Marine Data Analytics: Machine Learning Algorithms to Optimize Seaweed Growth

Brendan O'Flynn
WSN Group-MNS Centre
Tyndall National Institue
University College Cork
Cork. Ireland
brendan.oflynn@tyndall.ie

Oisin Campion
WSN Group-MNS Centre
Tyndall National Institue
University College Cork
Cork. Ireland
oisincam@gmail.com

Caroline Peres
WSN Group-MNS Centre
Tyndall National Institue
University College Cork
Cork. Ireland
caroline.peres@tyndall.ie

Masoud Emann
WSN Group-MNS Centre
Tyndall National Institue
University College Cork
Cork. Ireland
masoud.emam@tyndall.ie

***Abstract*— Aquaculture farming faces challenges to increase production while maintaining welfare of livestock, efficiently use of resources, and being environmentally sustainable. To help overcome these challenges, remote and real-time monitoring of the environmental and biological conditions of the aquaculture site is highly important. Multiple remote monitoring solutions for investigating the growth of seaweed are available, but no integrated solution that monitors different biotic and abiotic factors exists. A new integrated multi-sensing system would reduce the cost and time required to deploy the system and provide useful information on the dynamic forces affecting the plants and the associated biomass of the harvest. As part of the EU funded IMPAQT project a new multi modal seaweed sensing system was developed incorporating a variety of sensor to investigate Seaweed growth parameters.**

The growth rate of seaweed is significantly affected by wave parameters and sea conditions. The wave characteristics in an aquaculture farm are normally measured using expensive equipment, which is not affordable for many farmers or researchers, and is not easily relocated from place to place to evaluate wave conditions in a variety of locations. This research focuses on developing an artificial neural network that can estimate wave height using acceleration and angular velocity data recorded by a low cost IMU sensor.

Keywords— Aquaculture, Integrated Multi-Trophic Aquaculture, Artificial Neural Network, Wave Characteristics, IMU

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program for the IMPAQT Project under Agreement 774109. Aspects of this work have been supported by Science Foundation Ireland under Grant 12/RC/2289-P2-INSIGHT, 13/RC/2077-CONNECT, 16/RC/3835-VISTAMILK and 16/RC/3918-CONFIRM which are co-funded by the European Regional Development Fund (ERDF) under Ireland's European Structural and Investment Funds Programmes 2014-2020.

I. Introduction

Aquaculture faces challenges to increase production while maintaining welfare of livestock, using resources efficiently, and being environmentally sustainable. To achieve this, remote and real-time monitoring of environmental and biological conditions on these farms is highly important.

Increasingly marine farmers are looking to seaweed and multi trophic aquaculture as a significant growth opportunity. Species such as seaweeds provide many inter-sectorial benefits, and the global demand for seaweed is growing 8.1% per annum. Monitoring of water quality and environmental parameters in an aquaculture setting is well supplied by commercial off-the-shelf sensors. Their measurement parameters include temperature, light radiation, and water quality (dissolved oxygen, pH, salinity, nitrogen). However, these sensors are expensive and usually not co-located with the seaweed growth site.

Current monitoring solutions for seaweed and kelp also include satellite and aerial sensing, which cover large areas effectively. However, these methods do not offer high-resolution, specific local data for growing sites, and are usually limited by turbidity and weather conditions. Integrated Multi-Trophic Aquaculture (IMTA) [1] is acknowledged as a promising solution for the sustainable development of aquaculture. The high level ambition of the IMPAQT project [2] is to develop an intelligent management platform for IMTA. IMPAQT developed and deployed novel sensors and data sources, together with smart systems required for long term autonomous monitoring as shown in Fig 1.

One multimodal sensor system developed [3] as part of IMPAQT is designed to monitor such parameters as light, temperature, depth and accelerometery in a small form factor, embedded system (System level architecture depicted in Fig 2.) capable of edge analytics and data fusion of the sensor streams obtained when deployed in the marine environment as shown in Fig 3.

