Naval Architecture, Ocean & Marine Engineering Department at Strathclyde University. The laboratory hardware configuration and experimental setup has been previously presented and validated [40]. Even though it is smaller than a ship's engine room, the laboratory provides conditions of high interference and electromagnetic noise being a $3x3$ meter metal box with a glass window for visual access to the "CM12 Armfield" automotive four-cylinder, direct injection, high speed diesel engine. The engine was periodically switched on and off to generate interference and to record physical sensor pressure data. This environment realistically represents the engine room as it induces similar noise interference to the wireless communication signals. The duration was approximately one day (8 hours). The laboratory provides the required environment and access to only one physical sensor for measurements from the diesel engine. Thus, we combine the virtual sensors and the single physical sensor on the same hardware to create a single experiment in the correct environment but with the required data volume generated (Fig. 3(right)).

The IoT Edge compute capable system connected to a set of virtual sensors is presented in Fig. 3(left). The wireless transmitter connection is via the USB cable. Two units were deployed with one physical sensor and a maximum of 15 virtual sensors for each experiment in the IMS dataset. The sequence diagram in Fig. 3 demonstrates the process followed to generate analogue input signals from the IMS dataset files. In essence, each data point is read from the file, sent over an analogue output to the analogue input of an ADC chip. The signal is thus equivalent to the physical connection from any vibration sensor from this point onward. Then, the ADC translates the analogue signal back to digital and passes the digital signal to one of the Beagle's GPIO input pins for pre processing. The 1st experiment was Edge-processed and transmitted from one Beagle unit while the remaining transmitted sequentially from the second unit.

The facility allowed us to simulate an environment very close to the real vessel engine room. Nonetheless, the smart sensors' deployment on a real ship would need to address the following challenges. First, the sensors' network would need to operate in a potentially harsh environment, as the engine room is an area where oil, fuel and dirt are present. For this reason, a vessel's engine room has special demands for the installed equipment, which are usually controlled by the classification societies. Secondly, the deployment would need to account for the engine room's high overall ambient temperature, with extreme hot spots close to the exhaust pipes. Several areas will also be subjected to high vibrations, especially at the rear of the vessel, for example, in the steering gear room or above the propeller shaft tunnel. Moreover, due to limited space and restricted heights in the engine room, the sensors' network would also require deployment solutions whose designs must be carefully considered. In this respect, important factors for engine rooms applications should be considered: i) No maintenance and reduced power consumption (high energy efficiency); ii) Mechanical strength/impact resistance; iii) Shock and vibration resistance. The applicability of wireless communications has been previously evaluated in literature [26].

3.6 Comparative Analysis Methodology

To demonstrate the novelty of the proposed work a comparison to related work was also performed. The Github repositories identified trhough the systematic search were cloned and no modification were made to the code. At points the Makefile had to be updated to enable compilation. Dependencies were installed for all repositories as specified in the Readme.md files or as identified during compilation. In the case of Python, the relevant environment was sourced as indicated by the repository authors. In the case of C or C++ the code was compiled in the version of clang or gcc that the repository specified.

To provide a common target platform all the repositories as well as the proposed methodology were executed on an 3 GHz 6-Core Intel Core i5 and a Radeon Pro 560X 4 GB graphics card was also available. This was selected as some of the methods would not be executable on the Beagle Bone Black platform without excessive time requirements. During the experiment all other processes were sleeping or were inactive to enable a common baseline. For each codebase the full IMS dataset was processed and the output was used by the proposed post-processing and DSS components.

Additionally, to measure CPU utilisation and completion time one 10-minute window of the IMS dataset was processed. The same 10-minute window was provided as input. The CPU time and the clock time were recorded using the `time` Unix command line tool. This returns clock, CPU kernel-space, and user-space time in seconds. The first is real time of execution, the second and third are cumulative for all CPU cores. As a result the sum of kernel-space and user-space time can be larger than the clock time, if more than one CPUs were used. The sum is reported in this paper to evaluate real-time pre-processing capacity of each method.

Finally, for lossy algorithms the Root Mean Square Error was recorded. To record this the data was post-processed and compared to the raw data. For all methods the compression ratio was also recorded based on the data that would be transmitted for the full IMS dataset.

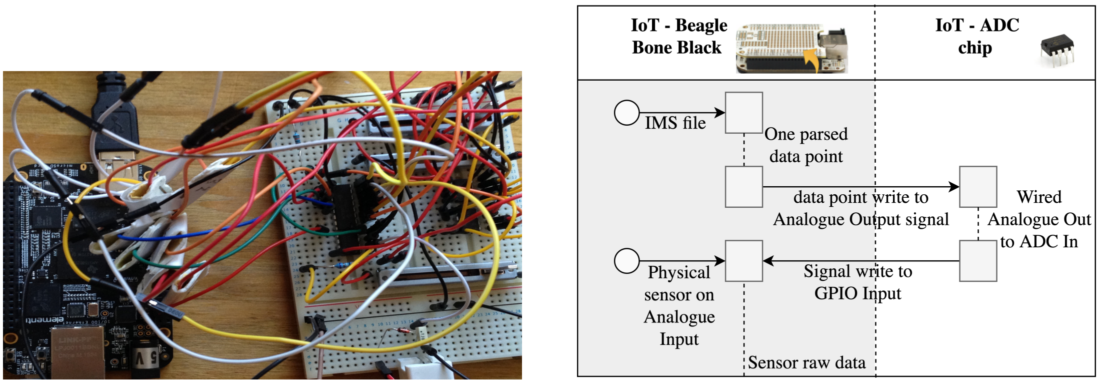

IoT - Beagle Bone Black
IoT - ADC chip
IMS file
One parsed data point
data point write to Analogue Output signal
Wired Analogue Out to ADC In
Physical sensor on Analogue Input
Signal write to GPIO Input
Sensor raw data

