INTELLIGENT MONITORING OF A LARGE CATAMARAN FERRY

Reference NO. IJME 791, DOI: 10.5750/ijme.v165iA1.791

B Shabani, **J Ali-Lavroff**, **D S Holloway**, University of Tasmania, Australia, **S Penev**, UNSW Sydney, Australia, **D Dessi**, Italian National Research Council, Italy, **G Thomas**, University College London, UK

KEY DATES: Submission date: 22.10.21 / Final acceptance date: 04.05.23

SUMMARY

Wave load cycles, wet-deck slamming events, accelerations and motion comfort are important considerations for high-speed catamarans operating in moderate to large waves. Although developing a hull monitoring system according to classification guidelines for such vessels is broadly acceptable, the data processing requirements for outputs such as rainflow counting, filtering, probability distribution, fatigue damage estimation and warning due to slamming can be as sophisticated to implement as the system components themselves. Advanced analytics such as machine learning and deep learning data pipelines will also create more complexities for such systems, if included. This paper provides an overview of data analytics methods and cloud computing resources employed for remotely monitoring motions and structural responses of a 111 m high-speed catamaran. To satisfy the data processing requirements, MATLAB Reference Architectures on Amazon Web Services (AWS) were used. Such combination enabled fast parallel computing and advanced feature engineering in a time-efficient manner. A MATLAB Production Server on AWS has been set up for near real-time analytics and execution of functions developed according to the class guidelines. A case study using Long ShortTerm Memory (LSTM) networks for ship speed and Motion Sickness Incidence (MSI) is provided and discussed. Such data architecture provides a flexible and scalable solution, leading to deeper insights through big data processing and machine learning, which supports hull monitoring functions as a service.

1. INTRODUCTION

This paper provides an overview of data analysis methods and cloud computing resources being considered for Volcan de Tagoro (see Figure 1), which has operated in Spain, Canary Islands in the route between Las Palmas and Tenerife since August 2019. Classified by DNV GL, the vessel has a maximum deadweight (DWT) of 1000 tonnes, can carry up to 1200 crew and passengers and 401 cars, and has a specification speed of 42.4 knots at 600 tonnes DWT. Large wave piercing catamarans (WPCs) have the capability of satisfying economic demands for fast sea transportation with improved seakeeping characteristics compared to conventional catamarans [1, 2]. Since the 1990s many large WPCs have been constructed at Incat's Hobart shipyard, ranging from 74 m to 112 m in overall length and being operated mainly as passenger/vehicle ferries. These aluminium vessels have a distinctive design due to an above-water centre bow connecting two demihulls at the forward region. The propulsion systems of the Incat WPCs are water jets and an active ride control system is installed to improve passenger comfort. **Error! Reference source not found.** shows the 111 m wave-piercing catamaran (Volcan de Tagoro, Hull 091) built by Incat Tasmania in 2019. Table 1 shows the main particulars of Hull 091. A hull monitoring system was developed for Hull 091 as previously described in [3] to continuously monitor the ship motions and structural responses.

It should be noted that wave load cycles, wet-deck slamming events, accelerations and motion comfort are important considerations for high-speed catamarans when operating in moderate to large waves. A guideline [4] presents the standard components to receive a Hull Monitoring System Notation (HMON) for different vessel types including high speed light craft. Although developing a hull monitoring system according to the classification guidelines for such vessels is broadly acceptable, the data processing requirements can be as sophisticated to implement as the system components themselves. One needs to consider rainflow counting, filtering, probability distribution, fatigue damage estimation, warning due to slamming and many others.

Machine Learning (ML) and Deep Learning (DL) algorithms [5-8] add a layer of intelligence to onboard monitoring systems, with their potential yet to be fully explored. However, ML and DL data pipelines will also create more complexities for such systems, as is also evident from the class recommended practices for data quality assessment (i.e. DNVGL-RP-0497) and data-driven algorithms and models (i.e. DNVGL-RP-0510).

One solution to overcome complexities in big data processing and machine learning of modern hull monitoring systems is cloud computing. This represents an alternative approach to onboard processing and storage. Both methods provide many advantages, with the latter being a common method within the industry. However, cloud computing resources are an excellent choice for big data processing [9] and real-time analytics and their applications across many industries are well understood. A cloud processing

Figure 1. Volcan de Tagoro, Hull 091 111m wave-piercing catamaran built by Incat Tasmania in 2019.

solution considering MATLAB Reference Architectures on Amazon Web Services (AWS) is introduced in this paper. Such a combination facilitated parallel processing of raw data and provided a framework for training various ML and DL algorithms on the cloud platform.

2. HULL MONITORING SYSTEM

A hull monitoring system has been developed for Hull 091 Incat vessel (see Figure 1). The vessel was instrumented in July 2019 using a motion reference unit (MRU), a bow accelerometer and 10 strain gauges. An ultrasonic wave sensor was placed in the bow area to measure the incident wave profiles. An overview of the remote monitoring system is shown in Table 2. In this paper, «remotely monitoring» refers to the use of cloud computing resources to monitor the motions and structural responses of a remotely located vessel. Figure 2 shows the approximate longitudinal locations of the cDAQ, laptop, MRU, strain gauges, bow accelerometer and the ultrasonic sensor. The sampling rates were set to 100 Hz for the bow and MRU acceleration data, 50 Hz for MRU heave, pitch, and roll data and 9 Hz for the ultrasonic sensor. The National Instruments (NI) module sampling rate was also 1000 Hz for strain measurements (due to a minimum requirement of NI-9236 module) but it was down sampled to 100 Hz before recording data to files.

