

This file is part of the following work:

Jahanbakht, Mohammad (2022) *Deep learning for internet of underwater things and ocean data analytics*. PhD Thesis, James Cook University.

Access to this file is available from:

<https://doi.org/10.25903/vwck%2Ds164>

Copyright © 2022 Mohammad Jahanbakht.

The author has certified to JCU that they have made a reasonable effort to gain permission and acknowledge the owners of any third party copyright material included in this document. If you believe that this is not the case, please email

researchonline@jcu.edu.au



Deep Learning for Internet of Underwater Things and Ocean Data Analytics

Mohammad Jahanbakht

A thesis submitted for the degree of

Doctor of Philosophy

at the

College of Science and Engineering
James Cook University

Supervisors:

Dr Mostafa Rahimi Azghadi
and Prof. Wei Xiang

2022

Declaration

I hereby declare that the work presented in this thesis entitled “Deep Learning for Internet of Underwater Things and Ocean Data Analytics” is my own work and has not been submitted in any form for another degree or diploma at any university or institute of territory education. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Mohammad Jahanbakht

October 2021

Statement of Access to This Thesis

I, the undersigned, the author of this work, understand that James Cook University will make this work available for use within the University, and via the Australian Digital Thesis Network, for use elsewhere.

I understand that as an unpublished work, a thesis has significant protection under the Copyright Act. I do not wish to place any restriction on access to this thesis. However, any use of its content must be acknowledged and could be potentially be restricted by future patents.

Mohammad Jahanbakht

October 2021

Acknowledgments

First and foremost, I would like to thank my primary adviser, *Dr Mostafa Rahimi Azghadi*, whose consistent assistance, inspiring guidances, and expert scientific advice have made this PhD thesis possible. His precious support was extremely helpful in my academic and professional development at James Cook University (JCU).

I am grateful to my secondary adviser, *Prof. Wei Xiang*, who contributed valuable expertise and excellent advice throughout all works in this thesis. More specifically, his constructive comments had a great impact on publishing the chapters of this thesis.

I am thankful to our collaborators *Prof. Lajos Hanzo* from Southampton University, *Dr Barbara Robson* from Australian Institute of Marine Sciences, and *Dr Nathan Waltham* from Australian centre for Tropical Water and Aquatic Ecosystem Research, who provided expert advice to improve the contexts of the published works.

I gratefully acknowledge the funding received from the Australian Government research training program scholarship. I am also thankful to my friendly colleagues and the supportive workplace at the college of science and engineering, JCU.

Finally, I dedicate this work to my wife, *Sepideh*, my *parents*, my children, *Amirali and Ehsun*, and my brother, *Taghi*; for their endless love and support.

Abstract

The Internet of Underwater Things (IoUT) is an emerging technological ecosystem developed for connecting objects in maritime and underwater environments. IoUT technologies are empowered by a large number of deployed sensors and actuators. From scientific perspectives, these ubiquitous sensing devices are considered as data collecting tools, which can be augmented with machine intelligence and big data analytics both for automated monitoring and for prospective studies. In this thesis, we have comprehensively surveyed the IoUT and marine data analytics to address several research gaps.

Using publicly available marine datasets, we have proposed three deep learning models for marine data timeseries. The proposed models include a timeseries forecasting ensemble of Deep Neural Networks (DNN) for sea surface temperature prediction, a next-frame prediction DNN framework for predicting total nitrogen in the Great Barrier Reef (GBR), and a Transformer-based next-frame prediction DNN framework for predicting total sediment in the GBR. Finally, an accurate and energy-efficient platform has been proposed for IoUT image processing for fish segmentation in realistic underwater video footages. This fast and low-bandwidth platform consists of a compressed DNN with low energy consumption and real-time edge-based inferencing on an embedded GPU.

The outcome of this thesis can facilitate developing tools for 1D and 2D spatiotemporal timeseries predictions. The proposed highly-accurate forecasting models can support decision makers to reach target water quality outcome in wide geographical areas like the GBR. Furthermore, the edge processing technique proposed in this thesis can take the marine video processing capabilities to the next level of intelligent in-situ IoUT systems.

Publications

Journal Articles	Impact Factor
M. Jahanbakht, W. Xiang, L. Hanzo, and M. Rahimi Azghadi, “Internet of Underwater Things and big marine data analytics—a comprehensive survey,” <i>IEEE Communications Surveys & Tutorials</i> , vol. 23, no. 2, pp. 904–956, Jan. 2021.	25.2
M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sea surface temperature forecasting with ensemble of stacked deep neural networks,” <i>IEEE Geoscience and Remote Sensing Letters</i> , vol. 19, pp. 1502605–1502609, Aug. 2021.	3.97
M. Jahanbakht, W. Xiang, B. Robson, and M. R. Azghadi, “Nitrogen prediction in the Great Barrier Reef using finite element analysis with deep neural networks,” <i>Environmental Modelling & Software</i> , vol. 150, pp. 105311–105327, Apr. 2022.	5.3
M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sediment prediction in the Great Barrier Reef using vision Transformer with finite element analysis,” <i>Neural Networks</i> , vol. 152, pp. 311–321, Aug. 2022.	8.1
M. Jahanbakht, W. Xiang, N. J. Waltham, and M. R. Azghadi, “Distributed deep learning in the cloud and energy-efficient real-time image processing at the edge for fish segmentation in underwater videos,” <i>IEEE Access</i> , Aug. 2022.	3.4

Contents

List of Tables	xi
List of Figures	xiv
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Problems	2
1.3 Original Contributions	3
1.4 Thesis Organization	5
2 Literature Review	7
2.1 Introduction	7
2.2 Internet of Underwater Things	11
2.2.1 Underwater Sensors and Devices for Data Collection	13
2.2.2 IoUT Communications	14
2.2.3 Improving IoUT Link Reliability	17
2.2.4 Underwater Channel Modeling	22
2.2.5 IoUT Network Topologies	24
2.2.6 Edge Computing in IoUT	25
2.2.7 Section Summary	27
2.3 Big Marine Data	28
2.3.1 Marine Data Acquisition	29
2.3.2 Marine Data Classification	32
2.3.3 Distributed and Cloud-based BMD Processing	41
2.3.4 Marine Data Applications	44
2.3.5 Section Summary	50
2.4 Machine Learning for BMD Analytics	50
2.4.1 Machine Learning Techniques	51
2.4.2 Deep Learning Frameworks and Libraries	53
2.4.3 Marine Data Cleaning for Machine Learning	56
2.4.4 Feature Extraction for Marine Data Analytics	59

2.4.5	Hardware Platforms for ML in IoUT	60
2.4.6	ML Techniques in BMD Applications	64
2.4.7	Section Summary	72
2.5	Challenges and Future Directions in IoUT and BMD	72
2.5.1	Underwater Network Management System	72
2.5.2	Energy Conservation and Harvesting in IoUT Devices	74
2.5.3	Development of Low-cost and Affordable Sensors	75
2.5.4	Large-scale IoUT Underwater Communications	77
2.5.5	Dynamic IoUT Signal Routing and Traffic Control	78
2.5.6	Deleterious Effects of Imprecise Channel Modeling	80
2.5.7	Sparse and High-maintenance Sensing Devices in IoUT	81
2.5.8	Poor Underwater Positioning and Navigation	81
2.5.9	Non-Destructive Testing in Underwater Applications	83
2.5.10	Lack of Strong Data Leveraging Tools	84
2.5.11	Training Deep Networks	85
2.5.12	Degraded Underwater Images	85
2.6	Conclusion	86
3	Sea Surface Temperature Timeseries Forecasting	87
3.1	Introduction	87
3.2	Background and Problem Formulation	88
3.3	Network Architecture	90
3.4	Results and Discussions	93
3.4.1	Description of Data Sources	94
3.4.2	Prediction Accuracy and Comparison	95
3.5	Conclusion	97
4	Finite Element DNN for Nitrogen Distribution Prediction in the Great Barrier Reef	98
4.1	Introduction	98
4.2	Background and Problem Definition	101
4.3	Proposed Model	104
4.3.1	FE-DNN Model	104
4.3.2	Stiffness Matrices	108
4.3.3	Physics-informed Neural Network	111
4.4	Results and Discussions	112
4.4.1	Data Sources	112

4.4.2	Prediction Accuracy	116
4.4.3	Computational Complexity	119
4.4.4	Ablation Study	120
4.4.5	Limitations	121
4.4.6	Future Directions	122
4.5	Conclusion	123
5	Finite Element Transformer for Sediment Distribution Prediction in the Great Barrier Reef	124
5.1	Introduction	124
5.2	Proposed Model	127
5.2.1	Finite Element Analysis	129
5.2.2	Stiffness Matrices	130
5.2.3	FE-Transformer Architecture	134
5.2.4	Physics-Informed Neural Network	134
5.3	Results and Discussions	135
5.3.1	Data Sources	135
5.3.2	Prediction Accuracy	138
5.3.3	Computational Complexity	142
5.3.4	Ablation Study	143
5.3.5	Limitations and Future Directions	143
5.4	Conclusion	144
6	IoUT Distributed Training and Edge Computing	145
6.1	Introduction	145
6.2	Fish Segmentation	148
6.2.1	Modified U-Net Architecture	148
6.2.2	DeepFish Dataset	149
6.3	Distributed Training	149
6.3.1	Cloud-based Distributed Computer Systems	150
6.3.2	Distributed Training Results	152
6.4	Underwater Edge Computing	153
6.4.1	Energy Management at the Edge	154
6.4.2	Edge Inferencing Results	160
6.4.3	Edge Inferencing Speed	160
6.5	Conclusion	161