Figure 3: All 7 virtual sensors connected from the Analogue I/O (ports 33 to 40) and connection to BNC for physical sensor input (left) and sequence diagram of raw data acquisition process for virtual (IMS data) and physical sensors (right).

4 Results

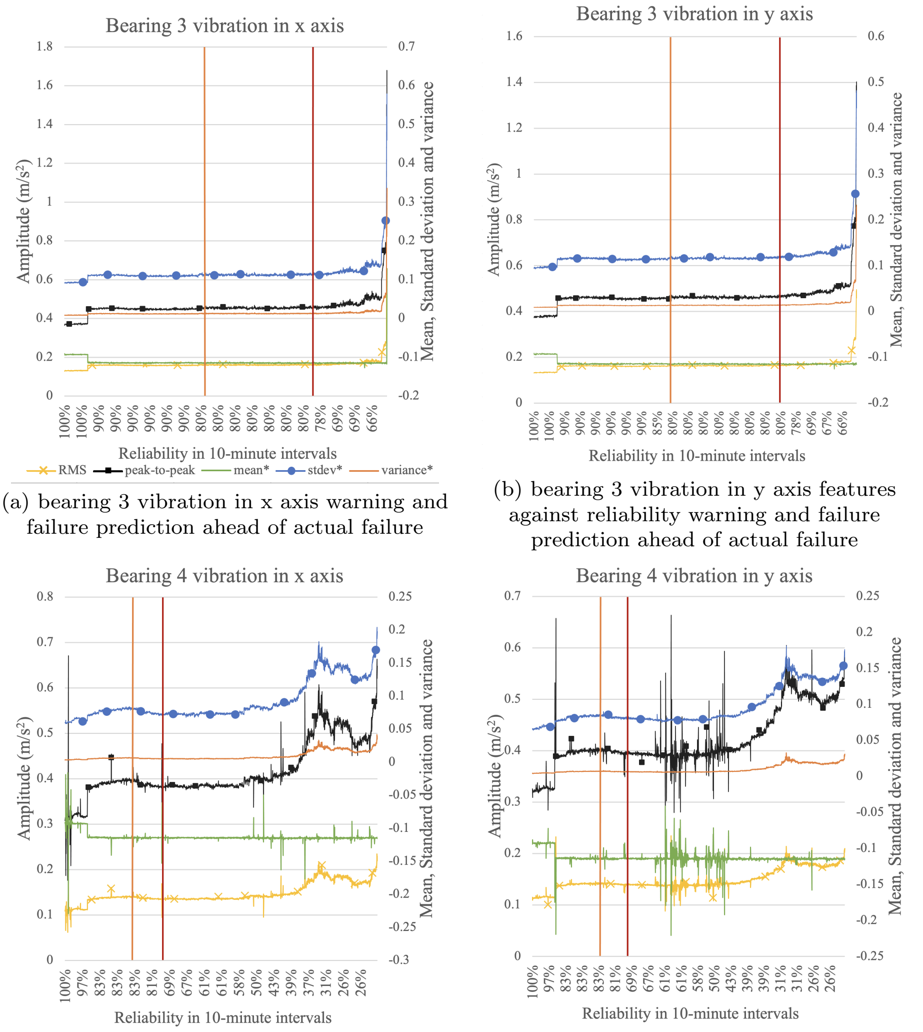
A total of 6.347 MB was transmitted over the wireless network (0.097% of the volume of raw data). Each bearing is considered as an independent system with independent data. Thus, the DSS predicts individual component warnings and/or failures. The scatter plots of the following figures map the pre-processing extracted features to the corresponding calculated reliability. Each point in the graph corresponds to the reliability calculated on a 10 minute measurement window. The predicted values are not represented in this graph. The horizontal axis is the change of calculated reliability in time.

In the 1st experiment (Fig. 4) bearing 1 reliability moves from 100% to 91%, bearing 2 to 89%, considered normal operation and thus the graphs are not presented. Bearings 3 and 4 displayed a different behaviour. There was no reported failure or warning by the DSS for bearings 1 or 2 as expected in Table 1. Bearing 3 demonstrates faster degradation (reliability of 40%) with a predicted warning (orange) and failure (red). The warning was detected 19 days in advance, failure predicted 8.6 days earlier, in accordance with Table 1. Material wear and tear was identified as root cause instead of the actual cause which was defect according to the Table. The cost associated with the warning stage was $2,000 for the asset, the cost of maintenance was $0 assuming onboard parts and labour, and the cost of delay was $5,000 for one day assuming the main engine of the ship would need to stop for the maintenance action. For all three bearings the y-axis vibration did not provide additional information regarding the condition as the DSS results were identical for each bearing. The DSS suggested actions in order of likelihood were:

1. ”Check Bearing 3 for material wear and tear”.
2. ”Check Bearing 3 for non-effective lubrication”.
3. ”Check Bearing 3 for overheating”.