Table 1: Main particulars of Hull 091

Variable	Value
Length overall	111.9 m
Length waterline	103.2 m
Beam Overall	30.5 m
Draft	4.1 m
Demihull beam	5.8 m
Max Deadweight	1000 tonnes
Speed	42.4 knots
Froude number	0.69

Table 2: Overview of Hull 091 remote monitoring components

Category	Item	Quantity	Description
Sensors /data source	Strain gauges	10	HBM 1-LY43-6/350
	Accelerometer (3-axial)	1	CrossBow- CXL04GP3
	Ultrasonic sensor	1	ToughSonic 50(TSPC-21SRM-485)
	Motion reference unit (MRU)	1	SBG Systems- Ellipse2-A
	Global Positioning System (GPS) receiver	1	Hull 091 GPS distributor
Data acquisition	Data acquisition (DAQ) module	1	National Instrument (NI cDAQ-9174)
	Strain gauges module (8 channel)	2	National Instrument (NI-9236)
	Universal Input module (4 channel)	1	National Instrument (NI-9219)
Computer, accessories and software	Laptop	1	Dell latitude 7490
	USB Hub	1	Powered USB hub
	Onboard Monitoring Software	1	Customised LabVIEW program
	Remote Desktop Access	1	TeamViewer
	Laptop tray	1	RAM Universal tray (RAM-234-3)
	Cabinet	1	PCLocs – Carrier 10
	NMEA to USB convertor	1	Digital Yacht
Storage & con-nectivity	External hard drive	1	Samsung- USB-C 1T SSD
	Cloud based storage	100GB (scalable)	Google Drive
	WiFi/LAN router	1	Digital Yacht 4G Connect PRO
	4G antennas	2	Digital Yacht 4G Connect PRO
	Sim Card	1	Simyo 4G

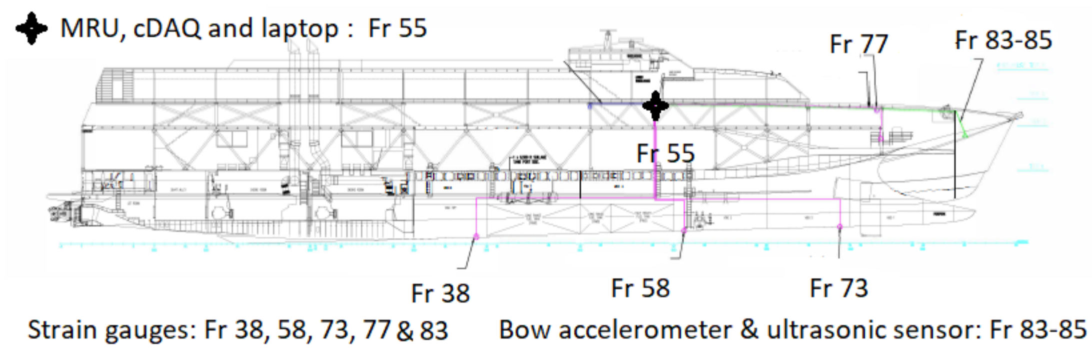


Figure 2. Hull 091 cabling diagram from frame 55 to frames (Fr) 38, 58, 73, 77, 83 and 85, showing the approximate longitudinal locations of the National Instruments cDAQ, laptop computer, MRU, strain gauges, bow accelerometer and ultrasonic sensor listed in Table 2

A LabVIEW program was developed to record and upload data automatically to a cloud storage. It is worth mentioning that developing a customised LabVIEW program enabled the integration of the MRU and GPS data. This was achieved by incorporating SBG Systems (Ellipse2-A) and NMEA-0183 GPS drivers, to the strain gauges, bow accelerometer and ultrasonic sensor data acquired from NI 9236 and 9219 modules. Furthermore, a file management function was included so that the files could be uploaded into the Google Drive and then deleted from the computer hard drive. The choice of Google Drive here is due to it having a general purpose, temporary storage for preliminary data access and some other user-friendly features such as automatic synchronisation and backup to manage large data files. The customised upload code developed in the LabVIEW program can be adjusted to upload the data into a cloud platform which enables advanced features such as machine learning, deep learning and near real-time monitoring. Such features, for example, are available in public cloud computing services such as Amazing Web Services (AWS) Google Cloud Platform (GCP), and Microsoft Azure.

The delivery voyage of Hull 091 from Hobart, Tasmania to Canary Islands, Spain took place between 15 July and 15 August 2019.

3. CLOUD COMPUTING AND PARALLEL PROCESSING

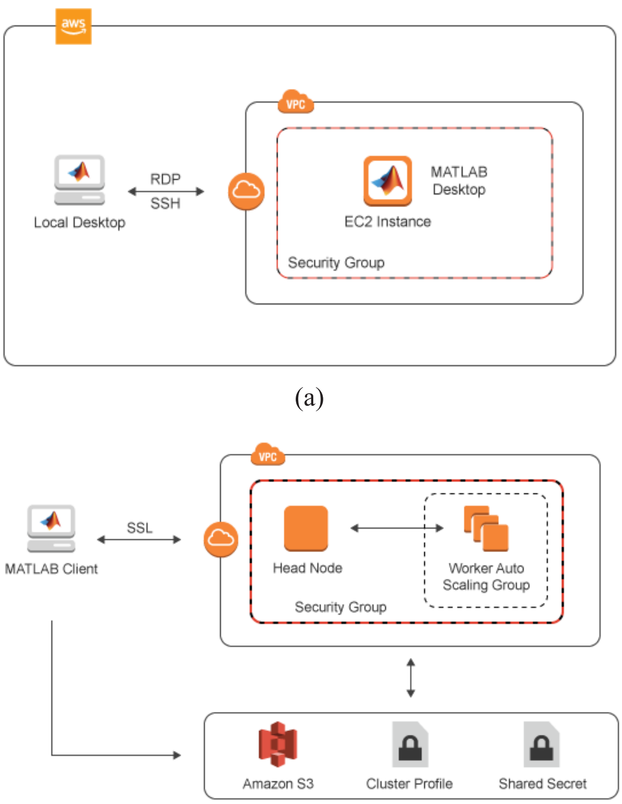
3.1 MATLAB REFERENCE ARCHITECTURES ON AMAZON WEB SERVICES (AWS)

In the proposed system, the data is collected on the vessel and transmitted to a cloud computing resources for processing and analysis, which enables remote monitoring of the vessel. As a public cloud service provider, among many others, AWS provides a broad range of services including computation, storage, analytics, machine learning, Internet of Things, and security. The full list of all services is available from the AWS management console. One of the services that provides all required resources in a specific, predefined format is AWS CloudFormation. The rationale for selecting AWS as the cloud provider is that the data was already stored on AWS S3 and the availability of a MATLAB reference architecture on the platform at the time of project planning.