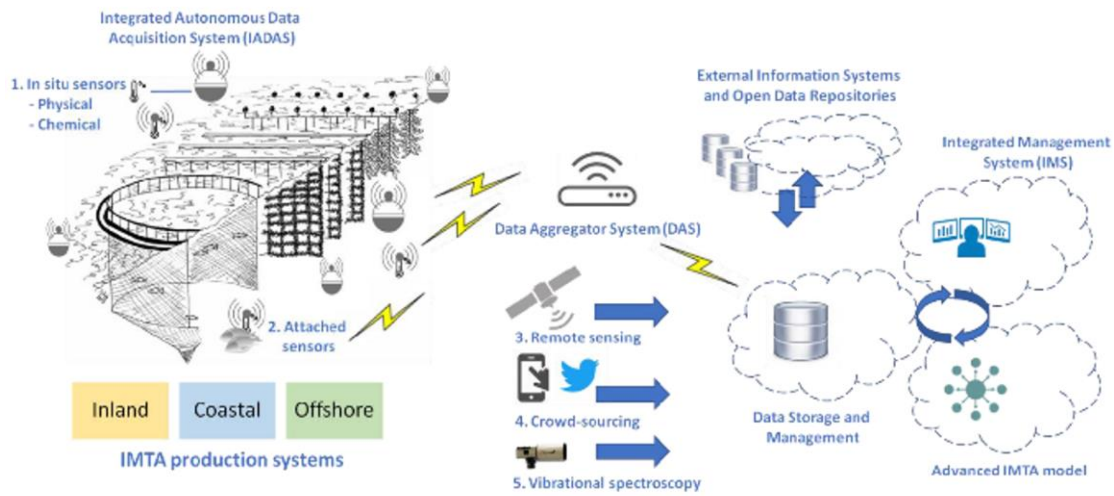

Fig 1. The IMPAQT System Articheture

II. Data Acquisition Platform

A. Data Acquisition System Level Articheture

Based on user requirements defined by end users, and aquaculture farmers, the device system was designed as represented in Fig 2.

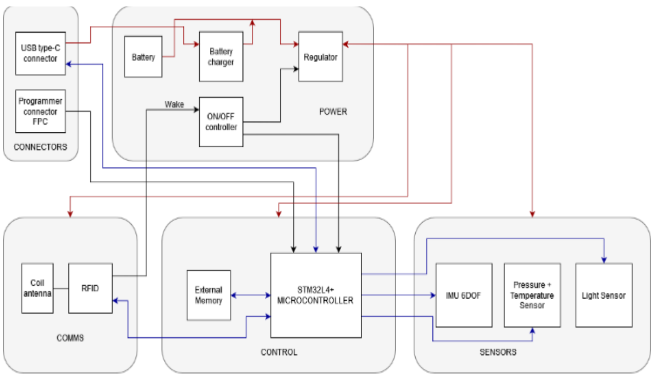

Fig 2. The system level architecture of the IMPAQT seaweed sensor

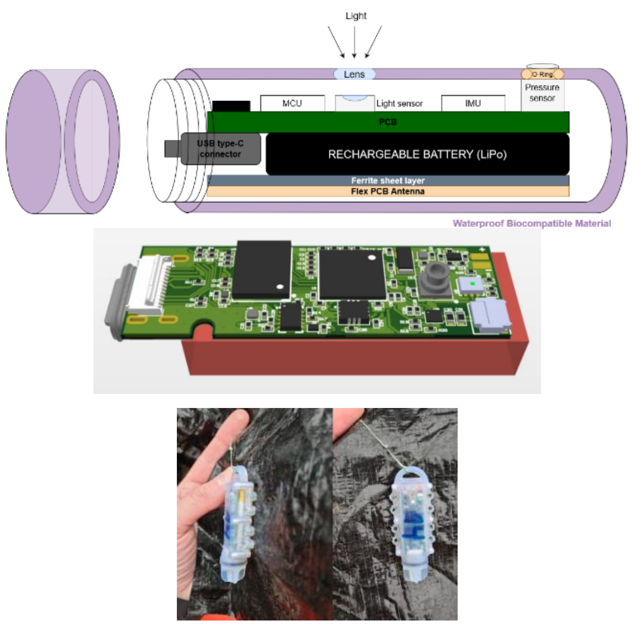

Fig 3. The IMPAQT seaweed sensor system for deployment

Power: Subsystem responsible for providing power, battery management, turning the system on and off, and voltage protections and regulations.