Bearing 4 also reported unusual values early on, also reporting differences in y and x axis vibration (RMS, μ and $peak-to-peak$). The warning was raised 29 days ahead of experiment end and the failure 25 days, correctly identifying a defect as root cause (Table 1). The costs calculated by the DSS were identical to bearing 3. The actions suggested by the DSS were:

1. ”Check Bearing 4 for manufacturer defect”.



(a) bearing 3 vibration in x axis warning and failure prediction ahead of actual failure

(b) bearing 3 vibration in y axis features against reliability warning and failure prediction ahead of actual failure

(c) bearing 4 vibration in x axis warning and failure prediction well ahead of actual failure

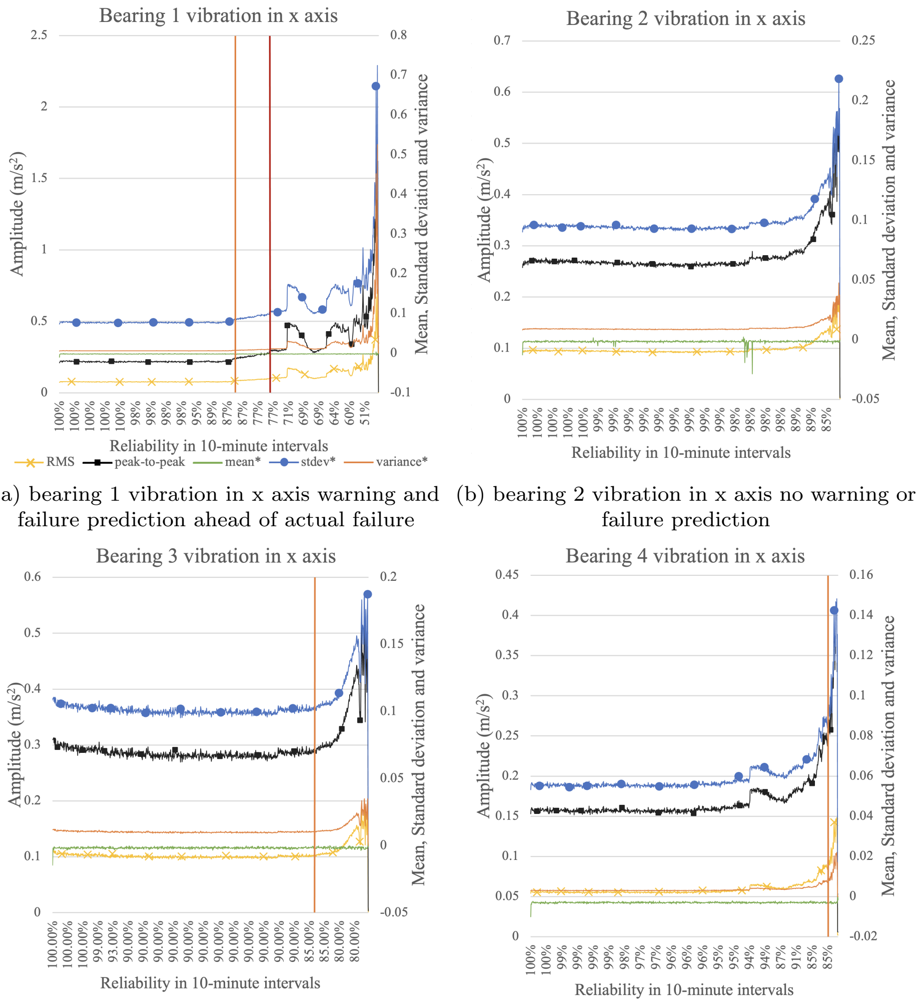
(d) bearing 4 vibration in y axis warning and failure prediction well ahead of actual failure

Figure 4: Statistical properties results of pre-processing of bearings 3 (a, b) and 4 (c, d) of experiment 1 mapped to the reliability results of post-processing. Legend is presented in (a) and shared by all sub-figures. Series marked with * in the legend are plotted on the secondary vertical axis (right side) while the remaining are plotted against the primary vertical axis (left side). DSS warnings as vertical orange lines and failures as red.

2. "Check Bearing 4 for material wear and tear".

3. "Check Bearing 4 for non-effective lubrication".

Results for experiment 2 data are presented in Fig. 5. Warnings were predicted for Bearing 1, 3 and 4 of experiment 2; for bearings 3 and 4 predicting a potential failure after the end of the experiment. It is thus impossible to validate those two potential failures due to lack of further data.