Figure 3 shows two different solution architectures using MathWorks products on AWS. The CloudFormation templates for each specific solution are available from the MathWorks page on GitHub [10]. Figure 3a presents a simple architecture in which the user can set up MATLAB Desktop on Amazon Elastic Compute (EC2) for a scalable cloud computing. The connection to the compute instance which is in a Virtual Private Cloud (VPC) is available by a Remote Desktop Protocol (RDP) client software and Secure Shell (SSH) tunneling, with the username and password to be specified during the launch of the template. The access of the compute instance to other resources such as Amazon Simple Storage Service (S3) will need to be enabled by attaching specific policy to the compute instance. Various types of EC2 instances are available to select, including general purpose such as M5 series, compute optimised such as C5 series, and accelerated computing such as P3 series. The accelerated computing type comes with graphics processing units (GPU), which is ideal for training deep neural networks and allows to significantly reduce training times when compared with Central Processing Units (CPU). It should be stressed that the MathWorks reference architectures are pre-configured to use single or multiple GPUs and this facilitates the process on both AWS and Microsoft Azure cloud services.

Parallel CPU processing is often required when training various machine learning algorithms and this can be performed by MATLAB Classification Learner or Regression Learner Apps. MATLAB recommends a minimum of 2 virtual Central Processing Units (vCPU) for each worker because two vCPUs share only one Floating Point Unit. To give an example, by selecting m5.xlarge, which comes with 4 vCPUs and 16 GB memory, only 2 workers are available in the cluster, while by selecting m5.24xlarge as a compute instance on AWS, which comes with 96 vCPUs and 384 GB memory, a maximum of 48 MATLAB workers are available to be used in the local parallel processing.

The second architecture (Figure 3b) presents another approach for parallel processing and scaling of the compute workers. In this method, MATLAB workers are distributed across two or several EC2 instances with a dedicated EC2 instance as the head node. The number of workers is adjustable by changing the desired capacity within the autoscaling policy and the connection to the head node is enabled via SSL. This can present a convenient model when there are multiple users and these users have MATLAB on their local computers as they can connect to the parallel processing cluster remotely and perform their analytics. When dealing with files stored in an S3 bucket, access to the bucket should be enabled for each worker by attaching an S3 access policy to the autoscaling group policy.



(a)

(b)

Figure 3. MathWorks reference architectures on AWS (top) MATLAB on AWS (bottom) MATLAB Parallel Server on AWS

Both architectures presented in Figure 3 are supported with MATLAB online licensing, which means individual account holders can setup their own resources on AWS. These architectures were used for processing raw data received from Hull 091 and training ML/DL algorithms to be briefly introduced in Section 4.

3.2 PRODUCTION SERVER

MATLAB Production Server on AWS was considered for real-time analytics and automated data processing. The reference architecture as shown in Figure 4 requires additional cloud resources in comparison with the two earlier architectures discussed. It can for example provide (i) a framework to automate data processing in relation to some customised monitoring functions such as MSI, rain flow cycle counts and peak over threshold; (ii) a ML production pipeline by hosting ML and DL classifiers; and (iii) a mechanism for calling MATLAB functions deployed on the production sever from either a MATLAB client or third-party clients/applications developed in other programming languages for realtime analytics or visualisation. More details of this architecture are given in [10].

Moreover, the analysis process can be fully automated by incorporating serverless resources such as AWS Lambda and DynamoDB for event triggering, concurrent executions and storing processed data. Such data architecture can provide a flexible, scalable, and cost-effective solution, leading to deeper insights through automated data processing. Consequently, it has been proposed that the hull monitoring functions and classifiers can be offered as a service, supporting Hull 091 and future vessels.

It should be noted that this study did not consider several options, and optimal solution is often not readily available. Therefore, it is encouraged to explore various options including software provided by the open source community and other packages. Amazon SageMaker Studio is also recommended for training and deploying ML algorithms as it simplifies compute resource allocations, experimenting with various models and hyper parameter tuning for optimal performance.

It should be noted that there is a limitation in the current work regarding the limited use of cloud computing resources due to connectivity issues and thus careful consideration is required in the implementation of the proposed method.

4. DATA ANALYSES

4.1 OVERVIEW

Developing appropriate functions and methods for analysing data collected from Hull 091 on the cloud is an ongoing project. The purpose of this section is to provide an overview of some standard analyses according to DNV GL guidelines and to highlight some selected analysis methods published in the literature for high speed vessels. This is important as the DNV GL guidelines do not specify signal processing requirements such as filtering, slamming, and whipping identification for such vessels.