7 Conclusion and Future Studies	163
7.1 Conclusion	163
7.2 Future Studies	165
References	167

List of Tables

2.1	The Contribution of This Thesis in the Field of IoUT and BMD Analytics, Compared to Other Previously Published Works	10
2.2	IoUT Sensors to Measure Underwater Physical, Optical, Fluid, and Chemical Parameters [1–5]	15
2.3	Data-Rates and other Characteristics of Communication Technologies in Underwater Applications [6–8]	20
2.4	Location of The Processing Unit in Different IoUT Edge Computing Paradigms	27
2.5	List of Data Acquisition Tools in Underwater Applications with the Percentage of Use of Each Item in Related ML Articles	30
2.6	Open Access and Active Observatories Worldwide; Providing Up-to-date Primary Data	35
2.7	Open Access Datasets of Still-images, Taken from Underwater Organisms .	36
2.8	Major Apache Frameworks Suggested for Distributed Processing of BMD .	43
2.9	Deep Networks with Input, Hidden, Mid-output, and Output Layer Neural Nodes to Evaluate Static, Dynamic, Sequential, or Hierarchical Input Data Types [9–13]	54
2.9	(Continued)	55
2.10	Comparing the Best Open-source Software Frameworks and Libraries for Deep Learning	55
2.11	Image Clustering Methods and Algorithms in Underwater Applications . . .	59
2.12	Data Processing and Machine Learning Feature-sets in Published Underwater Applications	61
2.13	Comparing Different Parallel Processing Hardware Platforms to Carry out a Typical Machine Learning Algorithm	63
2.14	Comparison of a selected number of CNN architectures submitted to the Annual ImageNet Challenge [14, 15]	66
2.15	Comparing the Precision of Multiple Machine Learning Techniques to Recognize Three Different Underwater Species	68
2.16	Comparing the Precision of Multiple Machine Learning Techniques to Recognize Objects in Sonar Imagery	69

2.17	Comparing the Mean Squared Error of ML Techniques in Sea Surface Temperature Prediction at Philippine Sea	71
3.1	Temporospatial Coordinates of the Sea Areas Under Study and Their Corresponding Land Weather Stations	89
3.2	Mean Squared Error of Area-Averaged SST Forecasting at the Bohai Sea, Compared with the Different Schemes Used in [16]	93
3.3	Mean Squared Error of Area-Averaged SST Forecasting at the South China Sea, Compared with the Different Schemes Used in [16]	94
3.4	Mean Squared Error of Area-Averaged SST Forecasting at the North Pacific Ocean, Compared with the Proposed Model in [17]	94
4.1	Conceptual comparison between FE-DNN and FEA	105
4.2	Statistics of the values of TN_{PDE} from <i>eReefs</i> marine models	115
4.3	MSE and R^2 metrics in TN prediction of the test dataset in 2018, using elements of size 7×7 pixels	115
4.4	Comparing the performance of the proposed FE-DNN model with two recently published works in the literature	116
4.5	Comparing time and memory complexities of the proposed FE-DNN with two recently published works in the literature	120
4.6	Evaluating the elimination of selected blocks on accuracy of the one-day TN forecasting	121
5.1	Statistics of the TS Values from <i>eReefs</i> (P_{PDE})	136
5.2	MSE and R^2 of TS Prediction for the Testing Dataset in 2018, using Elements of Size 7×7 Pixels	139
5.3	Comparing the Performance of Our FE-Transformer Model with Two Recently Published Works in Literature	141
5.4	Evaluating the Elimination of Selected Blocks on Accuracy of the Model	144
6.1	Hardware Specifications of the Utilized Local Computer, Distributed Computers, and the Edge Device	150
6.2	SCCE and SCF Loss Metrics, as well as SCCA Accuracy Metric of the Modified U-NET Model, Which has been Trained on a Local Computer and a Distributed Computer System	153
6.3	Average Power Consumption of the Original Segmentation DNN with FP32 Parameters, Compared with FP16 and Int8 Compressed DNNs	156

6.4 SCCE, SCF, and SCCA Metrics of the Original Segmentation DNN with
FP32 Parameters, Compared with FP16 and Int8 Compressed DNNs 157

List of Figures

1.1	Outline of the thesis structure.	5
2.1	Qualitative relationship between the diverse system components of the IoUT and BMD analytics, starting from underwater sensors and ending up to ML solutions and future directions.	11
2.2	Outline of Chapter 2 at a glance.	12
2.3	Reliable and energy-efficient multi-cluster network topology [18].	20
2.4	Reported network topologies in IoUT applications.	25
2.5	Schematic view of three different optical rays, which are reflected back to the camera, in active underwater imaging.	37
2.6	Distributed big data frameworks used in data analytics papers published in IEEE and Web of Science databases.	42
2.7	Using a fixed beacon at known location to decrease the location-uncertainty of an AUV from ‘A’ (intersecting with ‘B’) to ‘C’.	48
2.8	Conceptual categorization of the main topics in Section 2.4.	51
2.9	Percentage of deep learning usage in all machine learning publications, as searched between IEEE Explore as well as Web of Science databases.	53
3.1	Polar plot of (a) the area-averaged SST at Bohai Sea and (b) the air temperature at its iso-latitude Dalian City, both from the NOAA data source.	90
3.2	Block diagram of (a) the voting ensemble model for SST forecasting, which consists of two (b) stacked LSTM-MLP deep neural networks.	92
3.3	Time and memory complexities with respect to the number of data points for one-day $F^+(\text{SST}, \text{Air})$ forecasting in Bohai Sea.	96
4.1	(a) The study area frame is meshed into small pixels that hold the temporal TN values of each of the past days until today. The pixels of each of the N historical frames are grouped into 5×5 elements, which are then used to predict the TN value of a target pixel F days after today. (b) The elements are then swept across the study area to predict all the target pixels one-by-one, while leaving a narrow 2-pixel margin of unpredictable TN values.	102

4.2	The architecture of the proposed FE-DNN network to predict a TN frame in F days later, where d represents today. Here, N input elements (as shown in Fig. 4.1) surrounding a target pixel with coordinates of (x, y) as well as N stiffness matrices (calculated for month M by (4.3)) are required to predict the TN value at the target pixel $P_{d+F}^{x,y}$. The output of the model for the target pixel is fed into Fig. 4.5. The shown kernel sizes, filter numbers, activation functions, etc. are optimized for this study, and they can be different in other next-frame prediction applications.	106
4.3	Flowchart of the FE-DNN workflow from input data to forecasted results.	107
4.4	(a) Stiffness matrix calculation for the linear spring problem in Table 4.1 in the presence of white Gaussian noise. (b) The stiffness matrices in normalized logarithmic scale of three typical geolocations ① at (11.18°S, 143.25°E), ② at (11.18°S, 148.17°E), and ③ at (13.70°S, 147.57°E) in three months of January, May, and September, where the element size is 7×7	109
4.5	PINN loss function applied to our proposed FE-DNN model.	111
4.6	Effect of element size on the MSE metric in (a) 3-day, (b) 5-day, and (c) 7-day forecasting horizons, where number of past days used for training is $N = 3$. The horizontal dashed lines show the average MSE for the 7×7 element size.	113
4.7	True values, predictions, and their absolute differences (multiplied by 10^3) for the 2018 test TN_{\log} dataset in the GBR, for (a) 1-day (b) 3-day, (c) 5-day, and (d) 7-day forecasting horizons.	114
4.8	Pairwise mathematical distances between the input values of the last day, true values in F days later, and our prediction, using elements of size 7×7 pixels and $N = 3$, along with their corresponding DI.	117
4.9	Time and memory complexities with respect to the element size for one-day forecasting horizon, with $N = 3$	119
5.1	Three input frames ($N = 3$) are used to predict the next TS frame in F days after today (d). Towards this end, the pixels of historical TS values in the GBR are grouped into 3×3 elements $\mathbf{E}_{d-2}^{x,y}$, $\mathbf{E}_{d-1}^{x,y}$, and $\mathbf{E}_d^{x,y}$. The central coordinate (x, y) of these historical elements will be adjusted to predict TS at (a) pixel (2, 2) (i.e., $P_{d+F}^{2,2}$); and (b) pixel (2, 3) (i.e., $P_{d+F}^{2,3}$), while leaving a 1-pixel margin of unpredictable TS values. The 3×3 element size is used here for illustration purposes only. This size can be optimized to better address the requirements of TS prediction in the GBR.	128