(a) bearing 1 vibration in x axis warning and failure prediction ahead of actual failure

(b) bearing 2 vibration in x axis no warning or failure prediction

(c) bearing 3 vibration in x axis warning raised ahead the end of experiment

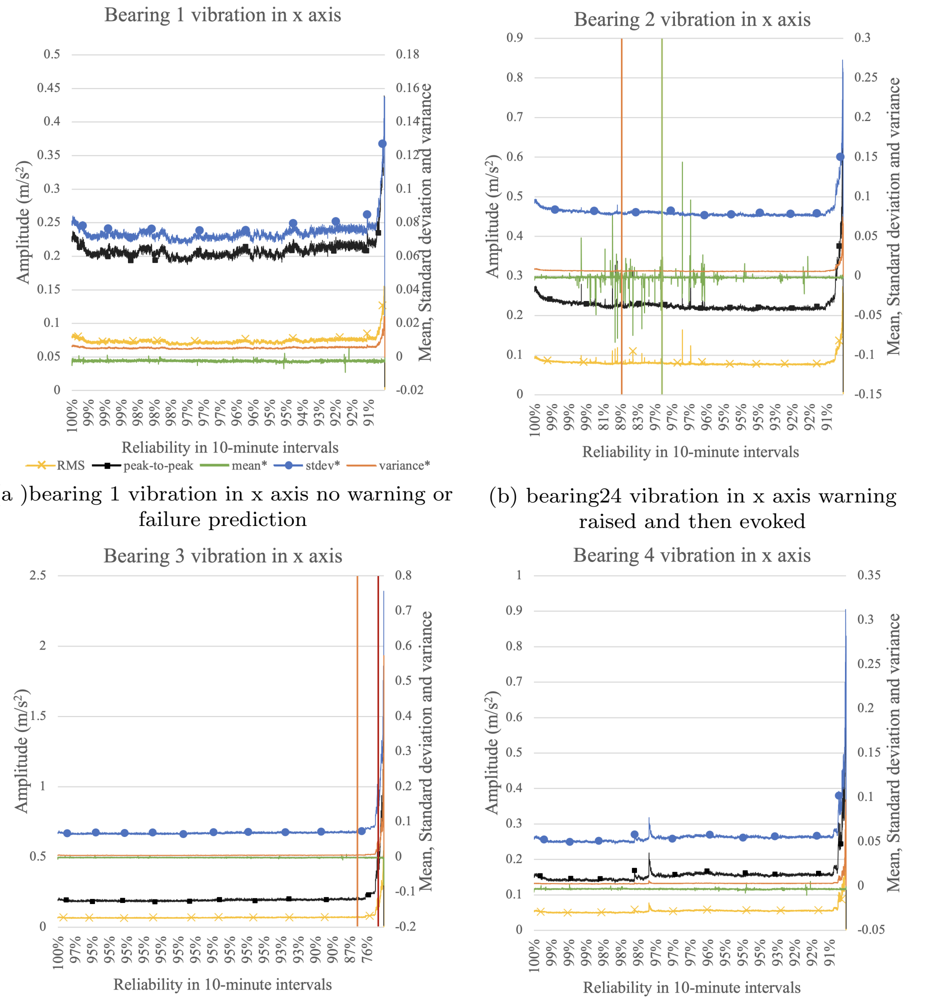
(d) bearing 4 vibration in x axis warning raised just before the end of the experiment

Figure 5: Statistical properties results of pre-processing in experiment 2 mapped to the reliability results for all bearings. Plot as in Fig. 4. Legend presented in (a).

The suggested actions and predicted cost are identical to experiment 1 bearing 3. The only predicted failure (bearing 1) matched the condition at the end of the IMS dataset (Table 1).

Experiment 3 results are presented in Fig. 6. A warning was raised for bearing 2 but later evoked as reliability increases. An additional warning was raised for bearing 3 leading to a predicted failure (9 and 2 days before experiment end respectively). Thus meeting the expectation (Table 1) but with short time-to-failure. Overall all ISM dataset failures were identified successfully and no false positives/negatives were produced.

These results demonstrate that the lossy extremely compressed transmitted volume does not sacrifice specificity or sensitivity in regards to the decision making process. To further investigate the novelty of the proposed method a comparative analysis was performed against state-of-the-art



(a)bearing 1 vibration in x axis no warning or failure prediction

(b) bearing24 vibration in x axis warning raised and then evoked

(c) bearing 3 vibration in x axis warning and failure prediction just ahead of actual failure

(d) bearing 4 vibration in x axis no warning or failure prediction

Figure 6: Statistical properties results of pre-processing in experiment 3 mapped to the reliability results for all bearings. Plot as in Fig. 4. Green vertical line denotes warning state evoked. Legend presented in (a).

methods identified in Section 2.

The results are summarised in Table 3. The transmitted volume in of our proposed approach is several orders of magnitude smaller than any other Edge computing or compression approach. Moreover, our proposed method has the smallest Root Mean Square Error (RMSE). The RMSE of 0.0003 justifies our earlier findings in terms of correct identification of warnings and failures and outperforms the LFZip approach even in the case of 10% error.

Further the time and CPU time of each approach are presented in Table 3 for the processing of a 10 minute window of raw data. This was measured to evaluate the suitability of each method for IoT deployment (Edge device target). As expected implementations in C have smaller CPU

Table 3: Comparative analysis. Transmitted volume (Trans.) reports only the output of each method for the IMS dataset. In the case of the proposed method, the physical sensor information was removed to allow an even baseline of comparison. The programming language (Lang.) is also reported as well as the CPU time (t.).