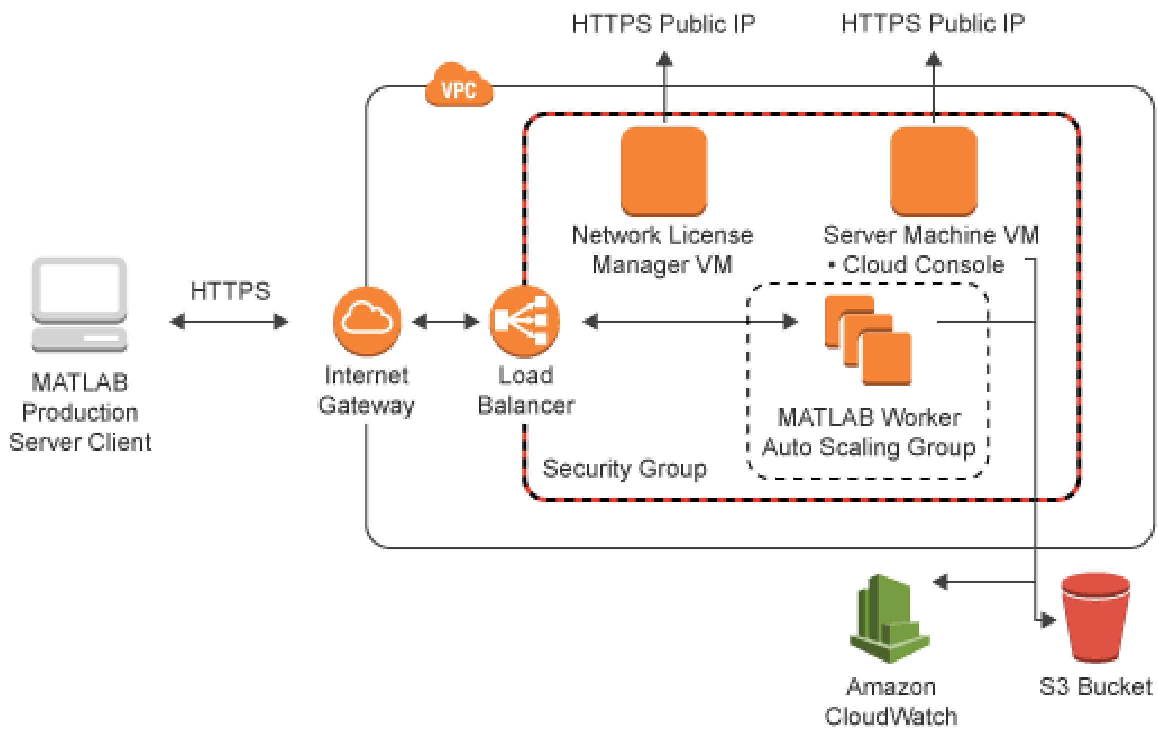


Figure 4. MATLAB Production Server on AWS

For the current project three main areas have been considered for the analyses: (i) evaluating motion comfort through MSI (ii) identifying rainflow cycles for fatigue damage analyses, and (iii) developing a method for slamming identification.

4.2 MOTIONS AND MOTION COMFORT

A function in MATLAB was developed to obtain weighted accelerations, according to ISO standard 2631-1. The outputs were then used to calculate MSI according to the vertical root mean square weighted acceleration (a_w) and exposure time (T) in seconds:

$$\mathrm{MSI} = \frac{1}{3} a_w \sqrt{T} \tag{1}$$

The above equation is a measure of whole-body exposure to low frequency vertical accelerations. Another parameter can be considered for higher frequency accelerations. This is the Vibration Dose Value (VDV) as recommended by DNV GL, which includes accelerations in x-, y- and z-directions. It should be noted that MSIs at other locations can be calculated by considering rigid body motions. Moreover, the vertical and lateral accelerations, pitch and roll can also be considered in evaluating passenger comfort and safety according to some threshold values given in [11]. The method presented in [12] to evaluate motion response amplitudes has also been considered. This requires the Fast Fourier Transform (FFT) of heave, pitch, roll and wave response amplitudes. For Hull 091, a helper function needs to be developed to estimate the wave elevation by relative displacement at the location of the bow ultrasonic sensor. More details for the estimation of wave spectra using an ultrasonic altimeter can be found in [13].

4.3 STRESS CYCLES ANALYSES

The DNV GL procedure for fatigue assessment of ship structures (DNVGL-CG-0129) has been considered for the analyses. The objectives of the analyses are to estimate fatigue rate, defined as the ratio of the measured fatigue damage and the budget damage per unit time. The stress cycle distribution was calculated using the rain flow cycle counting method according to ASTM Standard E-1049 by incorporating the MATLAB rainflow function. The outputs include amplitude and mean stress levels for each cycle, identified through the procedure. More discussions on fatigue and stress distributions for high speed catamarans are given in [14-16]. The standard was considered during the development of the methodology for remotely monitoring the motions and structural responses of the catamaran, and specific sections were implemented as relevant to the project. A combination of strain gauges and accelerometers to measure the structural responses of the catamaran. The strain gauges were used to measure the strain on the structure, and the accelerometers were used to measure the acceleration of the vessel. The stress was then calculated using calibration factors, the strain measurements, and the known properties of the structure [17].

4.4 SLAMMING

Three approaches are mentioned in the rules for slamming identification, using (i) pressure transducers, (ii) an accelerometer in the bow area, or (iii) whipping responses from the global strain sensors. One of objectives of the system that was developed and analysed in this study is to identify slam-induced bending stresses acting on either local or global structures of the ship. Providing an early warning due to slamming in terms of probability of slamming, reporting the number of slam peaks crossing certain thresholds and determining the location of slamming are desirable analyses.

Methods of analysing slam pressures and loads in irregular waves can also be found in [18, 19]. Wet-deck slamming loads pressures, slam induced bending moments, and slamming kinematics of wave piercing catamarans were investigated in regular waves [1, 20-23]. Slamming identification methods for high speed crafts are also summarised in [24], from which the stress derivative threshold [25, 26], whipping-based [27] and fatigue-based criteria [24, 28] can be highlighted. A ML approach for classifying slamming events is also proposed in [3].

4.5 STATISTICAL ANALYSES

According to DNV GL, configuring threshold values, alarm levels, statistical calculations, probability distribution analyses, and trend predictions are other requirements of a standard hull monitoring system. For trend predictions in our proposed system, the establishment of a sequence of all features calculated from each individual sensor was considered However, it is important to note that there may be alternative predictive capabilities and methods that take advantage of the full set of spatially distributed measurements. Other analyses include obtaining several features such as descriptive statistics for each signal, mean zero crossing periods, peak to peak analyses, histogram of peaks, and distribution analyses. In addition, extreme value analyses and the probability of exceeding certain thresholds based on estimated probability curves need to be calculated. A method for predicting structural responses based on hull monitoring data is given [29]. Such analyses should also consider the combined effects of slam-induced and wave induced bending moments [30, 31] so that data driven models can be supported by Finite Element stress analyses and design load determination methods proposed in the literature.

5. LONG SHORT-TERM MEMORY NETWORKS

As mentioned earlier, establishing a sequence of features calculated from each individual sensor is required for trend predictions. Recurrent Neural Networks (RNNs) are for sequencing data and there have been numerous successful applications in various domains such as automated translation, natural language processing, speech recognition and signal processing [32, 33]. The proposal is to train Long Short-Term Memory (LSTM) networks, a variant of RNNs, by feeding data collected from Hull 091, and use MATLAB Production Server to host the LSTM classifiers for near real-time predictions. The network architecture consists of (i) an input layer (i.e. a sequence of data), (ii) an LSTM layer, (iii) a fully connected neural network layer and, (iv) an output layer (i.e. a regression layer).

Figure 5 shows an overview of an LSTM layer with LSTM blocks. Such a layer represents *C* number of features or channels, each with *S* length, where *S* represents the number of time steps. The LSTM layer has *S* LSTM blocks, with the first block requiring the feature vector at the first-time step and the initial state of the network to output *D* number of hidden units. Figure 5 also shows four components of each LSTM block: input gate (*i*), forget gate (*f*), cell candidate (g) (g) and output gate (*o*). The mathematical operation of each component is given in Table 3. There are two activation functions: gate activate function, denoted by σ_g, and state activation function, denoted by σ_c.