5.2	(a) Stiffness matrix of the classic linear spring problem in the presence of Gaussian noise. (b) Normalized logarithmic scale stiffness matrices of three typical geolocations: Ⓐ at (11.18°S, 143.25°E), Ⓑ at (11.18°S, 148.17°E), and Ⓒ at (19.10°S, 147.93°E), where the element size is 7×7	131
5.3	Architecture of the proposed FE-Transformer network to predict pixel $P_{d+F}^{x,y}$ in F days after today (d), where the N input elements are illustrated in Fig. 5.1, and the N stiffness matrices for month M are calculated by (5.5). The shown kernel sizes, filter numbers, activation functions, Transformer layers, etc. are optimized for TS prediction in the GBR.	132
5.4	Effect of element size on the metrics of (a) MSE and (b) R^2 with 3-day forecasting horizons and $N = 3$. The results are averaged per quarter of the testing year of 2018.	137
5.5	True values, predictions, and their absolute differences for TS_{\log} in the GBR at 2018 test dataset, for (a) 1-day, (b) 3-day, and (c) 7-day forecasting horizons, using 7×7 elements with $N = 3$	138
5.6	Time and memory complexities with respect to the element size for one-day forecasting horizon with $N = 3$	142
6.1	Modified architecture of the U-NET convolutional neural network for underwater image segmentation. The weights of the encoder part are transfer-learned from a pretrained MobileNetV2 on the ImageNet. The decoder part employs Pix2Pix for upsampling.	147
6.2	The SCCE loss metric convergence for both (a) the augmented train and (b) the augmented validation datasets, versus the training time. The model is initially transfer learned on ImageNet and is shown while fine tuning with the DeepFish dataset.	151
6.3	Conceptual comparison between wired underwater communication and wireless communication made possible by performing data processing at the data collection edge and only communicating the processing results.	154
6.4	Underwater energy management plan with solar and tidal wave energy harvesting, along with the motion detection technology and DNN compression.	155
6.5	Example outputs of the modified U-NET segmentation model trained on 20 cloud-based distributed computers of the AWS SageMaker, and then deployed with Int8 parameter resolution on a Jetson Nano device.	158
6.5	(Continued.) Example outputs of the modified U-NET segmentation model trained on 20 cloud-based distributed computers of the AWS SageMaker, and then deployed with Int8 parameter resolution on a Jetson Nano device.	159

6.6 Image or video frame segmentation speed of an edge device with multiple compression levels is compared to a local computer that has been inquired by HTTP protocol. 161

7.1 Employing as many exogenous factors as desired to predict targeted endogenous variable with the proposed ensemble model of stacked DNNs. . . . 166

Chapter 1

Introduction

1.1 Background and Motivation

The Internet of Things (IoT) is a promising technology, which is revolutionizing every aspect of our lives. Nowadays, the domain of IoT influence has been expanded from smart homes and cities [19] to medical [20] and agricultural things [21, 22], and further to the Internet of Underwater Things (IoUT) [23].

The underwater section of IoT, i.e., IoUT has not attracted as much attention as it deserves and it is a rather unexplored research area. IoUT technology is intricately linked with intelligent vehicles, intelligent ports, positioning and navigation, maritime security, oceanic exploration, as well as with smart monitoring. The IoUT influence scale ranges from a scientific observatory, to a small port, and further to global trades. The heterogeneous network architecture of IoUT and its harsh environment imposes major challenges in terms of underwater communications, whilst relying on limited energy resources.

Additionally, the volume, velocity, and variety of data produced by sensors, hydrophones, and cameras in IoUT is enormous, giving rise to the concept of Big Marine Data (BMD), which has its own processing challenges. Hence, conventional data processing techniques will falter, and bespoke deep learning solutions have to be employed for automatically learning the specific marine data behaviour and features facilitating knowledge extraction and decision support.

In summary, motivations of this thesis are, first, to advance knowledge in the area of IoUT and big marine data processing by providing a detailed synthesis of these topics, and second, to improve and advance the state-of-the-art deep learning methods for achieving higher accuracy in marine data analytics. Besides, augmenting IoUT sensors with the advanced and intelligent edge devices is another motivation to make underwater video and image processing possible.

1.2 Research Problems

IoUT augmented with machine intelligence and marine data analytics is expected to revolutionize maritime studies in almost every technological area, and enhance our knowledge of this unknown environment. As stated in the previous section, IoUT has not attracted as much attention as it deserves. Our knowledge of the underwater environment is very limited for its harsh and dynamic spatiotemporal behaviours. Despite oceans cover more than 70% of the earth's surface and about 90% of international trades are through marine transportations, but 95% of sea area remains unexplored. Extending this knowledge toward a better understanding of oceans is essential and should be targeted in more scientific research works. Toward that end, this thesis addresses the following major research problems.

- Although the IoUT has many technical similarities with its ground-based counterpart (IoT) such as its structure and function, it has many technical differences arising from its different environment, communication/telecommunication limitations, computational difficulties, etc. To address these technical differences and devise new solutions for IoUT challenges, technical concepts in the field of IoUT need to be extensively reviewed. This review must include the underwater sensors and devices, as well as the heterogeneous nature of the IoUT communications. It is also required to review and discuss novel technological solutions for BMD handling, including software packages, open access databases, distributed and cloud-based BMD processing platforms, machine learning algorithms for underwater applications, and their hardware realizations. The large quantity of underwater sensory and imagery data sources also require modern marine data processing techniques. Consequently, the state-of-the-art technologies in underwater data processing using ML/DL needs to be reviewed as well.
- The problem of accurate sea surface temperature forecasting is studied next. Oceanic thermal energy has a great impact on global climate and worldwide ecosystems, as its anomalies have been shown to have a direct impact on atmospheric anomalies. The major parameter for measuring the thermal energy of oceans is the Sea Surface Temperature (SST). However, SST prediction is challenging due to the involvement of complex and nonlinear sea thermodynamic factors. To address this challenge, novel deep neural network for accurate SST forecasting need to be designed to incorporate the major thermodynamic factors into an accurate SST forecasting model.
- Nutrients and fine sediments are considered to be the primary land-based pollutants that significantly reduce ocean water quality. It has been shown that this can affect

the corals of Great Barrier Reef (GBR) in Australia. In this thesis, the problem of nitrogen as well as sediment distribution forecasting in GBR is addressed. This can provide decision support in reaching target water quality outcomes. To elaborate, we will consider the whole GBR as a frame and we will design two innovative next-frame prediction models to forecast nitrogen and sediment spatiotemporal distribution over the whole GBR. In this regard, we must

- Overcome the problem of insufficient ground truth training data in the wide GBR, and
- Devise a new solution to improve the forecasting ability of the state-of-the-art next-frame prediction models which produce blurred prediction frames.

Such a high-resolution model can produce accurate predictions by leveraging existing hindcasting simulation models.

- Wireless acoustic communication plays an important role in IoUT. However, this low-frequency communication technique has some limitations like the low transmission bandwidth, high signal attenuation, and high propagation delays. The last research problem addressed in this thesis is to employ efficient DNNs to enable high-resolution image and real-time video data processing at the IoUT underwater edge. In other words, the proposed edge processing technique is employed to improve processing power of image-based IoUT devices, which in turn can mitigate the inevitable drawbacks of narrow bandwidth underwater acoustic communication. Furthermore, using big marine data to train deep learning models is not efficient, or sometimes even possible, on local computers. We have investigated on how distributed learning in the cloud can help more efficiently process big data and train more accurate deep learning models.

1.3 Original Contributions

The original contributions presented throughout this thesis can be summarized as follows:

- A comprehensive review of the recent advances in IoUT and the extension of its influence both to coastal and open sea areas is provided in this thesis. This is the first IoUT review text that synthesizes research directions in a smooth progression, starting from IoUT data collection, followed by networking, through to marine data analytics and processing. This flow of logic evolves from the distinct research directions to their amalgamation into marine data analytics for IoUT. None of the existing IoUT survey papers are similar to ours in terms of their coverage. Putting

- the depth of each discussion in other papers aside, there are many topics that are only covered in our chapter on IoUT and marine data analytics. The review and discussions mentioned above are presented in the *IEEE Communications Surveys & Tutorials* [24].
- A new ensemble of stacked DNNs is designed to accurately forecast long-term sea surface temperatures in a variety of locations. The proposed lightweight and highly accurate new DNN structure leverages the correlation between SST datasets and air temperature at nearby isolatitude weather stations. It is demonstrated that our model significantly outperforms the state-of-the-art SST prediction algorithms. The results of this study are published in the *IEEE Geoscience and Remote Sensing Letters* [25].
 - Distribution of total-nitrogen and total-sediment in the oceans are highly temporally and spatially dynamic, and they are usually modelled using a complex set of Partial Differential Equations (PDEs). In this thesis, two novel DNNs are designed for the first time to predict the nitrogen and sediment distribution in the wide GBR. Both these models are inspired by the well-known Finite Element Analysis (FEA) to mesh the GBR into small elements, and to calculate a stiffness matrix for each element using the historical training dataset. To predict the total nitrogen and total sediment, we have devised two separate deep learning models which are fed with our FEA-inspired stiffness matrices. For nitrogen prediction, we have designed a DNN utilising CNN layers, while for our sediment prediction model, we have used transformers. Additionally, the Physics-Informed Neural Network (PINN) technique is successfully employed for training the proposed models using sparse marine measurement data. By innovatively integrating the FEA concept with the state-of-the-art deep neural networks, the proposed models were able to efficiently learn and predict long-term nitrogen and sediment dependencies. The results of the spatiotemporal nitrogen prediction in GBR are published in the *Environmental Modelling & Software* [26]. The spatiotemporal sediment forecasting results are published in the *Neural Networks* [27].
 - Finally, an energy-efficient platform has been proposed in this thesis to tackle the long-lasting barriers in underwater acoustic communications, i.e., the narrow bandwidth and high latency. To elaborate, a deep neural network for fish segmentation in underwater video frames is trained on distributed cloud computers and deployed on a GPU-powered edge device. The proposed DNN is 18% more accurate than training on local computers, and 4 times faster than an on-premises Hypertext Transfer