Connectors: USB-C connector for battery charging, communication with PC host, and programming the microcontroller.

Communication (comms): Subsystem responsible for the NFC/RFID communication with host.

Sensors: Contains the sensing capabilities of the system (motion, pressure, temperature, and light).

Control: Subsystem that contains the microcontroller (responsible for the control of the whole system) and the external memory to store data collected.

A microcontroller (MCU) was added to the design to control and manage the whole system, transfer data from sensors to external memory, and communicate with hosts. The requirements for the MCU include: low power consumption, small size, high processing power, large RAM, enough peripherals for all the sensors and external circuits, and Floating Point Unit (FPU). The MCU chosen was the STM32L4R5QII6, due to its low power, machine learning compatibility and high CoreMark and ULPMark scores compared to other MCUs of same power consumption. To communicate, download data collected, and wake-up the system, two communication interfaces are present in the device design: a wired interface using a USB type C connector and a wireless communication interface using NFC/RFID at 15.56 MHz. USB was chosen because of its ease of use and compatibility with any personal computer. The USB connector chosen is waterproof with an O-ring around its external chassis, which adds another layer of protection to the device when at sea (Fig 3).

III. Machine Learning Algorithims

A. Background

Multimodal data analytics is the process of analyzing multiple data sources simultaneously in order to gain a more comprehensive understanding of a given phenomenon. In the context of underwater sensing, multimodal data analytics may involve combining data from multiple sensors or types of sensors (e.g. temperature sensors, depth sensors, etc.) in order to gain a more complete picture of the underwater environment. Machine learning techniques can be applied to the multimodal data in order to identify patterns and make predictions about the environment. This can be useful for a variety of applications, such as measuring wave conditions in the marine environment [4]. In Fig 4 is shown a data capture of IMU raw data sets from the IMPAQT system in deployment on a seaweed farm.

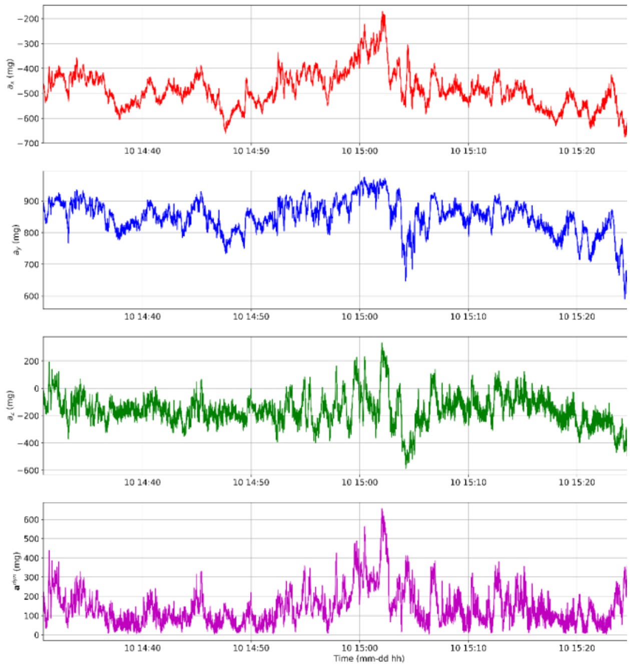

Fig 4. The IMPAQT seaweed sensor and sample IMU raw data set .

A variety of Machine learning algorithms exist including:

1. Ensemble Learning Algorithms: These algorithms combine the predictions of several different models to create a more accurate prediction. Examples of ensemble learning algorithms include boosting, bagging, and stacking.

2. Artificial Neural Networks (ANN): Neural networks use multiple layers of neurons to create complex models that can learn from data. Examples of neural network algorithms include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Self-Organizing Maps (SOMs).

3. Reinforcement Learning Algorithms: Reinforcement learning algorithms use trial and error to learn from their environment. Examples of reinforcement learning algorithms include Q-learning and deep reinforcement learning.

4. Decision Tree Algorithms: Decision tree algorithms use a tree-like structure to classify data. Examples of decision tree algorithms include C4.5, ID3, and CART.