Table 3: LSTM block components and formula

Component	**Formula**	
Input gate	$i_t = \sigma_g\left(W_i \mathbf{x}_t + R_i \mathbf{h}_{t-1} + b_i\right)$	(2)
Forget gate	$f_t = \sigma_g\left(W_f \mathbf{x}_t + R_f \mathbf{h}_{t-1} + b_f\right)$	(3)
Cell candidate	$g_t = \sigma_c\left(W_g \mathbf{x}_t + R_g \mathbf{h}_{t-1} + b_g\right)$	(4)
Output gate	$o_t = \sigma_g\left(W_o \mathbf{x}_t + R_o \mathbf{h}_{t-1} + b_o\right)$	(5)

Each component has its own input weights (*W*), recurrent weights (*R*) and bias (*b*) and they are referred to as learnable parameters of an LSTM layer. The cell and hidden states at time step *t* are,

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (6)$$

$$h_t = \sigma_t \odot \sigma_c\left(c_t\right), \quad (7)$$

where $\odot$ denotes element-wise multiplication of vectors, C_t and h_t denote cell state and hidden state at time *t* and other variables are defined in Table 3. The default activations functions in MATLAB LSTM layer are a hyperbolic tangent function (tanh) and sigmoid function for σ_g and σ_c, respectively. Hyper parameters are learning rates, L_2 regularization factors for input weights, recurrent weights and biases and different initializers can be set for each *W*, *R*, and *b* matrices. The number of hidden units needs to also be specified by the user.

The sequence input layer can perform some standard or customised normalisation and by default this is a

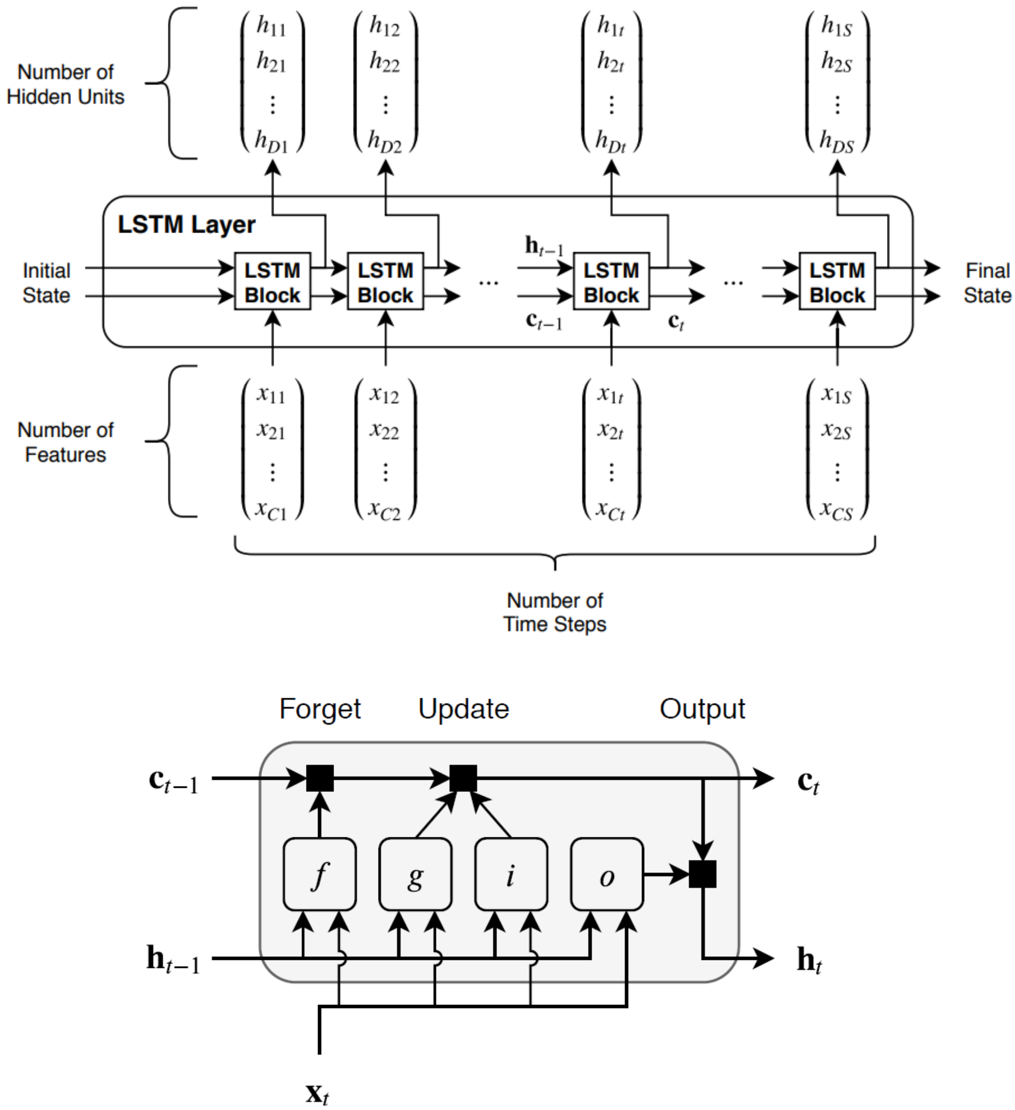


Figure 5. (a) LSTM Layer architecture and, (b) LSTM block architecture [34]

channel-wise normalisation in the recent version of MATLAB. The number of features is a user input. A fully connected layer means all neurons within a layer are connected to all neurons of the previous neurons where inputs are multiplied by a weight matrix and then added by a bias vector. In a fully connected layer, hyper parameters are learning rates and L_2 regularization for input weights and biases. The output layer is a regression layer in which the loss function is defined (i.e. mean squared error function).

It should be noted that the normalisation performed by the sequence input layer is designed to improve the convergence and stability of the algorithm during training. However, this process should not significantly affect the ability of the algorithm to capture the underlying physics of the system, as long as it assumed the normalisation does not introduce bias into the training data.