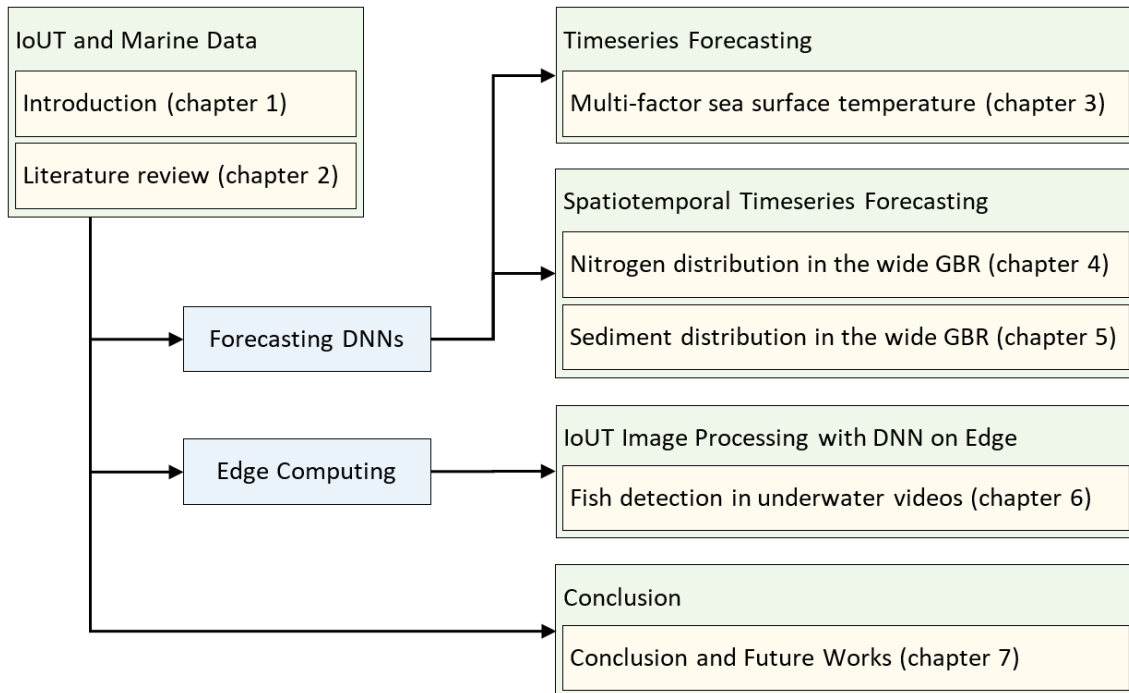


Figure 1.1: Outline of the thesis structure.

Protocol (HTTP) server. By compressing the DNN model for fish segmentation, 61% of power saving has been achieved compared to a FP32 DNN, which makes the underwater video processing feasible. An underwater acoustic network can easily transmit a few bytes of edge processing results. This is in contrast to the multi-MB raw image data submission in traditional wired communication for processing the image on-premises. The application report of this modern platform is published in *IEEE Access* [28].

1.4 Thesis Organization

As illustrated in Fig. 1.1, this thesis is organized in seven chapters to convey all the research objectives in a coherent way. The current Chapter 1 introduces the research background and motivation, research questions, and thesis objectives.

Chapter 2 conducts a literature review on IoUT and BMD analytics. At first, it presents an overview of the IoUT by associating its ecosystem to the concepts and methodologies defined for IoT. Then it discusses the challenges in the field of BMD and provides insights concerning oceanic sensor, image, and video data. In order to advance the knowledge and coordinate efforts in the field of BMD, sophisticated data analytic techniques and

methodologies are required, which have also been reviewed in this chapter. This chapter is published as [24].

Chapter 3 starts by explaining the importance of SST on affecting the global mean air temperature and influencing the marine ecosystem. This chapter follows by formulating the SST forecasting problem and its relationship with air temperature, as two ingredients of the proposed ensemble of two stacked DNNs. The performance of the stacked networks as well as the ensemble model are also discussed. This chapter is published as [25].

Chapters 4 and 5 respectively propose FE-DNN and FE-Transformer as novel solutions to the problem of nitrogen and sediment distribution prediction in the GBR. The concepts of total nitrogen and total sediment in the GBR will be defined, and challenges in their high-resolution spatiotemporal prediction will be also discussed. The physics-informed neural network technique for measured and simulated data fusion is described. The importance of this technique in sparse data applications will be discussed. The accuracy of the proposed distribution forecasting models will be evaluated, where a detailed investigation of both the computational complexity and the ablation properties of the models are also provided.

The combination of DNNs with edge computing will be studied in Chapter 6. This combination forms a modern platform to tackle the underwater communication barriers. The proposed platform can be employed in a wide range of marine applications. This chapter starts by designing a modified U-Net model for fish segmentation in real-life underwater images. The modified model will be accurately trained on a cloud-based distributed computer system, and it will be compressed and deployed on a GPU-enabled edge device for fast and efficient inferencing. Capabilities of the proposed edge computing platform in underwater data processing will be also investigated.

Finally, the thesis is concluded in Chapter 7, where the findings in other chapters are summarized. Besides, the recent advances in IoUT technologies are considered in this chapter to address the future works in the area of undersea data acquisition, data communication, BMD handling, and oceanic data processing.

Chapter 2

Literature Review

The scientific background for the research presented in this thesis is reviewed in this chapter. It firstly focuses on the Internet of Underwater Things (IoUT) and its challenges in the hostile underwater environment. Then the Big Marine Data (BMD) has been reviewed and the machine/deep learning approaches are introduced as robust solutions for marine data analytics. Finally, the challenges and future directions are described. This chapter is published in *IEEE Communications Surveys & Tutorials* as

- [24] M. Jahanbakht, W. Xiang, L. Hanzo, and M. Rahimi Azghadi, “Internet of Underwater Things and big marine data analytics—a comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 904–956, Jan. 2021.

2.1 Introduction

The Internet of Things (IoT) augmented with machine intelligence and big data analytics is expected to transform and revolutionize the way we live in almost every technological area. Broadly speaking, IoT can be defined as an infrastructure of the information society that connects equipment/devices (things) to the Internet and to one another. By this means, the IoT could connect devices in any place on earth to help us have better interaction with our living environment [29].

To date, the existing networks in terrestrial and urban areas have been the domain of influence for the IoT and have been researched extensively. This has made a fairly strong foundation for the industrial IoT developments, which are emerging with an astonishing pace at the time of writing [30]. However, the underwater section of IoT, i.e. IoUT has not attracted as much attention as it deserves and it is a rather unexplored research area. This is mainly because underwater applications are still in their infancy and the new era of scientific endeavor to better understand, control, and interact with the oceans and seas through underwater technologies is yet to flourish.

Although 44% of the earth's population lives within 150 km of the sea, 95% of sea area remains unexplored by the humankind [31]. Oceans cover more than 70% of the earth's surface and 90% of international trades are through nautical transportations [32]. Astonishingly, 12 people have spent 300 hours on the surface of the moon, while only 3 people have spent about 3 hours at 6 km depth of the ocean. In addition, about 90 million tons of salt-water fish are caught worldwide each year, and the coral reefs are estimated to provide food for almost 500 million people [33]. Hence, underwater research and development could have a significant impact on many aspects of human's life by establishing and rolling out the IoUT.

“On the surface of the ocean, men wage war and destroy each other; but down here, just a few feet beneath the surface, there is a calm and peace, unmolested by man.”

— Jules Verne

Although the IoUT has many technical similarities with its ground-based counterpart (IoT) such as its structure and function, it has many technical differences arising from its different communication/telecommunication environments, computational limitations, and constrained energy resources. To address these gaps between the IoT and IoUT, technical concepts in the field of IoUT will be extensively discussed. These include both the underwater communications [7,34,35], as well as the underwater sensors and devices [4].

By connecting an increasing number of devices and machines to the Internet, the IoT and IoUT ecosystems produce enormous amounts of data. This high volume of data is referred to in parlance as *big data*. Big data is currently being generated by various technological ecosystems and perhaps the most ubiquitous data types in today's world is the data produced throughout the IoT. This is also set to increase, since the number of Internet-connected devices is projected to increase from the current 30 billion to over 50 billion by 2020 [36].