5. Support Vector Machines (SVMs): SVMs use a hyperplane to classify data. Examples of SVM algorithms include linear, polynomial, and Radial Basis Function (RBF).

6. Naive Bayes Algorithms: Naive bayes algorithms use Bayes' theorem to classify data. Examples of naive bayes algorithms include Gaussian, Bernoulli, and multinomial naive bayes.

For the purposes of this early stage research investigating the potential for edge AI, the focus was on the use of the use of as a starting point for the M/L activity. CNNs are a good starting point to build a deep learning models for time series data – particularly for resource constrained embedded systems due to the availability of AI accelerators for such systems (for example the MAX78000 from Analog Devices [9]). Alternative M/L algorithms will be investigated in future work associated with this analysis in the context of their suitability for resource constrained, battery powered embedded systems incorporating edge AI.

IV. Experimental Methodology

A. Data Gathering

Data was gathered in the Deep Ocean Basin at the Lir National Ocean Test Facility [5]. Five IMPAQT seaweed sensors were attached to a buoyant block of foam and placed in the water alongside a wave gauge which measured the true wave height. A number of specific wave types were generated, of varying amplitude and frequency.

The IMU records 3D acceleration and angular velocity. There is also an onboard clock that generates timestamps and keeps track of time, although this is unreliable and drifts over time. The sampling frequency of the IMU sensor is 12 Hz.

B. Data Processing

The data from the IMUs is first normalized to a feature range of (0,1), for each recorded parameter, across all IMUs. That is, the same scaling is used regardless of which IMU the data comes from. Then, the data from the IMUs is divided into each test. The clock on the IMU drifts over time, and so for each test the clocks are manually aligned between the wave gauge and the IMU devices. This is done by plotting the wave gauge against the acceleration data and shifting the time to align the start of the wave on each.

The effect of the alignment shown in Fig 5. The wave gauge data is plotted in black, and in colour is the Gyro_x data from IMPAQT sensor module 14 (scaled and shifted vertically for illustration purposes). Technically, a different time shift is needed for each IMU, but they ultimately this is not needed once we cut around a stable wave.

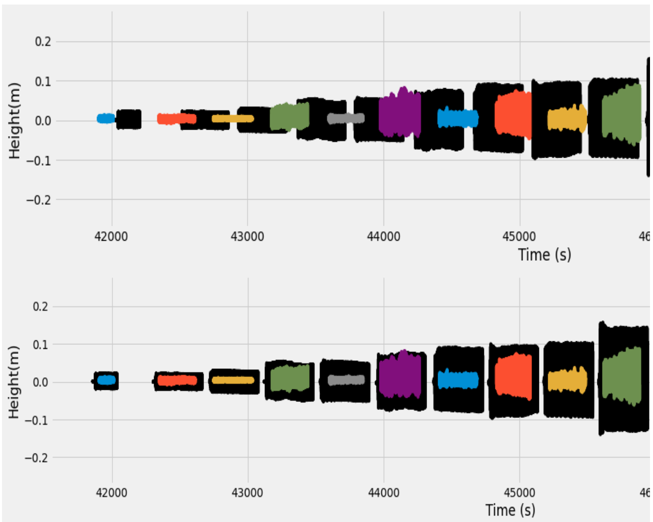

Fig 5. Wave Guage and IMU Data sets (Height on Y axis, Time on X axis)

The next step is to find the time cutoffs of a stable wave for each test, which can be done easily by eye after plotting the gauge data. We then restrict our IMU data to this range. By increasing this restriction slightly, we also ensure that for every IMUs, we only look at data that occurs during the wave, even if they are out of sync by a couple of seconds

1) Window Creation

The next step (shown in Fig 6) is to apply a moving window function to the IMU data. We want our model to take in 32 seconds of IMU data and produce a wave height estimate. Therefore, we divide our IMU data into overlapping windows of 32 seconds. This duration at 12hz produces 384 data readings, and so the result is a 384×6 matrix of data per IMU. The window moves over 3 readings at a time, so the overlap is 31.75 seconds. This is illustrated below, although the overlap is greater in reality.