Git Repo	Category	Lang.	Trans.	CR	Time (s)	CPU t. (s)	RMSE
ride [3]	Edge Server	-	6.1G	0	-	-	-
EdgeComp	Edge Server	-	6.1G	0	-	-	-
sldc [11]	Lossless	C	628M	9.94	0.07	0.06	0
Huffman [23]	Lossless	C	2.6G	2.36	0.173	0.146	0
FastLZ [24]	Lossless	C	2.6G	2.38	0.038	0.014	0
bzip2 [51]	Lossless	C++	2G	3.06	2.907	3.058	0
cuSZ [58]	Lossy	CUDA	-	-	-	-	-
LFZip error 0.01 [5]	Lossy	Python	484M	12.93	0.222	0.630	0.005
LFZip error 0.1 [5]	Lossy	Python	125M	50.09	0.271	0.705	0.058
Proposed	Edge lossy	C	**4M**	**1553.39**	**0.014**	**0.013**	**0.0003**

time requirements than C++ or Python. Total was below 1 second for the manority of cases. However to meet real-time requirements the results of sldc, FastLZ and our proposed method are more appropriate. The smallest times are achieved by our proposed method closely followed by the FastLZ which implements the LZ77 algorithm. Additionally, C++ and Python implementations demand more than one CPU for their execution, demonstrated through the $CPUtime > Time$ relationship. As the majority of IoT processors utilise a single CPU these methods are not portable. In the future more capabilities are expected on IoT devices including multiple CPUs (e.g. Big-Little arm architectures) and GPUs (e.g. Raspberry Pi 3). However, this is not yet the case. Thus, approaches such as cuSZ using CUDA are not yet suitable in a vast majority of IoT devices. Unfortunately, in the case of cuSZ the process crashed when attempting to compress the IMS dataset and no results were obtained. Finally, in the case of Edge computing approaches, the full raw data is transmitted to the Edge server so measurements of time and RMSE were not relevant and are not reported. Also the Edge components execute on Software Defined Network infrastructure and the language is not reported as multiple languages are used.

Finally, the LFZip compression results when the error is set to 10% were post processed and results were obtained through the DSS. This replicated the same steps used in our proposed method with the input of the post processing algorithm being replaced with the 125M of transmitted data. In summary, the DSS did not identify 2 of the expected warnings at the correct time (experiment 2 bearings 3 and 4) and did not identify 3 of the expected failures (experiment 1 bearing 3, experiment 2 bearing 1 and experiment 3 bearing 3). Thus, a $CR = 50.09 : 1$ is detrimental to the decision making process when events are not identified.

5 Discussion

5.1 Evaluation

In this paper we verify the capability of the system to process different inputs simultaneously. Additionally, we verify and validate the data management and processing method through comparing the DSS output with reported failures of the IMS dataset. In all experiments the failures were predicted successfully and no false predictions were recorded. The DSS only failed to identify the correct root cause in one occasion. In most cases (1^{st} & 2^{nd}) a long time-to-failure was recorded; 1/2 of bearing's life-time or earlier. Though in some cases the warning to-failure period was very short (worst case 9 days). Some warnings that did not match reported failures were raised. These could not be validated due to available data limitations. However, all bearings were operating in excess of their useful life-time according to the manufacturer, making a failure a high possibility.

A total cost of $17,000 for each bearing was predicted for the shipping company; on effect of failure to the system, its maintenance, and vessel's operation. Suggested actions were validated against those identified by the experts for bearing maintenance.

Through all experiments, 0.097% of the raw data volume was transmitted over the network. A reduction which exceeds the best performing lossy compression for CM data with $CR = 92 : 1$ [55] and the best performing LFZip lossy approach with $CR = 50.09 : 1$ by more than 1 order of magnitude, equating to $CR = 1553 : 1$. For vibration data, we report a volume reduction of 3 orders of magnitude without compromising the DSS output and decision making process, thus without sacrificing specificity or sensitivity as evident through the $RMSE = 0.0003$. This is higher than any reported or evaluated compression algorithm for this type of data making the proposed approach a significant contribution to IoT and Edge computing systems for CM.

5.2 Impact

The proposed approach advances ship CM and advances the technical results reported in literature in the performance and suitability of wireless networks in shipping CM. The system addresses the identified gap in literature providing wireless data transfer of extracted features with very low network overheads.

The impact to stakeholders and benefits are well documented; reduction of failures, business target management and compliance, reduction of human error, maintenance of higher quality, and reduction of maintenance cost, leading to less maintenance at sea, no need for permanent crew that has experience on the particular vessel, better performance and condition of ship, reduction of risk, better emission management, and increased availability of the ship [35, 60].