In the age of sparse data production, analytical mathematics and statistical techniques, widely known as data mining, were employed to infer knowledge from data. However, in the current era of data proliferation, when the produced data volume in the last five years exceeds the whole amount of data generated before that, conventional data processing techniques will soon fall short [37,38]. These traditional big data handling methods relying on statistical descriptive, predictive, and prescriptive analytics usually suffer from the lack of generalization. That is, they cannot automatically learn the behavior and features in smaller datasets and use them in big data scenarios.

To address this significant problem, machine learning has risen as one of the practical solutions. ML has been created to facilitate an automatic approach to learning and extracting knowledge from data. This could revolutionize various aspects of our lives,

ranging from treating formidable diseases, to boosting the economies, to understanding the universe, to defense and military decisions [39–41]. As listed bellow, ML has also been used in a variety of sparse and big underwater data applications, including:

- *Evaluation and discoveries*: Examples of these include the evaluation of corals and their inhabitants [42], seabed analysis and mapping (photo-mosaicking) [43], object classification and discovery [44, 45], plant identification [46], the automatic recognition of fish [40, 47], lobster [48, 49], plankton [50], and other species, as well as tracking and direction finding [51].
- *Monitoring and management*: Examples of these include environmental monitoring (e.g. water quality and pollution) [52], fish farming [32], pipeline monitoring and corrosion investigation (e.g. in oil and gas industry) [5, 53], harbor security and military surveillance [31], navigation assistance [54], marine forecast and warning systems (e.g. tsunami, red-tide, flood) [55], and maritime geographic information systems [56].

ML as an indispensable tool in the IoUT, offers intelligent solutions for analyzing BMD, and thus it will be thoroughly investigated in this chapter.

This chapter is motivated by the fact that the IoUT, BMD, and machine/deep learning are salient topics emerging in the scientific literature. Nevertheless, there is no comprehensive survey to cover the joint applications of these three. In other words, many previous articles can be found in the literature that cover IoT [29, 36], big data [37], the joint aspects of IoT and big data [57], and even big data analytics in IoT [38, 58]. In a clear contrast, the amount of research published on the IoUT and BMD is very limited. Of these limited publications, some cover the IoUT [31, 32], while others cover BMD [34, 59]. However, to the best of our knowledge, this chapter is the first survey that provides a comprehensive overview of the IoUT and BMD analytics relying on the most recent radical machine/deep learning approaches. This makes the present text beneficial for data scientists, ML engineers, data analyst, big data engineers, and policy makers in the marine-related disciplines.

To boldly and explicitly illustrate the contributions of this thesis in the fields of IoUT and BMD analytics, Table 2.1 contrasts our unique contributions to other published treatises in the area. For all other articles in this table, a tick mark (✓) is granted, even if those dedicated only a few relevant sentences to the given subject. However, none of those surveys are similar to ours in terms of their coverage. Putting the depth of each discussion in other papers aside, there are many topics that are only covered in this chapter on IoUT and BMD analytics.

The synthesis of research directions in this chapter is handled in a smooth progression, starting from IoUT data collection, followed by networking, through to big data analytics

Table 2.1: The Contribution of This Thesis in the Field of IoUT and BMD Analytics, Compared to Other Previously Published Works

Discussed Subjects	Domingo [32], 2012	Kao <i>et al.</i> [31], 2017	Li <i>et al.</i> [34], 2018	Xu <i>et al.</i> [1], 2019	Jouhari <i>et al.</i> [6], 2019	Qiu <i>et al.</i> [60], 2020	Raj <i>et al.</i> [61], 2020	Khalil <i>et al.</i> [62], 2020	This Work
Internet of Underwater Things									
Sensors for Marine Data Collection	✓	✓		✓				✓	✓
Undersea Non-Destructive Testing									✓
Energy Consumption and Harvesting	✓	✓		✓	✓		✓	✓	✓
Communication Technologies	✓	✓	✓		✓	✓	✓	✓	✓
Wired and Wireless Channel Modeling		✓	✓		✓				✓
Edge Computing in IoUT						✓			✓
Big Marine Data									
Data Acquisition, Aggregation, and Fusion	✓					✓			✓
Sensor, Image, and Video Data Evaluation									✓
Open Access Databases				✓					✓
Distributed and Cloud-based Data Proc.						✓			✓
Applications (Monitoring, Tracking, etc.)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Machine Learning and Deep Learning for BMD Analytics									
Machine Learning Techniques Briefing									✓
Sensor, Image, and Video Data Cleaning									✓
Feature Extraction from BMD			✓						✓
Hardware Platforms in BMD Analytics									✓
ML for Sensor, Image, and Video BMD		✓	✓		✓				✓

and BMD processing. This flow of logic is illustrated in Fig. 2.1, which evolves from the distinct research directions to their amalgamation into big data analytics for IoUT. The four blue blocks labeled in this figure correspond to the remaining Sections 2.2, 2.3, 2.4, and 2.5. Additionally, any text or symbol inside these blocks represents a dedicated

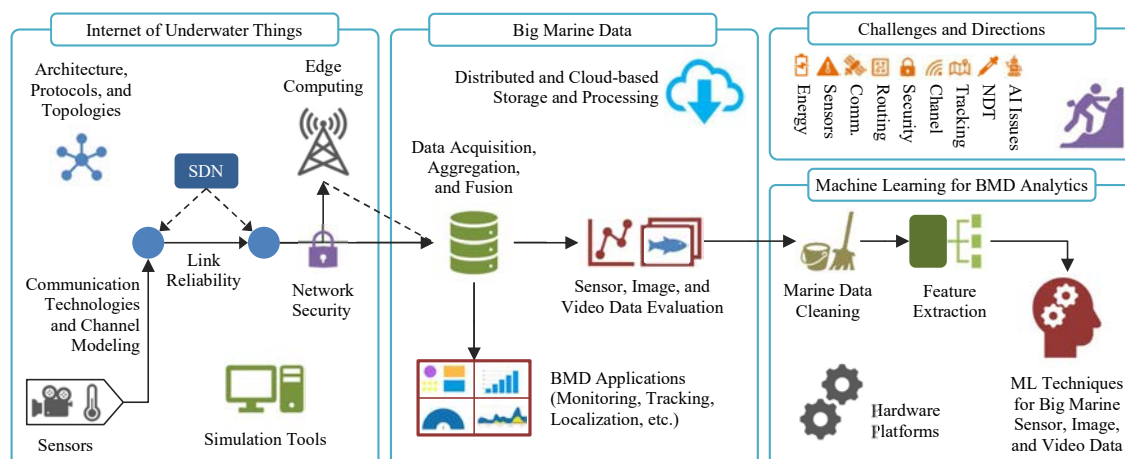


Figure 2.1: Qualitative relationship between the diverse system components of the IoUT and BMD analytics, starting from underwater sensors and ending up to ML solutions and future directions.

subsection. Accordingly, the reader will gradually become familiar with state-of-the-art tools and techniques, whilst gaining an insight into the challenges and opportunities in the broad areas of IoUT, BMD, and processing BMD in IoUT.

This treatise is organized as illustrated in Fig. 2.2. Section 2.2 presents an overview of the IoUT by associating its ecosystem to the concepts and methodologies defined for IoT. In Section 2.3, we discuss the usual challenges in the field of BMD and provide insights concerning oceanic sensors, image, and video data, which are widely available through several databases. In order to advance our knowledge and coordinate efforts in the field of BMD, sophisticated data analytic techniques and methodologies are required. As mentioned, one of the main approaches to meet this demand is to use ML techniques. Section 2.4 reviews several ML techniques conceived for automatic data leveraging from growing big marine databases. Finally, in Section 2.5, the challenges and opportunities in the emerging fields of IoUT, BMD, and underwater data processing are discussed, whilst offering further insights into the opportunities and potential solutions to the challenges. The chapter is concluded in Section 2.6.

2.2 Internet of Underwater Things

The concept of networks is broadly defined as a collection of independent machines, which exchange meaningful data through pre-arranged technologies (e.g. Ethernet, Wi-Fi, Bluetooth). Accordingly, the Internet (Worldwide Interconnected Networks) can be considered as a distributed network, or simply a network of networks. This network has an open

Chapter 2, Literature Review	
2.1. Introduction	
2.2. Internet of Underwater Things	
1	Underwater Sensors and Devices for Data Collection
2	IoUT Communications
3	Improving IoUT Link Reliability
4	IoUT Network Architecture
5	Underwater Channel Modeling
6	IoUT Network Topologies
7	Edge Computing in IoUT
8	Section Summary
2.3. Big Marine Data	
1	Marine Data Acquisition
2	Marine Data Classification
3	Distributed and Cloud-based BMD Processing
4	Marine Data Applications
5	Section Summary
2.4. Machine Learning for BMD Analytics	
1	Machine Learning Techniques
2	Deep Learning Frameworks and Libraries
3	Marine Data Cleaning for Machine Learning
4	Feature Extraction for Marine Data Analytics
5	Hardware Platforms for ML in IoUT
6	ML Techniques in BMD Applications
7	Section Summary
2.5. Challenges and Future Directions in IoUT and BMD	
1	Underwater Network Management System
2	Energy conservation and harvesting in IoUT devices
3	Development of Low-cost and Affordable Sensors
4	Large-scale IoUT Underwater Communications
5	Dynamic IoUT Signal Routing and Traffic Control
6	Deleterious Effects of Imprecise Channel Modeling
7	Sparse and High-maintenance Sensing Devices in IoUT
8	Poor Underwater Positioning and Navigation
9	Non-Destructive Testing in Underwater Applications
10	Lack of Strong Data Leveraging Tools
11	Training Deep Networks
12	Degraded Underwater Images
2.6. Conclusion	

Figure 2.2: Outline of Chapter 2 at a glance.

standard and constitutes a widely accessible ecosystem with lots of users and a variety of applications. Within the Internet, IoT is the largest sub-ecosystem, which connects devices in any place on earth to the World Wide Web.