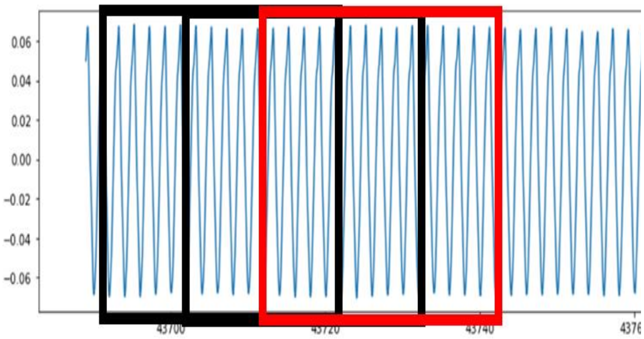

Fig 6. The moving window function for IMU data

14Gyro_x	14Gyro_y	14Gyro_z	14ACC_x	14ACC_y	14ACC_z
-4.375	11585.000	-5923.750	-1027.911	-31.842	-23.790
9821.875	5293.750	-4060.000	-1065.182	-69.906	-44.408
14844.375	-446.250	3255.000	-1054.385	-82.899	-16.653
15968.750	306.250	4777.500	-1028.155	-49.044	-4.880
15106.875	1487.500	-1347.500	-1017.480	-24.034	-29.585
...	...	...	...	...	...
1347.500	-2533.125	7507.500	-1027.484	-171.532	-22.326
-2695.000	1435.000	15155.000	-1003.999	-144.265	-26.047
-2178.750	3408.125	12919.375	-1033.340	-58.499	-38.735
-616.875	1557.500	1356.250	-1059.326	-38.186	-46.787
-1120.000	-1229.375	-3810.625	-1039.196	-110.837	-42.334

Fig 7. Sample IMU data correlated to maximum wave height 15.2 cm

We pair this window of IMU data with the maximum wave height recorded by the wave gauge at the same time. Between 14 tests and 5 IMUs, this results in 33,580 window-height pairs. The 32 second window was originally chosen since this time was necessary to provide an accurate estimation of the wave frequency using the gauge data and Fast Fourier Transform (FFT). This became redundant later, and it could be that a smaller window would be sufficient to give estimates of wave height. A step of 3 data readings was chosen to reduce computational load, since a step of 1 data reading produces about 3 times as many samples. This led to problems with the RAM limitations.

2) Model Structure

A convolutional neural network was selected as they have been shown to been effective in deal with time series data and data structures where there is temporal dependency. Since each window is a 384×6 matrix, the first plan was to treat this matrix as an image and feed it into a 2 dimensional convolutional neural network, along with max pooling layers and dropout layers.

A convolutional layer looks at an image with a number of filters (called kernels), and discovers certain features of the image, from edges and curves to more complicated features. The 2D convolutional approach was effective on the training data, but ultimately misguided. Any horizontal features that the model was learning were incidental and not important, since really the order of the parameters in the matrix should be irrelevant. This reduced the ability of the model to generalize.

There was another problem with the validation set up. Initially, the 33,580 windows were shuffled, and then divided into an 80/20/20 train/validation/test split. However, this set up made it difficult to determine the generalization abilities of the model, since it was being tested on windows from the exact same frequency/height combination that it had been trained on.

The model shown in Fig 8 performed excellently on the test set under this regime with all estimates within 0.5cm of the real height. However, if we trained the model on 13 of the tests only, and tested it on the test left out, then the estimates were completely random and not at all anywhere near the true value.

The interesting question was how the model will perform on a new height/frequency combination. A change was made in the split. The model was trained on all the windows from 13 of the tests, and then validated on all of the windows from the one left out. This was repeated for every test.

The structure of the model also changed. Instead of looking at a 384×6 matrix, the data was looked at as a vector of length 384 across 6 channels, which could then be fed into a one-dimensional CNN. This is analogous as to how a colour image can be viewed as three matrices for the red, green and blue values. One way to think about it is that instead of pairing a matrix with the true wave height, we are pairing 6 columns with the wave height.The complexity of the model was reduced by reducing the number of filters and altering the sizes of the pools in the max pooling layers, in order to reduce the problem of overfitting. Increasing the rate in the dropout layers also helped to prevent overfitting. The final model structure is shown in Figure 8.