Deeper networks can be designed by adding either LSTM layers or fully connected layers in combination with dropout layers. A dropout layer is required to avoid overfitting during training stage. The input of a dropout layer is a threshold probability that determines how many input elements are set to zero randomly.

It should be noted that the network architecture for classification is slightly different from regression problems with a classification layer as the output layer, which is preceded by a softmax layer that follows the last fully connected layer of the network. The classification layer takes the outputs of the softmax function, computes the cross-entropy loss, and maps each input to one of the K mutually exclusive classes, where K is the number of classes in a multi-class classification problem. MathWorks documentation [34] provides a full list of functions and layers available in MATLAB for RNNs.

6. CASE STUDY

6.1 TRAINING AND TESTING

Short-term predictions of vessel speed and rms weighted acceleration are considered in this section as examples of using RNNs for the Hull 091 remote monitoring system. The aim is to train individual LSTM networks for each parameter, predict next step values and update the network state at each time step for improved predictions. An initial dataset consisting of all voyages in a month was selected and partitioned (i.e. the first 90% for training and the last 10% for testing the network). The window size for calculating the most frequent (i.e. modal) speed and rms weighted acceleration was 5 minutes, which is the length of each individual file received from the Hull 091 monitoring system. The rms weighted vertical acceleration was calculated using ISO standard 2631-1. Data preparation involved filling out missing data using 1-dimensional interpolation technique. Both parameters were standardized so that they have unit standard deviation and zero mean.

The individual networks were trained so that they can learn to predict the vessel speed and rms weighted acceleration at the next time step. Hyperparameters are learning rate, number of LSTM layers, number of hidden units in each layer, number of iterations or epochs, mini batch size, and threshold values for drop out layers. Over 20 initial combinations of network were selected, and the networks were trained on a single CPU to evaluate the performance of each, including the variations of training loss and training RMSE as a function of iterations during the training phase and also testing RMSE. A table hyperparameter was then created to perform hyperparameter tuning in a more systematic way by using MATLAB Experiment Manager on AWS by considering a P3 instance (i.e. a GPU accelerated computing resource).

A larger dataset was considered compared to the initial dataset by including voyages from September 2019 to June 2020. The train-test split ratios were 0.9-0.1 for this case study. Table 4 shows the range of hyperparameters considered. In the first experiment, a network with 200 hidden units, 3 LSTM layers, 0.002 learning rate and 200 epochs was the champion net among 24 trained networks. A second experiment with 12 options was then

designed to select the final net as listed in Table 3. The networks were trained using the speed dataset. Hyperparameter tuning was not performed for rms weighted acceleration since it was seen that the final net was also a good choice for that.

The training times on the single GPU varied from 53 seconds to 9 minutes depending on the training options considering about 24000 training datapoints. The choice for the optimiser was “Adam” [35] for all training experiments.

6.2 PREDICTION PERFORMANCE

Figure 6 compares predicted speeds and rms weighted accelerations with that observed. The predicted values in this figure are based on the observed values of the immediately previous time step, with a duration of 5 minutes between each time step. The predictions are generally consistent with those observed except for a few points in each voyage, particularly when the ship speed reaches the maximum level during the voyage. RMSE achieved for rms weighed accelerations and speeds were approximately 5% and 9% of the max values observed on the test dataset, respectively. It was found that RMSEs can be minimised by data clustering techniques, but the discussion on the approach is beyond the scope of the present study. Further investigations are recommended to benchmark the best possible performance of LSTM networks for various types of monitoring data as required by DNV GL guidelines in terms of trend predictions and sequencing.

6.3 DEPLOYMENT DISCUSSION

The deployment of DL trained regression models discussed earlier can be useful for two reasons relating to the influence of vertical accelerations on motion comfort. First, it can provide an early awareness of expected MSI and predicted speed so the ship captains can adjust the

Table 4: Hyperparameter tuning details

	Hyperparameters	First experiment	Second experiment	Final
1	Learning rate	0.001, 0.002	0.002, 0.003	0.003
2	Number of hidden units	100, 200	200	200
3	Number of LSTM Layers	1,3,5	3	3
4	Max epoch	100, 200	200	200
5	Dropout Layer Threshold	0.25	0.25, 0.5	0.25
6	Minibatch Size	1	1,5,10	1
Total trained net		24	12	1

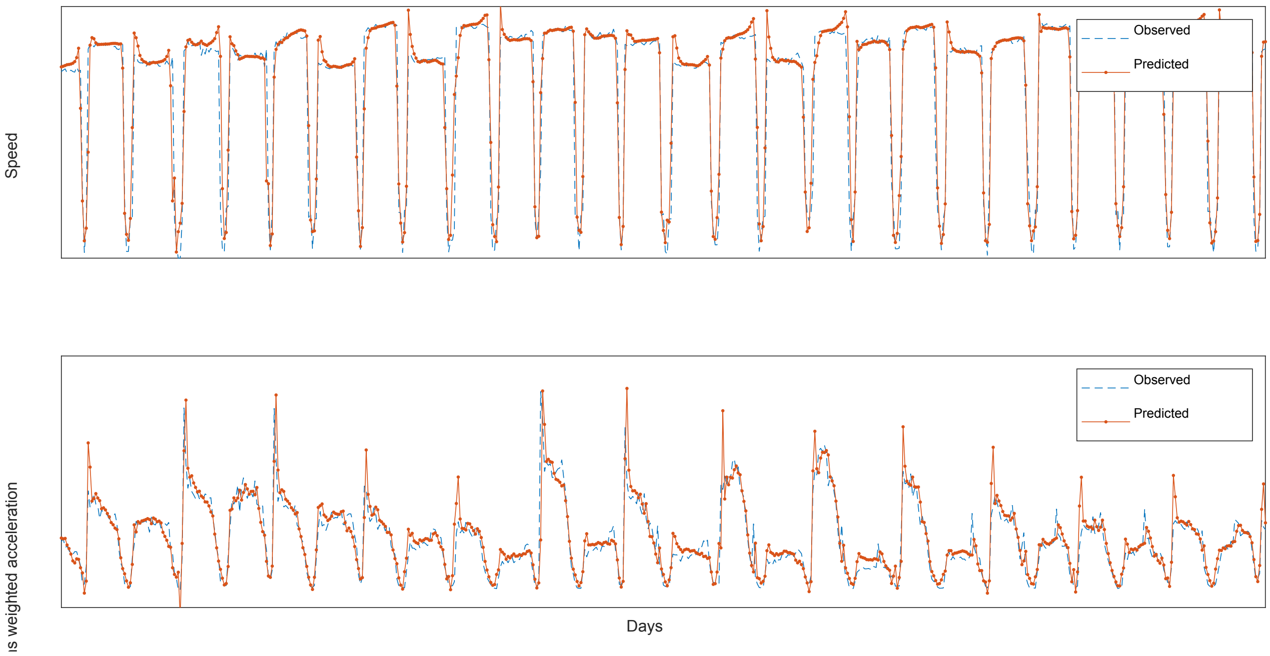


Figure 6. Observed speeds and rms weighted accelerations and predicted values for test voyages using trained LSTM networks