Similar to the definition of IoT, IoUT may also be defined as worldwide interconnected networks of digitally identified underwater objects, which all obey the communication protocols of a pre-specified reference model such as TCP/IP or OSI [29]. Based on this definition, a detailed discussion of the IoUT objects (e.g. sensors) and underwater communications will be provided. Additionally, the family of IoUT network standard models and protocols will be surveyed.

2.2.1 Underwater Sensors and Devices for Data Collection

Our knowledge of the underwater environment is rather limited. This is a consequence of having underdeveloped monitoring technologies for this environment. In addition, due to the large operational areas, sea and coastal monitoring tend to suffer from sparse sensor deployment [63]. To overcome these shortcomings, low-energy sensors that are capable of working in the vast, hostile, and dynamic underwater conditions are required.

As mentioned in the introduction section, the perception achieved by understanding the data collected using sensors in oceanic areas are essential both to human life and to environmental sustainability. These sensors, for instance, can evaluate the impact of human activities on resources in marine ecosystems and also make us aware of the amount of pollution dumped into the sea [64]. At the time of writing small-scale Underwater Wireless Sensor Networks (UWSN) and hydrographic research vessels that contain a variety of marine sensors are deployed locally to assess environmental pollution and also to evaluate the seawater quality. However, the main disadvantage of these UWSNs is their small coverage area, which cannot cover the seas in scales of thousands of square kilometers [3].

Accordingly, it is important to connect all the existing sensor networks to the Internet, giving birth to the IoUT, in order to create an infrastructure for monitoring marine life on a global scale. This internationally accessible IoUT that measures essential chemical and physical parameters at sea, provides both historical and real-time measurements from myriads of marine locations worldwide. The collated oceanic sensor data will help experts predict future phenomena and also help policy makers ratify informed decisions [65].

Therefore, the IoUT infrastructure should consist of sensing objects and communication components in its underwater layers of the architectural model. These objects and components are known as nodes and sinks [31]. To elaborate [1],

- Underwater endpoint nodes are the end devices at the underwater side of the network, including various types of sensors, cameras, hydrophones, data storage microchips, actuators, acoustic tags, radio frequency tags, tag-readers, etc.
- Underwater mid-layer nodes are deployed above the underwater endpoint nodes, and are composed of data redistribution points, modems, gateways, repeaters, relays, etc.

- Sink nodes in IoUT terminology are the overwater nodes along with the land-side facilities, like buoys, exploration platforms, ships, satellites, onshore stations, etc.

Perhaps, the most important data collection components among all the different IoUT endpoint nodes are sensors, which not only collect data, but help activate other underwater components such as cameras, hydrophones, data storage micro-chips, and actuators. Some of the most popular environmental parameters in underwater applications are listed in Table 2.2. Typical industrial sensors that offer accurate measurement of these parameters are also listed. The BMD generated from the continuous operation of these sensors is transferred, stored and processed in the IoUT ecosystem.

In Sections 2.3 and 2.4, further discussions on marine sensors, as well as on sensor data storing and processing, will be provided. In addition, a number of IoUT-based observatory systems will be introduced. Before any sensor can be deployed in an underwater environment, efficient and reliable underwater data communications should be realized. This poses one of the greatest challenges for the pervasive sensor deployment in IoUT ecosystems due to the extremely low acoustic and electromagnetic channel capacities and high signal attenuations over long maritime distances, which affect reliable underwater data communications. The next section will discuss and survey data transportation as well as communication methods and protocols of the IoUT.

2.2.2 IoUT Communications

Today's marine vehicular communication systems utilize the Very High Frequency (VHF) automatic identification system to provide essential shipping information [3] (e.g. vessel name, position, speed, destination, etc.). In addition, high-speed satellite communications are available as an expensive alternative to existing VHF systems. However, none of these systems are capable of supporting long-range underwater applications, where the acoustic waves are the dominant communication media. Nonetheless, these acoustic carriers also suffer from high propagation delay, fading, narrow-bandwidth, and high-attenuation. Therefore, in underwater applications, it is sometimes inevitable to use a combination of different technologies (i.e. acoustic, electromagnetic, and optical) to overcome the communication challenges such as signal attenuation.

There is always a high level of signal attenuation, when passing through water. This attenuation affects every telecommunication technology in a different way. The signal attenuation is directly related to the main design constraints such as the maximum reliable data-rate and the maximum possible communications distance. In the following subsections, the signal attenuation is discussed in detail.

Table 2.2: IoUT Sensors to Measure Underwater Physical, Optical, Fluid, and Chemical Parameters [1–5]

Environmental Parameters	Typical Sensor Products	Brief Description
Temperature (°C)	<ul style="list-style-type: none"> • RBRCoda T by RBR • SBE series by Sea-Bird 	Digital underwater temperature readings in up to 10,000 m depth
Conductivity (S/m) and Salinity (ppt)	<ul style="list-style-type: none"> • TTurb by TriOS • 4319 and 4419 by Aanderaa • SBE 4 series by Sea-Bird 	Digital conductivity measurement in the range of 0 ~ 200 mS/cm and up to 6000 m depth rating
Depth (m) and Pressure (Bar) for Bathymetry	<ul style="list-style-type: none"> • PTM, PR36, and 2600 Series by Omni • 8000 series by Paroscientific 	Analog and digital sensors can measure water depth and pressure with an accuracy of $\pm 0.1\%$
Hydrophone (dBV/ μ Pa)	<ul style="list-style-type: none"> • Variety of products by Benthowave Instrument • icListen and icTalk series by Ocean Sonics 	Analog and digital 10 mHz~2 MHz active and passive hydrophones with wide beamwidth for up to 2000 m underwater
Turbidity and Visibility (NTU)	<ul style="list-style-type: none"> • WQ730 by xylem • 4112 by Aanderaa 	Analog and digital turbidity measurement in up to 6000 m depth
Optical Attenuation (Absorption) (m^{-1})	<ul style="list-style-type: none"> • FAS series by Sea-Bird • OLAS by Werne & Thiel 	Scanning of the light spectrum in up to 5000 m depth rating
Photosynthetic PAR ($\mu mol m^{-2} s^{-1}$)	<ul style="list-style-type: none"> • LI-192 by LI-COR • RBRCoda PAR by RBR 	Analog and digital sensors for PAR read in up to 7000 m depth
Water Flow and Current Velocity (m/s)	<ul style="list-style-type: none"> • Current meters by Valeport • Water flow meters by OTT • ISM series by HS Engineers 	Analog and digital current flow sensors with an accuracy of $\pm 0.5\%$ and up to 6000 m depth rating
Tide and Wave Elevation (m) and Direction (°)	<ul style="list-style-type: none"> • 4648, 5218, and 44xx series by Aanderaa • ACM-PLUS series by Falmouth Scientific 	Tide height, wave height, and wave direction read for applications like wave Fourier spectrum, tide pressure, etc.
Nutrients (e.g. NO ₂ , NO ₃ , PO ₄ , etc.) (ppm)	<ul style="list-style-type: none"> • AP Series by Aquaread • Ion-Selective Electrodes by Eureka Water Probes 	Nutrient sensors by solid-state, liquid membrane, or UV to measure nitrates, phosphates, etc.
Dissolved Oxygen (μ Mol/L)	<ul style="list-style-type: none"> • AP Series by Aquaread • Oxygen sensors by AMT 	Analog and digital sensor in up to 12,000 m depth rating
CO ₂ and pCO ₂ (ppm)	<ul style="list-style-type: none"> • C-sense by Turner Designs • pCO₂ series by SubCtech 	Analog and digital sensors in up to 6000 m depth rating

Electromagnetic signal attenuation

Electromagnetic waves and radio frequency signals do not propagate well underwater. This is mainly due to the high conductivity of seawater. The penetration depth of electromagnetic waves is inversely proportional both to the conductivity (σ [S/m]) and to the frequency (f [Hz]) [66]:

$$\delta \approx \frac{1}{\sqrt{\pi \mu f \sigma}}, \quad (2.1)$$

where μ is the water permeability and the penetration depth expressed in δ [m] is defined as the distance that an electromagnetic wave travels before becoming attenuated to e^{-1} of its initial amplitude. Based on this formula, only low frequency signals (with small channel capacity) can travel long distances in seawater, before they completely fade out. Therefore, the antenna size (L [m]) increases, as the frequency decreases [66]:

$$L \propto \frac{v}{f}, \quad (2.2)$$

where v is the speed of the electromagnetic wave in water that is almost equal to the speed of light in free space.