3) Model Training

There are several hyperparameters to control when training the model. These were determined based on a number of trials. Convolutional Networks are very sensitive to the tuning of these parameters as they can greatly impact both the speed of training and the overall effectiveness of the model. It took a lot of trial and error to find suitable values. Error in this sense means a model that is very slow to train; or fast to train but underfits on the training data; or fast to train but overfits on the training data. There are rules of thumb available in the literature for sensible starting values, but the overriding message is that it depends on the particular dataset.

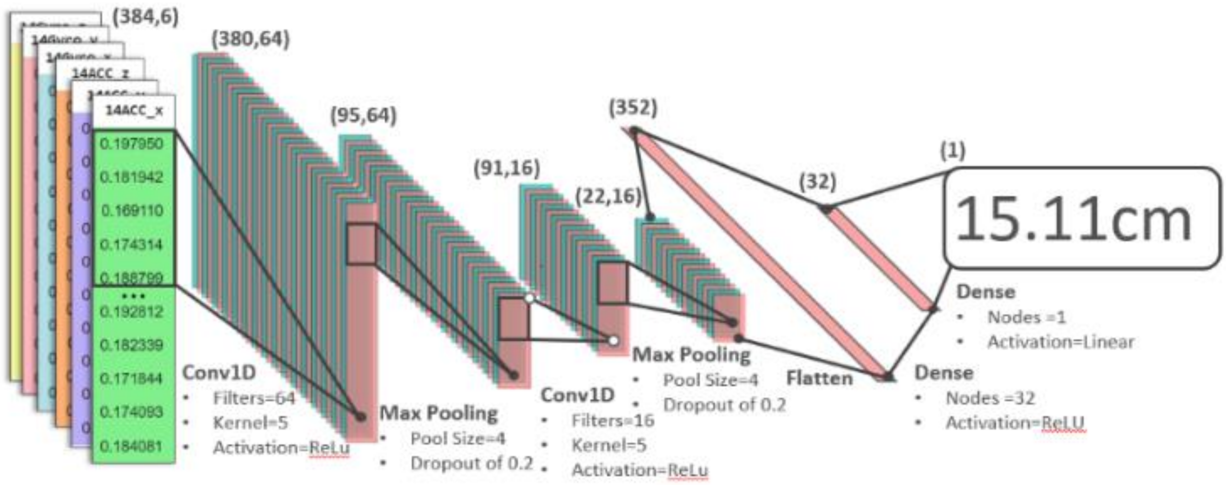

Fig 8. The IMPAQT model structure

The batch size was set to 256, and the learning rate stepped from 0.001 to 0.0001 after 10 epochs. For each test, the model was trained for 120 epochs using the Adam optimizer in Keras [6], with Mean-Squared-Error (MSE) as the loss function. We were able to monitor the loss during training with the TensorBoard [7] extension. There was a callback protocol in place that would save the best model based on validation loss (on the test left out), but this did not need to be used since the model continuously improved up until the last epoch, although improvement was very slow for the last number of epochs. Further exploration of learning rate schedules, along with the optimal number of epochs, could be explored to improve performance.

V. RESULTS

A. Background

As described before, the model was trained on 13 of the tests, and then validated on the final test. This was repeated for all of the tests, with the validation results being collated.

The results are given in the Table 1. The true height was calculated as the average of the labels for that test; the wave height might have changed very slightly during the test, but by no more than half a centimeter. The true height generally differed from the height that was scheduled to be generated, but this was to be expected. The estimates are plotted against the true prediction in Fig 9. The Median Absolute Error (MAE) across all estimates was 1.33cm. Note that this differs slightly from the value in the table, since the value in the table is an average of the average MAE for each test and does not factor in the number of windows generated from each test, which differ slightly due to different windows of stability.

B. Discussion

If we consider an arbitrary cutoff to categories waves into two groups: small (with height less than 11 cm) and large (with height over 11 cm), we can evaluate the accuracy of the estimation in classifying the waves. The 11cm cutoff in the range of amplitudes (2.07cm,24.75cm). It also made sense that the cutoff would provide an almost equal number of large and small tests. For 11 cm, we have 6 large and 8 small as shown on the last two columns of Table 1.