speed accordingly. They may decide to speed up, slow down or maintain speed based on the projected MSI and previous records. Second, a recommender system can be developed so that a suggested speed is returned according to a MSI threshold, which can either be a constant value or a dynamic value calculated for each voyage. The suggested speed needs further investigation to find an optimum.

In relation to the second point, a simplistic approach might be to suggest a lower speed (e.g. 5-10 %) than that projected if the expected MSI is above the threshold according to the industry standard threshold for passenger/vehicle ferries). If, for example, a 5% speed reduction is suggested when the predicted MSI is above the 95th percentile of all time MSI (i.e. for 5% of the time), this could affect the operational efficiency, so another consideration could be when to recommend a higher speed. On the other hand, how recommended speeds will improve rms weighed accelerations and the motion comfort due to the vertical accelerations should be evaluated. The exposure time, encounter wave frequency, relative vessel heading with respect to waves, arrival time, fuel efficiency and other operational factors are also important factors, and thus a full analysis needs to be conducted. Therefore, further research is required to find an optimal speed for passenger comfort by also considering ride control performance on MSIs at various speeds and headings. The optimal strategy is to be formulated as a solution to a multi-criterion optimisation problem.

Furthermore, it should be noted that the MSI should be calculated for various locations to better estimate the overall MSI. Ideally, the overall MSI should be as close as possible to the overall value obtained when the exact locations of seated passengers are known. This is because the interior design of the passenger deck can entice some passengers to be seated in specific locations or in the VIP section, and some may prefer to seat at extreme ends for better view of the sea.

The ISO equation for motion comfort due to whole body vibration is also questionable as it is a recommended metric for the general population. Frequent travellers may likely have better performance against possible sea sickness, in particular for a short exposure time (less than 2 hours).

The connectivity to the ship to the cloud cannot always be presumed and therefore the onboard monitoring system should be used as alternative when the connection to the production server is not available. However, cloud-based predictions and online reports are feasible at the beginning and at the final stages of each voyage. Such reports and dashboards can be valuable for day to day operations at sea in terms of motion comfort.

7. CONCLUSION

A system for hull monitoring from a remote location was proposed as an alternative to standard hull monitoring systems of high-speed catamarans and that can also apply to monohulls. This should be of interest to ship designers and builders. Although the paper does not explicitly address the topics of design load estimations and extreme values analysis of global bending moments, the proposed system has the potential for future integration of these methods due to the scalability of its cloud-based analytical model. This can lead to improved design load estimations and extreme values analysis of both global bending moments and slamming based on full-scale in-service measured data in random seas. In addition, motion comfort and operational efficiency can be of importance to passenger/vehicle operators for day to day operations.

Several options are available for data engineering on a cloud computing environment. The data processing method proposed in this paper incorporated MATALB products on Amazon Web Services (AWS) including MATLAB Production Sever and Parallel Processing toolboxes, as well as other toolboxes for signal processing, machine learning and statistics.

A case study for trend predictions using Recurrent Neural Network (RNNs) was presented, in which sequences of speed and rms weighted acceleration were learned from measured data for the particular operational route of Hull 091. Due to the size of the dataset, GPU accelerated resources on AWS were used. Using MATLAB Experiment Manager, several Long Short-Term Memory (LSTM) network architectures were trained to obtain a final model.

Some data processing requirements and methods for slamming, fatigue and motion comfort were highlighted, mainly from DNV GL guidelines and published papers relevant to high-speed catamarans. Although MSIs can be used for motion comfort predictions as discussed, an optimal strategy is yet to be formulated as a solution to a multi-criterion optimisation problem, which includes vessel speed, rms vertical and lateral accelerations, encounter wave frequency, relative vessel heading with respect to waves, arrival time, fuel efficiency and other operational factors.

A complete data processing framework for high-speed vessels in random waves can be sophisticated. Although not all monitoring systems may benefit from the use of ML approaches, ML approaches can be useful for improving data processing and analysis in the proposed system, as well as in future monitoring systems that do not rely on cloud computing. Such approaches require investigations on feature engineering techniques too. A recommender system for improved motions and fuel efficiency is of interest for future investigation. Exploring novel techniques for onboard motion sickness evaluations, such as using deep neural networks and image processing, could also merit further considerations.

It appears that the classifier should be remotely deployed to an onboard system. However, it is important to note that the training phase of the classifier may still require cloud-based resources. Therefore, the use of shipboard alternatives to address connectivity issues is highly recommended which may lead to further research into the development of more lightweight and efficient approaches, and/or the use of hybrid approaches that combine shipboard and cloud resources for optimising and providing a more robust monitoring system.

8. ACKNOWLEDGEMENTS

This work was undertaken in collaboration between the University of Tasmania, Revolution Design, Incat Tasmania, University of New South Wales Sydney, Italian National Research Council and University College London through the support of the Australian Research Council Linkage Grant number LP170100555. The support of MathWorks team for providing licenses and consultation is appreciated.