5.3 Security, Transferability, Limitations & Threats to Validity

In the current work we make no guarantees for the *security* of the data. In this direction, well known approaches are available [27], while low-cost IoT generation approaches are investigated [64]. However, as raw data is never transmitted over the network privacy increases and inherently security. *Transferability* to any other domain where CM is applicable is possible if alternative post-transmission processing and DSS components are utilised (e.g. robotics, power generation systems, wind turbines, remote locations). The *limitations* of the proposed system are mostly defined by hardware considerations; maximum number of sensors, CPU or Bandwidth, battery power. Also, post-processing some of the vibration datasets could limit the capacity of the system to interface to any analysis tool (e.g. high fidelity requirements of ANNs). *Threats to the validity* are lead by assumptions made. One was that the raw data stream was accurate. However, this is out of the scope of this paper and the DSS output was not affected. Another threat to validity was the laboratory environment.

6 Conclusion

The system decentralises data analytics, effectively Edge-processes and wirelessly transmits equipment and machinery CM data on IoT, providing low cost CM for the maritime industry. The system can parallel process several input sensors. At minimum we extract (i) a subset of the frequency spectrum, and (ii) the statistical properties of the recorded signal with a single computation over the input measurement. We thus successfully carry the required information without sacrificing specificity or sensitivity and without compromising the decision-making process. We reduce the traffic of data from the data acquisition point to the final presentation point. Along with the DSS's reduced training overheads and improved benefit identification, it supports enhanced utility and applicability providing several benefits to industry stakeholders. In conclusion, the system addresses the current industry needs and contributes to existing research in processing data away from central servers within the ship CM domain and a constrained context while achieving 10^3 volume reduction and $RMSE = 0.0003$.

7 Further work

Further case studies could investigate the optimal percentage of frequency spectrum peaks that is sent. This would potentially increase the amount of information that is successfully transmitted and enhance compatibility with post-transmission processing tools. Additionally, other measurements could be added to the system for completeness (e.g. speed). Finally, the experimental setup could be improved. However, laboratory facilities were closed due to COVID19 and visits on-board vessels impossible at this period. Thus such work will need to be undertaken in the future when normal operations resume.

8 Acknowledgements

To Jeremy Singer, for the feedback and internal review of the paper.

References

[1] K. D. Adamson. *Maritime Satellite Communications & Applications 2016 Research White Pape.* Mayfair, London, 2016.

[2] Yuan Ai, Mugen Peng, and Kecheng Zhang. "Edge computing technologies for Internet of Things: a primer". In: *Digital Communications and Networks* 4.2 (2018), pp. 77–86.

[3] Kyle E Benson et al. "Ride: A resilient iot data exchange middleware leveraging sdn and edge cloud resources". In: *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI).* IEEE. 2018, pp. 72–83.

[4] Srdja Bjeladinovic. "A fresh approach for hybrid SQL/NoSQL database design based on data structuredness". In: *Enterprise Information Systems* 12.8-9 (2018), pp. 1202–1220. DOI: 10.1080/17517575.2018.1446102. eprint: https://doi.org/10.1080/17517575.2018.1446102. URL: https://doi.org/10.1080/17517575.2018.1446102.

[5] Shubham Chandak et al. "LFZip: Lossy compression of multivariate floating-point time series data via improved prediction". In: *2020 Data Compression Conference (DCC).* IEEE. 2020, pp. 342–351.

[6] J Chen et al. "Neural network based diagnosis of mechanical faults in IC engines". In: *10th International Conference on Vibrations in Rotating Machinery.* Woodhead Publishing, 2012, pp. 679–690. DOI: 10.1533/9780857094537.10.679.

[7] Diego Coronado and Katharina Fischer. *Condition monitoring of wind turbines: State of the art, user experience and recommendations.* Tech. rep. VGB-Nr.383, -Nr.106070. Fraunhofer Institute for Wind Energy and Energy System Technology IWES Northwest, 2015.

[8] DNV GL. “Beyond condition monitoring in the maritime industry”. In: *Det Norske Veritas and Germanischer Lloyd (DNV GL) Strategic Research & Innovation* (2014).

[9] DoD. *DOD guide for achieving reliability, availability, and maintainability.* Tech. rep. USA Department of Defence, 2005.

[10] EC. *Study on the Analysis and Evolution of International and EU Shipping.* Tech. rep. European Commission, University of Antwerp, maritime-insight, Panteia, Significance, PWC, 2015.

[11] ECMA. *ECMA 321 Streaming Lossless Data Compression algorithm (SLDC).* June 2001. URL: `https://www.ecma-international.org/publications-and-standards/standards/ecma-321/`.

[12] ECSA. *Annual report 2012-2013.* Tech. rep. European community ship owners’ association, Brussels, Belgium, 2013.

[13] EP and EC. “Regulation (EU) 2015/757, of the European Parliament and of the Council on the monitoring, reporting and verification of carbon dioxide emissions from maritime transport, and amending Directive 2009/16/EC, the European Parliament and the Council of the European Union”. In: *Official Journal of the European Union.* L 123 (Apr. 2015), pp. 55–76.

[14] Guo-Jin Feng et al. “Implementation of envelope analysis on a wireless condition monitoring system for bearing fault diagnosis”. In: *International Journal of Automation and Computing* 12.1 (2015), pp. 14–24.

[15] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. “Mobile cloud computing: A survey”. In: *Future Generation Computer Systems* 29.1 (2013), pp. 84–106.