Finally, it made sense to not place the cutoff too near a real test value. If the cutoff was 14.25cm, the average height of test 9, then no matter how accurate the model was, we would still expect about half of the estimates from that test to be classified incorrectly, assuming they are normally distributed. Changing the 11cm cutoff will affect the classification accuracy, but only for tests very near the border. is slightly arbitrary. It made sense that was a near midpoint

Table 1. Table of Results Wave Height Analysis

Test	Train Size	Test Size	True Height	True Freq	Train Loss	Val Loss	Mean % Diff	Mean Abs Diff	Worst Diff	Mean Estimate	Estaimte sd	under 11	over 11
14	33455	2125	20.64	0.2945	3.30E-05	0.0026	24.11	5	7.54	15.65	0.66	0	2125
13	33035	2545	24.75	0.4	1.02E-04	0.0069	32.83	8.13	11.88	16.63	1.65	0	2545
12	33330	2250	15.1	0.28	6.15E-05	6.89E-05	4.48	0.67	2.65	15.1	0.87	0	2250
11	33270	2310	18.55	0.46	9.37E-05	5.77E-04	11.35	2.11	7.13	19.79	2.27	0	2310
10	33345	2235	13	0.278	6.23E-05	2.36E-04	10.68	1.39	3.72	12.12	1.14	179	2056
9	33015	2565	14.25	0.5	1.05E-04	2.65E-04	10.32	1.47	3.21	15.56	0.96	0	2565
8	33010	2570	10.24	0.27	9.70E-05	2.06E-04	12.4	1.27	3.24	9.12	0.9	2429	141
7	33195	2385	9.32	0.57	6.30E-05	8.48E-05	8.9	0.83	1.9	9.14	0.86	2372	13
6	32885	2695	8.93	0.26	1.01E-04	4.33E-04	17.82	1.6	4.8	7.44	1.42	2695	0
5	32805	2775	7.29	0.67	9.63E-05	1.16E-04	12.54	0.9	2.98	7.67	0.95	2775	0
4	33180	2400	5.23	0.26	7.34E-05	1.83E-04	22.91	1.2	3.34	5.25	1.37	2400	0
3	33040	2540	4.98	0.8	9.05E-05	3.41E-04	31.99	1.58	4.22	4.55	1.6	2540	0
2	32790	2790	2.85	0.25	7.68E-05	6.19E-05	24.7	0.7	1.47	3.56	0.344	2790	0
1	32185	2545	2.07	1	1.14e0-4	2.30E-03	218.15	4.51	6.95	6.58	1.16	2545	0
						Means:	18.18354969	1.645482882	3.986764617		1.0561848		
						Median:	15.18	1.43	3.53		1.0561848		

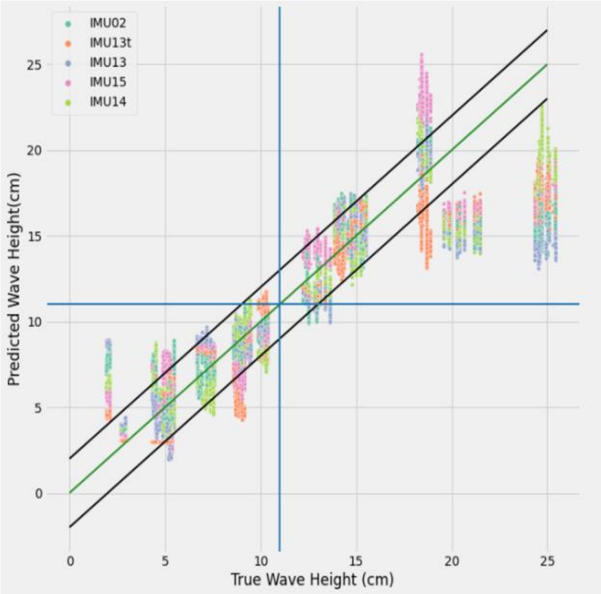

Fig 9. The IMPAQT wave height estimation Vs predicted values

The estimates were then classified and compared to the true estimate. Overall, the model correctly classified 98.9% of windows as large or small. The blue lines in the Fig 9 portray a 2×2 confusion matrix. Dots in the lower left and upper right are in the correct class, and any others are classified incorrectly.