9. REFERENCES

1. SHABANI B, LAVROFF J, HOLLOWAY DS, DAVIS MR, THOMAS GA. (2019) *Wet-deck slamming loads and pressures acting on wave piercing catamarans*. Int Shipbuild Prog.;66:201-31.
2. FANG C-C, CHAN H-S (2007). *An investigation on the vertical motion sickness characteristics of a high-speed catamaran ferry*. Ocean Eng.;34:1909-17.
3. SHABANI B, LAVROFF J, HOLLOWAY D, PENEV S, DESSI D, THOMAS G (2019). *Classifying bow entry events of wave piercing catamarans in random waves using unsupervised and supervised techniques.* International Conference on Marine Industry 40. p. 39-54.
4. DNV-GL (2018). Rules for classification: Ships — Hull monitoring systems - HMON: DNVGL;.
5. KUBAT M. *An introduction to machine learning*. second edition ed. New York, NY, US: Springer-Verlag.; 2017.
6. WITTEN IH, FRANK E, HALL MA, PAL CJ (2016). *Data Mining: Practical machine learning tools and techniques.* Cambridge, MA, US: Morgan Kaufmann.
7. WANG J, MA Y, ZHANG L, GAO RX, WU D (2018). *Deep learning for smart manufacturing: Methods and applications.* J Manuf Syst.; 48:144-56.
8. ISMAIL FAWAZ H, FORESTIER G, WEBER J, IDOUMGHAR L, MULLER P-A (2019). *Deep learning for time series classification: a review*. Data Mining and Knowledge Discovery.;33:917-63.
9. HASHEM IAT, YAQOOB I, ANUAR NB, MOKHTAR S, GANI A, KHAN SU (2015). *The rise of "big data" on cloud computing: Review and open research issues*. Information systems.;47:98-115.
10. MathWorks (2020). MathWorks Reference Architectures.

11. YUN L, BLIAULT A, RONG HZ (2018). *High speed catamarans and multihulls: technology, performance, and applications*. Springer.
12. FRENCH BJ, THOMAS GA, DAVIS MR (2015). *Slam occurrences and loads of a high-speed wave piercer catamaran in irregular seas.* Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment. 229:45-57.
13. CHRISTENSEN KH, RÖHRS J, WARD B, FER I, BROSTRÖM G, SAETRA Ø (2013). *Surface wave measurements using a ship-mounted ultrasonic altimeter*. Methods in Oceanography;6:1-15.
14. NEPHI R. JOHNSON JPLMDC (2017). *Response and fatigue assessment of high speed aluminium hulls using short-term wireless hull monitoring*. Structure and Infrastructure Engineering.
15. NEUBERG O, DRIMER N (2017). *Fatigue limit state design of fast boats*. Marine Structures; 55:17-36.
16. MAGOGA T, CANNON S, THOMAS, G (2019). *Interdependencies between variables in fatigue analysis of a weightoptimised naval ship*. First International Symposium on Risk and Safety of Complex Structures and Components, Elsevier. p. 267-74.
17. WARREN M, ALI-LAVROFF J, MCVICAR J, MAGOGA T, SHABANI B, HOLLOWAY D (2022). *Fatigue estimation on a high-speed wave piercing catamaran during normal operations*. International Journal of Maritime Engineering;164.
18. FRENCH BJ, THOMAS GA, DAVIS MR (2015). *Slam occurrences and loads of a high-speed wave piercer catamaran in irregular seas.* Proc Inst Mech Eng Part M J Eng Marit Environ.; 229:45-57.
19. DAVIS MR, FRENCH BJ, THOMAS GA (2017). *Wave slam on wave piercing catamarans in random head seas*. Ocean Eng.;135:84-97.
20. SHABANI B, LAVROFF J, HOLLOWAY DS, DAVIS MR, THOMAS GA (2018). *The effect of centre bow and wet-deck geometry on wet-deck slamming loads and vertical bending moments of wave-piercing catamarans.* Ocean Eng.;169:401-17.
21. SHABANI B, LAVROFF J, DAVIS MR, HOLLOWAY DS, THOMAS GA (2019). *Slam loads and pressures acting on high-speed wave-piercing catamarans in regular waves*. Marine Structures; 66:136-53.
22. SHABANI B, LAVROFF J, HOLLOWAY DS, DAVIS MR, THOMAS GA (2019). *Centre bow and wet-deck design for motion and load reductions in wave piercing catamarans at medium speed*. Ships and Offshore Structures; 1-17.
23. LAVROFF J, DAVIS M, HOLLOWAY D, THOMAS G, MCVICAR J (2017). *Wave impact loads on wave-piercing catamarans*. Ocean Eng.;131:263-71.
23. MAGOGA T, AKSUS S, CANNON S, OJEDA R, THOMAS G (2017). *Identification of slam events experienced by a high-speed craft*. Ocean Engineering;140:309-21.
25. THOMAS GA, DAVIS M, HOLLOWAY D, WATSON N, ROBERTS T (2003). *Slamming response of a large high-speed wave-piercer catamaran*. Marine Technology; 40:126-40.
26. JACOBI G, THOMAS G, DAVIS MR, DAVIDSON G (2014). *An insight into the slamming behaviour of large high-speed catamarans through full-scale measurements*. Journal of Marine Science and Technology;19:15-32.
27. DESSI D, CIAPPI E (2013). *Slamming clustering on fast ships: From impact dynamics to global response analysis.* Ocean Engineering; 62:110-22.
28. THOMAS G, DAVIS MR, HOLLOWAY DS, ROBERTS T (2006). *The effect of slamming and whipping on the fatigue life of a high-speed catamaran.* Australian Journal of Mechanical Engineering;3:165-74.
29. MONDORO A, SOLIMAN M, FRANGOPOL D (2016). *Prediction of structural response of naval vessels based ona vailable structural health monitoring data*. Ocean Eng. 295-307.
30. MCVICAR JJ, LAVROFF J, DAVIS MR, THOMAS G (2015). *Effect of slam force duration on the vibratory response of a lightweight high-speed wave-piercing catamaran.* J Ship Res.; 59:69-84.
31. ALMALLAH I, LAVROFF J, HOLLOWAY DS, SHABANI B, DAVIS MR (2019). *Global load determination of high-speed wave-piercing catamarans using finite element method and linear least squares applied to sea trial strain measurements.* Journal of Marine Science and Technology. 1-13.
32. SHRESTHA A, MAHMOOD A (2019). *Review of deep learning algorithms and architectures*. IEEE Access.; 7:53040-65.
33. MU R, ZENG X (2019). *A review of deep learning research*. KSII Trans Internet Inf Syst; 13:1738-64.
34. MathWorks. lstmLayer. 2020.
35. KINGMA DP, BA J. ADAM (2014). *A method for stochastic optimization*. arXiv preprint arXiv:14126980.