[16] Matteo Frigo and Steven G Johnson. “The design and implementation of FFTW3”. In: *Proceedings of the IEEE Special issue on Program Generation, Optimization, and Platform Adaptation* 93.2 (2005), pp. 216–231.

[17] Xiuju Fu et al. “AIS Data Analytics for Intelligent Maritime Surveillance Systems”. In: *Maritime Informatics.* Springer, 2021, pp. 393–411.

[18] Natascha Harth, Christos Anagnostopoulos, and Dimitrios Pezaros. “Predictive intelligence to the edge: impact on edge analytics”. In: *Evolving Systems* (Aug. 2017). ISSN: 1868-6486.

[19] Björn Hassler. “Oil Spills from Shipping: A Case Study of the Governance of Accidental Hazards and Intentional Pollution in the Baltic Sea”. In: *Environmental Governance of the Baltic Sea.* Ed. by Michael Gilek et al. Cham: Springer International Publishing, 2016, pp. 125–146. ISBN: 978-3-319-27006-7. DOI: `10.1007/978-3-319-27006-7_6`.

[20] Jie-Fu Huang, Geng-Hua Zhang, and Sun-Yuan Hsieh. “Real-time energy data compression strategy for reducing data traffic based on smart grid AMI networks”. In: *The Journal of Supercomputing* (2021), pp. 1–20.

[21] Ivor Humphreys, Gerrit Eisenblätter, and Garret E. O’Donnell. “FPGA based monitoring platform for condition monitoring in cylindrical grinding”. In: *Procedia CIRP* 14 (2014), pp. 448–453.

[22] IMS. *NASA Ames Prognostics Data Repository: The Center for Intelligent Maintenance Systems (IMS), University of Cincinnati Bearing Data Set.* URL: `http://ti.arc.nasa.gov/project/prognostic-data-repository`.

[23] Mazharul Islam et al. “HEliOS: huffman coding based lightweight encryption scheme for data transmission”. In: *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems Computing, Networking and Services.* 2019, pp. 70–79.

[24] Jansson Jesper, Kunihiko Sadakane, and Wing-Kin Sung. *Fast LZ-compression algorithm.* Tech. rep. Technical report, Department of Computer Science and Communication Engineering, Kyushu University, Japan.

[25] K. Hentschel et al. “Supersensors: Raspberry Pi Devices for Smart Campus Infrastructure”. In: *IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud).* Aug. 2016, pp. 58–62.

[26] Hussein Kdouh et al. “Wireless sensor network on board vessels”. In: *2012 19th international conference on telecommunications (ICT).* IEEE. 2012, pp. 1–6.

[27] R. Klempous et al. “Byzantine Algorithms in Wireless Sensors Network”. In: *2006 International Conference on Information and Automation.* Dec. 2006, pp. 319–324.

[28] Yousun Ko et al. “Lane Compression: A Lightweight Lossless Compression Method for Machine Learning on Embedded Systems”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 20.2 (2021), pp. 1–26.

[29] Manos Koutsoubelias et al. “System Support for the In Situ Testing of Wireless Sensor Networks via Programmable Virtual Onboard Sensors.” In: *IEEE Trans. Instrumentation and Measurement* 65.4 (2016), pp. 744–753.

[30] Chiman Kwan and Yvonne Luk. “Hybrid sensor network data compression with error resiliency”. In: *Data Compression Conference, Snowbird, Utah.* 2018.

[31] Malcolm Latarche. *The Connected Ship – DNV GL.* Ed. by ShipInsight. Accessed: 2016-07-25. July 2016. URL: `https://www.shipinsight.com/the-connected-ship-dnv-gl/`.

[32] Arthur W Lees. “Recent advances and prospects in condition monitoring”. In: *Vibration Engineering and Technology of Machinery.* Springer, 2015, pp. 51–63.

[33] Zhixiong Li et al. “Intelligent fault diagnosis method for marine diesel engines using instantaneous angular speed”. In: *Journal of Mechanical Science and Technology* 26.8 (2012), pp. 2413–2423.

[34] M Lind et al. “Digitalizing the port call process”. In: *UNCTAD Transport and Trade Facilitation Series* 13 (2019).

[35] Lloyds’ Register. *Cyber-enabled ships. Deploying information and communications technology in shipping - Lloyd’s Register’s approach to assurance, Fifth edition.* Lloyds’ Register, 2016. URL: `https://issuu.com/lr_marine/docs/lr_guidance_note_cyber-enabled_ship`.

[36] Tshilidzi Marwala. *Condition Monitoring Using Computational Intelligence Methods, Applications in Mechanical and Electrical Systems.* Springer London, 2012. ISBN: 978-1-4471-2380-4. DOI: `10.1007/978-1-4471-2380-4`.

[37] Jay McFadyen and K. D. Adamson. “Buying the Cow”. In: *Furturenautics* 10 (2016), pp. 8–13.

[38] Hong Mei and Yanjie Yin. “Studies on marine oil spills and their ecological damage”. In: *Journal of Ocean University of China* 8.3 (2009), pp. 312–316.

[39] MEPC. *The Marine Environment Protection Committee, RESOLUTION MEPC.203(62), 62/24/Add.1, Amendments to the Annex of the protocol of 1997 to amend the international convention for the prevention of pollution from ships, 1973, as modified by the protocol of 1978 relating thereto (Inclusion of regulations on energy efficiency for ships in MARPOL Annex VI).* IMO, 2016.