Acoustic signal attenuation

Due to the hostile electromagnetic underwater environment, most communications in the IoUT are based on acoustic links, which also suffer from a narrow frequency bandwidth. However, the attenuation of low frequency acoustic waves (α [dB/km]) is lower than that of electromagnetic waves [67]:

$$\alpha \approx F_1(f, pH) + F_2(f, T, S, z), \quad (2.3)$$

where f is the frequency in [kHz], pH is the water acidity (i.e. almost 6.0 ~ 8.5 for both freshwater and seawater), T is the water temperature in centigrade, S is the water salinity in parts-per-thousand (ppt), and z is depth in kilometer.

Optical signal attenuation

Another possible communication technology for underwater environments relies on an optical channel. However, similar to the previous pair of channel types, optical channels also suffer from signal attenuation.

Absorption and scattering are the two main causes of optical signal attenuation under water. Several previous studies have performed numerical simulations to estimate the attenuation [7, 68, 69]. These include solving the radiative transfer equation, which is

time-consuming and complex, but precise; or using simplified models (e.g. Monte Carlo), which are typically fast but imprecise [7].

One of the most successful examples of these simplified Monte Carlo-based approaches was proposed by Gabriel *et al.* [68], where the spectral beam attenuation coefficient is calculated as,

$$c(\lambda) = a(\lambda) + b(\lambda), \quad (2.4)$$

where $a(\lambda)$ and $b(\lambda)$ are the spectral absorption and spectral scattering coefficients, respectively, and λ [m] is the wavelength. Both of these coefficients constitute intrinsic optical properties and are calculated by simple volume integration over a solid angle (Ψ).

Alternatively, the relationship between the received light intensity ($L(t, r, \theta, \varphi)$) and the transmitted optical power ($S(t)$) expressed in spherical coordinates (r, θ, φ) can be defined by the well-known radiative transfer differential equation [70]. One of the most accurate solvers for this differential equation in underwater optical communications was proposed by Illi *et al.* [69], who have formulated their time-domain approach as,

$$\left[\frac{1}{v} \frac{\partial}{\partial t} + \vec{n} \cdot \nabla \right] L = -c(\lambda) L + S + \iint \text{VSF} \times L \sin(\varphi') d\theta' d\varphi', \quad (2.5)$$

where v is the speed of light, \vec{n} denotes the direction of propagation, and $c(\lambda)$ accounts for the same absorption and scattering coefficients of (2.4). The Volume Scattering Function (VSF) will be discussed in Section 2.3.2.

The improvement in solving (2.5) was achieved by enhancing the finite-difference method as well as by proposing a better approximation for the definite integral. Again, this is an accurate, but computationally expensive method of optical signal attenuation characterization.

The signal attenuation encountered by different telecommunication technologies discussed above directly affect the bit error rate and the overall underwater link reliability in the IoUT, which is discussed in the following section.

2.2.3 Improving IoUT Link Reliability

The underwater channel quality is significantly affected by several dynamic factors, including tidal-waves, pressure gradients, temperature gradients, floating sediments, and changes in water ingredients (chemical compounds). These channel dynamics affect the signal amplitude (distortion), frequency (dispersion), and speed (refraction) [71]. They also result in different delays, corresponding to delay jitter. Due to these challenges, all

underwater channels are considered as unreliable links, which requires us to define a reliability metric and then, use this metric for quantifying and optimizing the quality of service in our IoUT network [72].

Previous contributions in this field tend to assess the link reliability either by software-based or hardware-based metrics [72–75]. Hardware-based metrics tend to measure Signal-to-Noise Ratio (SNR) and signal-to-interference ratio as their reliability metrics [73]. The signal strength values can be directly read from the hardware transceiver. On the other hand, software-based metrics predominantly rely on comparing the overall end-to-end data delivery of the communication systems [72]. The channel or link reliability metric is defined as the ratio of bits as well as packets that are delivered successfully through the link. This metric can be evaluated either at the bit- or the packet-level. At the bit-level, the Bit Reception Ratio (BRR) is the reciprocal of the Bit Error Rate (BER):

$$\text{BER} = \frac{N_{\text{Erroneous Receptions}} [\text{bits}]}{N_{\text{Total Transmissions}} [\text{bits}]}, \quad (2.6)$$

$$\text{BRR} = 1 - \text{BER}. \quad (2.7)$$

By contrast, for a packet of m bytes (i.e. $8 \times m$ bits), successful reception of a packet means that all the m bytes were received correctly. Thus, the link reliability metric at the packet-level can be defined by the Packet Reception Ratio (PRR) as [72]:

$$\text{PRR} = (1 - \text{BER})^{8m}. \quad (2.8)$$

Another commonly used alternative to the BER of (2.6) for defining a link reliability metric is the Required Number of Packet transmissions (RNP), defined as [74]:

$$\text{RNP} = \frac{N_{\text{Total Transmissions}} [\text{packets}]}{N_{\text{Correct Receptions}} [\text{packets}]}. \quad (2.9)$$

In contrast to the BER that is measured at the receiver side, the RNP metric is designed to be measured at the transmitter side.

Another software-based link reliability metric, namely the Expected Transmission Count (ETX), is defined as the number of expected transmissions that a node requires for successful delivery of a packet [75], which can be directly calculated as the reciprocal of the PRR value of:

$$\text{ETX} = \frac{1}{\text{PRR}}. \quad (2.10)$$

Based on the band-limited nature of both the overwater and underwater telecommunication channels, the BER, RNP, and ETX values that have been respectively calculated in

(2.6), (2.9), and (2.10) are related to the originally transmitted data-rate, initially transmitted power, as well as to the distance between the consecutive transceiver pairs [66]. In this regard, reliable communication in the presence of random background noise, requires a certain minimum received power. This guarantees reliable data flow throughout the IoUT infrastructure for the ensuring big marine data processing.

The data-rate in bits-per-second (bps) and the transmission range, alongside the other characteristics of communication technologies are shown in Table 2.3. The simplified simulation plot of attenuation vs. transmission range in this table is calculated using the same conditions as in the studies discussed in Section 2.2.2. Here, the electromagnetic conductivity of the seawater and freshwater are considered to be 44,000 and $100 \mu \text{ S/cm}$, respectively [76]. The salinity values of the seawater and freshwater are also considered to be 35 and 0.4 ppt. The water depth only has a minor effect on the results and it is considered to be 1 km and 1 m for seawater and freshwater, respectively. The water temperature and its pH value are assumed to be 10°C and 7.7.

By comparing the details provided in Table 2.3, one can readily conclude that reliable underwater wireless communications are restricted to low data-rate acoustic waves for long distances or high data-rate optical rays for short distances. Short-distance and low data-rate electromagnetic waves are substantially outperformed by the other two technologies.

Despite the restrictive nature of the above-mentioned underwater communication technologies, innovative techniques can be developed to boost both the software-based and the hardware-based reliability metrics of underwater communication links. These techniques include *ad hoc* routing improvements and hop-count optimization, which are discussed in the following subsections.

Routing improvement

The specific choice of the link reliability metrics to be optimized has a substantial influence on the routing design of *ad hoc* networks [77]. Therefore, the design and implementation of a reliable network under the above-mentioned restrictive conditions of underwater channels requires an efficient data routing scheme [63]. One such scheme has been proposed by Rani *et al.* [18] for UWSN, which can be adapted to IoUT nodes as well. As shown in Fig. 2.3, the whole underwater wireless network in [18] is divided into multiple sub-regions or clusters. In every cluster, relay nodes cooperate with the local normal nodes to forward data to the cluster-heads. The cluster-heads of each cluster are responsible for routing and transmission of data to the next cluster-head situated in the upper sub-region, termed as *cluster coordinator*.