The absolute test accuracy is a more interesting result to investigate. In the Fig 9, the black lines are 2cm error bars, that is any dots within the bars were within 2cm of the true value. The model clearly performed the worst on the final two waves and the very first wave, which should make sense. For a model trained on waves of amplitude in the range (2cm, 20cm), we would not necessarily expect it to perform well outside that range. These outlier values also affect the mean absolute error, which is why the median is also reported.

For the question of generalization, we must define carefully what we want the model to achieve. One level of generalization is that a model trained on waves in the range (10cm,20cm) should be able to perform well on the range (5cm, 25cm). Another level of generalization is that a model trained on wave of amplitude 2cm, 4cm, 6cm, 8cm, 10cm should perform well on waves of amplitude 3cm, 5cm, 7cm, 9cm. There are also questions about the frequency of the waves, the position of the IMU and the stability of the waves.

It appears the model does not perform well outside of the test range. It does perform well on new values within the test range, for both frequency and height, although the frequency is less varied.

The model also performs remarkably well regardless of the position and orientation of the IMU device, despite the effects of gravity which skew the acceleration readings. For the tests, the devices were in a fixed orientation, and it is not clear how the model would perform on a device that rapidly and drastically changes orientation. It is also not clear how the model would perform on a new fixed orientation that is different to the 5 from the test. The Mean Absolute Error was 2.21cm. The mean is larger than the median due to the presence of outliers; for the very largest waves, and the very smallest, the model under (and over) predicted those windows, since when those waves are left out, it hasn't seen anything on that scale before.

VI. Conclusions & Future Work

The wave model performed very well in cross validation using the data from the Deep Ocean Basin. This performance was unaffected by the orientation of the IMU device. There are indications that the model performs well on new values within the training range, but less well outside the training range. Making the system appropriate for wave height estimation in the field to understand the opimisation of seaweed growth parameters. In future work a variety of ML techniques will be contrasted and compared as regards their suitability for edge AI applications in resource constrained systems

Acknowledgment

This work has been funded as part of the IMPAQT project [2]– and it has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 774109. Aspects of this publication have been funded in part by a research grant from Science Foundation Ireland (SFI) - co-funded under the European Regional Development Fund under Grant Number 16/RC/3835 – VISTAMILK 16/RC/3835, INSIGHT 13/RC/2077_P2 and CONNECT 13/RC/2077.

The authors of the paper would like to appreciate the help and supports of MaREI, the SFI Research Centre for Energy, Climate and Marine [8]; and LiR National Ocean Test Facility [5], UCC, Ireland for facilitating the deployment of the IMPAQT seaweed sensor system. An especial thanks to Dr. Milad Zabihi and Mr. Ian Power for their help and support during the tests.

References

[1] B. H. Buck, M. F. Troell, G. Krause, D. L. Angel, B. Grote, and T. Chopin, "State of the art and challenges for offshore integrated multi-trophic aquaculture (IMTA)," *Frontiers in Marine Science*, vol. 5, p. 165, 2018.

[2] "IMPAQT Project." https://impaqtproject.eu/ (accessed 2023).

[3] C. Peres, M. Emam, H. Jafarzadeh, M. Belcastro, and B. O'Flynn, "Development of a low-power underwater nfc-enabled sensor device for seaweed monitoring," *Sensors*, vol. 21, no. 14, p. 4649, 2021.

[4] M. Emam *et al.*, "Wave height estimation using a novel seaweed-attached sensor," 2021.

[5] "Lir National Ocean Test Facility." www.lir-notf.com (accessed 2023).

[6] F. Chollet. "Keras." https://keras.io (accessed 2023).

[7] "TensorBoard." https://www.tensorflow.org/tensorboard (accessed 2023).

[8] "MaREI, the SFI Research Centre for Energy, Climate and Marine." https://www.marei.ie/ (accessed 2023).

[9] https://www.analog.com/en/products/max78000.html#product-overview (accessed September 2023)