[40] A. L. Michala and I. Vourganas. “A smart modular wireless system for condition monitoring data acquisition.” In: *COMPIT'17*. Volker Bertram, Hamburg, May 2017, pp. 212–225.

[41] A. L. Michala et al. “Wireless condition monitoring for ship applications”. In: *Smart Ship Technology*. The Royal Institution of Naval Architects, Jan. 2016, pp. 59–66.

[42] AL Michala and I Lazakis. “Ship machinery and equipment wireless condition monitoring system”. In: (2016).

[43] R. Keithy Mobley. “Regulatory Compliance Issues in the US”. In: *Plant Engineer's Handbook*. Ed. by R. Keith Mobley. Butterworth-Heinemann, 2001. Chap. 64, pp. 1151–1158.

[44] C Oswald and B Sivaselvan. “An optimal text compression algorithm based on frequent pattern mining”. In: *Journal of Ambient Intelligence and Humanized Computing* 9.3 (2018), pp. 803–822.

[45] Ralph W. Peters. “Defining Maintenance Strategies for Critical Equipment With Reliability-Centered Maintenance (RCM)”. In: *Reliable Maintenance Planning, Estimating, and Scheduling*. Ed. by Ralph W. Peters. Gulf Professional Publishing, 2015. Chap. 9, pp. 145–155.

[46] George Pitblado. “Maintenance Management in UK”. In: *Plant Engineer's Handbook*. Ed. by R. Keith Mobley. Butterworth-Heinemann, 2001. Chap. 48, pp. 845–856.

[47] Zoya Pourmirza, Sara Walker, and John Brooke. “Data reduction algorithm for correlated data in the smart grid”. In: *IET Smart Grid* (2021).

[48] A Quezada. *Open Source FFT Spectrum Analyzer*. Accessed: 2017-03-03. 2017. URL: `https://hackaday.io/project/`.

[49] Gerald Rofle. *Condition Based Maintenance to optimize Asset efficiency*. 2015. URL: `http://www.green4sea.com/condition-based-maintenance-to-optimize-asset-efficiency/`.

[50] Colin Samplawski et al. “Towards Objection Detection Under IoT Resource Constraints: Combining Partitioning, Slicing and Compression”. In: *Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*. 2020, pp. 14–20.

[51] Robert Sedgewick and Kevin Wayne. *Algorithms, fourth edition*. Addison-Wesley, 2011.

[52] Mahdi Shadab Far and Yuan Wang. “Approximation of the Monte Carlo Sampling Method for Reliability Analysis of Structures”. In: *Mathematical Problems in Engineering* 2016 (2016), p. 9.

[53] Daniel C. Shorten. *Marine Machinery Condition Monitoring. Why has the shipping industry been slow to adopt?* Tech. rep. Lloyd's Register, EMEA, 2012.

[54] SKF. *SKF Integrated Condition Monitoring Industry-leading tools and technologies for optimized machine maintenance and reliability*. 2016. URL: `http://www.skf.com/binary/28-75560/Interactive-brochure-SKF-Integrated-Condition-Monitoring.pdf`.

[55] Julio Cesar Stacchini de Souza, Tatiana Mariano Lessa Assis, and Bikash Chandra Pal. “Data compression in smart distribution systems via singular value decomposition”. In: *IEEE Transactions on Smart Grid* 8.1 (2017), pp. 275–284.

[56] Philip Sparks. “The route to a trillion devices”. In: *White Paper, ARM* (2017).

[57] M. Stopford. *Maritime economics 3rd edition*. London: Routledge, 2009.

[58] Jiannan Tian et al. “Cusz: An efficient gpu-based error-bounded lossy compression framework for scientific data”. In: *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*. 2020, pp. 3–15.

[59] Masaru Tsujimoto and Hideo Orihara. “Performance prediction of full-scale ship and analysis by means of on-board monitoring (Part 1 ship performance prediction in actual seas)”. In: *Journal of Marine Science and Technology* (2018), pp. 1–18.

[60] Sanjay Tyag and Sameer Sharma. *Intelligent Maritime Modern Ships and Ports*. Tech. rep. DELL Technologies and intel Corporation.

[61] B. Varghese et al. “Edge-as-a-Service: Towards Distributed Cloud Architectures”. In: *ArXiv e-prints* (Oct. 2017). presented at the EdgeComp Symposium 2017; will appear in Proceedings of the International Conference on Parallel Computing.

[62] Xin-yan Xiong et al. “Vibration monitoring system of ships using wireless sensor networks”. In: *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 90–94.

[63] Ibna Zaman et al. “Challenges and Opportunities of Big Data Analytics for Upcoming Regulations and Future Transformation of the Shipping Industry”. In: *Procedia engineering* 194 (2017), pp. 537–544.

[64] Y. Zhang et al. “Low-Cost and Confidentiality-Preserving Data Acquisition for Internet of Multimedia Things”. In: *IEEE Internet of Things Journal* 5.5 (Oct. 2018), pp. 3442–3451. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2781737.