The scheme presented in [18] relies on a pair of efficient algorithms. The first algorithm

Table 2.3: Data-Rates and other Characteristics of Communication Technologies in Underwater Applications [6–8]

Communication Technology	Data Rate	Latency	Advantages	Simplified Simulation of Attenuation vs. Transmission Range
	Tx Range	Efficiency	Disadvantages	
	Attenuation	Main Channel Parameters		
Acoustic	~ kbps	High	<ul style="list-style-type: none"> • Proven and widely used • Long communication distances • Works in non-line-of-sight 	<p>Electromagnetic: 20 MHz Optical frequency: N/A Acoustic: 500 kHz</p>
	~ Kilometers	~ 100 bits/Joules	<ul style="list-style-type: none"> • Low data rate • Harmful on marine life • Large latencies 	
Electromagnetic	< 150 dB/km	<ul style="list-style-type: none"> • Temperature • Salinity • Pressure 	<ul style="list-style-type: none"> • Works in non-line-of-sight 	
	~ Mbps	Low	<ul style="list-style-type: none"> • Very limited range 	
	~ Tens of meters	N.A.		
Optical	< 100 dB/m	<ul style="list-style-type: none"> • Conductivity • Permittivity 		
	~ Gbps	Low	<ul style="list-style-type: none"> • Ultra-high data rate • Low cost 	
	~ Hundreds of meters	~ 30,000 bits/Joules	<ul style="list-style-type: none"> • Highly affected by environmental parameters • Requires a line-of-sight • Limited range 	
	< 15 dB/m	<ul style="list-style-type: none"> • Absorption • Scattering 		

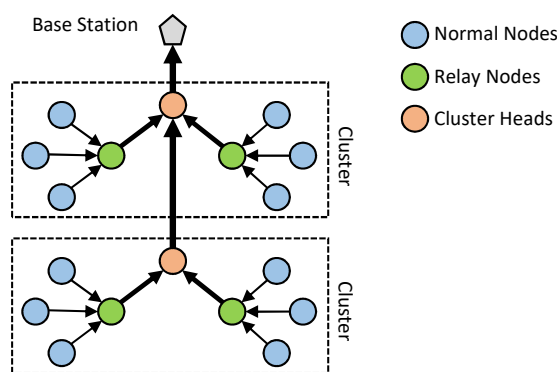


Figure 2.3: Reliable and energy-efficient multi-cluster network topology [18].

constitutes a location-free and energy-based policy that is devised to group individual nodes into cluster sets, aiming to increase SNR reliability metric. The second algorithm involves allowing cluster-heads and relay nodes to store data before routing and transmission. This helps the whole system to increase the RNP metric in (2.9), by avoiding retransmission of the same data packets. By implementing these ideas, the protocol tends to transmit fewer packets of data with a higher reliability and also carries out load balancing in packet routing throughout the network.

Another noteworthy data routing scheme was proposed by Tran-Dang *et al.* [78] for underwater acoustic sensor networks. Their work follows the same structure as [18] in Fig. 2.3. By contrast, their cooperative routing algorithm does not divide the UWSN into multiple clusters. Instead, every normal node independently selects its own relay node and cluster-head. This selection is based on the SNR link reliability metric of Section 2.2.3, on the physical distances represented by hop-count in Section 2.2.3, and the time of arrival, which is extracted from the timestamps of the packages. Moreover, this algorithm constantly monitors the environment in UWSN to estimate the SNR, hop-count, and time of arrival throughout the network. These parameters are frequently updated to adapt to the dynamic underwater environment. However, the simulation results reported in this chapter require experimental fieldwork for validation.

More discussions on the effects of network topology on the software-based link reliability metrics (e.g. BER) are provided in Section 2.2.5. In the next section, a hardware-based reliability metric (i.e. SNR) will be used for the evaluation and optimization of the hop-counts in IoUT.

Hop-count optimization

Another subcategory of methods capable of increasing the underwater link reliability is to optimize the hop-count, which is defined as the number of intermediate hardware devices (Modem, Gateway, Switch, Router, HUB, and Repeater) conveying the data between successive sources and destinations [79]. For example we could minimize the number of hops, provided that the transceiver is capable of reliably communicating over higher distances with the aid of higher transmit power and or more sophisticated receivers.

To characterize the effect of the number of hops on the system's reliability, we have to consider the fact that almost all of the wireless underwater endpoint and mid-layer nodes are battery-limited and the adequate operation of the entire network depends on the charge of their batteries. On the other hand, increasing the transmission distance, which is required for most underwater applications, decreases the effective bandwidth. This in turn increases the power consumption of delivering the payload at a minimum

SNR.

To address the challenges imposed by battery constrained underwater nodes on the link's SNR reliability metric, Li *et al.* [34] suggest reducing the hop-distance by deploying relay nodes along the underwater link to improve the overall transmission performance. This relay-aided transmission scheme is based on power-bandwidth-range dependency, and at the time of writing it offers an energy-efficient method supporting sustainable high data-rate delivery in underwater scenarios.

To overcome the challenges of wireless communications, we basically need to undertake an analytical or numerical approach, to computationally evaluate data transmission throughout the network. Taking this mathematical approach will consequently empower us to optimize our communication system in the favor of having more reliable IoUT infrastructure. However, analyzing and optimizing the physical layer of the network requires us to provide the simulation tool with an appropriate underwater physical channel model, which is the topic of the next section.

2.2.4 Underwater Channel Modeling

As discussed in the previous sections, the underwater physical layer exhibits different behavior in response to both different propagation modes and different channel types. For example, the signal attenuation was found out in Section 2.2.2 to be calculated differently for electromagnetic, acoustic, and optical carriers. The physical layer also exhibits different behavior in the case of wired and wireless channels. Regardless of the channel types and propagation technologies, we require to have an appropriate channel model to have a better insight on underwater data transmission. This proper model can then be used for predicting the performance of our communication system, designing the optimum underwater location of nodes, and decreasing the overall energy consumption of the system, before its actual deployment [80].

The channel models of wired and wireless networks vary with the choice of the communication technology. In other words, it is not possible to design an accurate channel model for universal employment in every application [71]. On the other hand, a feasible model has to undergo some degrees of simplifications, which is strongly correlated with the requirements of the problem itself [81]. As an example, we might simply neglect the water salinity in an acoustic channel model, while this simplification is not possible in underwater electromagnetic propagation.

Considering the aforementioned factors, channel models are somehow tailored to their own specific use cases. Accordingly, one usually needs to modify and combine the main features of multiple stand-alone models to come up with a dedicated solution to a specific

application. For instance, each of the following works considered a specific aspect of the acoustic wave propagation underwater (i.e. layered ocean water, particles in seawater, and the slope in seabed) and neglecting the rest, for simplification.

- It is very common in underwater channel modeling to consider a constant phase velocity throughout the entire medium (isovelocity). But in the contribution of Naderi *et al.* [82], the non-isovelocity acoustic wave propagation in shallow-water environments has been evaluated. They split ocean water into multiple layers of piece-wise linear sound speed profiles. Afterward, they use traditional sound wave propagation techniques to extract the time-variant channel transfer function. The complexity level of their solution can be adjusted by increasing the number of the linear layers. This could easily result in instability or inaccuracy in case of extra-thin or extra-thick layers, respectively.
- Relatively long wavelength of acoustic waves compared to the floating particles in ocean, makes them less vulnerable to backscatter and forward scatter phenomena. This is not the same for optical waves, as will be studied in Section 2.3.2. In a relevant study, Zhou *et al.* [81] have considered the impact of the scattering particles on the propagation characteristics of acoustic waves in underwater environments. They randomly distribute those particles on an assumptive rectangular cross section of ocean. Despite their interesting method, two main drawbacks exist in their statistical approach. First, their evaluation is only a 2D vertical cross section of seawater, neglecting realistic 3D sections. Second, they do not consider the stochastic size of the scattering particles.
- It is generally considered safe to assume the sea surface as a flat plane in underwater acoustic wave propagation, but this is not the case for the seafloor. In the work done by Naderi *et al.* [83], the up/down slopes in seabed have been taken into account. This important consideration is particularly essential in the case of shallow waters in the presence of coral reefs and plants, and cannot be neglected.

As mentioned, in order to have a desired customized and simplified underwater channel model, relevant existing models can be combined and unified. The majority of acoustic channel models (including all of the above-mentioned) follow the same mathematical approach. This approach relies on the superposition of a single Euler wave ray, over all the possible propagation paths in the time- or frequency-domain. This superposition relying on resolving the time-variant channel impulse responses can be written as [82],

$$h(t) = \sum_n c_n e^{-j(2\pi f_0 t + \theta_n)} \delta(t - \Delta t_n) , \quad (2.11)$$

$$\theta_n = \frac{2\pi f_0}{v_0} \Delta L_n, \quad (2.12)$$

where f_0 and v_0 denote the frequency and the acoustic wave speed in water, Δt_n and ΔL_n stand for the propagation delay and the difference in propagation path length, θ_n is the propagation phase shift, and $\delta(\cdot)$ is the Dirac delta function. Here in (2.11), c_n encapsulates both the initial gain and the subsequent losses by incorporating the concept of statistical random variables to model the stochastic environmental parameters (i.e. effects of scattering particles, motion-induced Doppler shifting, location uncertainty of nodes, changes in received power, dynamic seabed topology, and other natural variables). Selection of these environmental parameters for encapsulating in c_n differs from one use case to another, depending on the scale of these parameters' effect in every case [80].

The same concept of acoustic superposition in (2.11) can be used to model electromagnetic channels. But as discussed in Section 2.2.2, high-attenuating electromagnetic waves are not common in underwater data transmission and therefore, they are not evaluated here. In a clear contrast, using line-of-sight optical communications as well as fiber-optic channels are very popular in high data-rate underwater communications [7]. In this regard, the subject of optical fibers is out of the scope of this thesis and the reader is encouraged to follow it up from other resources [84]. Meanwhile, studying the line-of-sight optical channel properties in underwater applications will be conducted in Section 2.3.2.

As stated at the beginning of this section, the channel modeling is the backbone of every IoUT network simulation to evaluate its proper operation and to optimize its parameters [85–87]. One of the very basic parameters in IoUT networks that relies on channel modeling is the node location and it is referred to in parlance as *network topology*, which will be discussed in the next section.

2.2.5 IoUT Network Topologies

By utilizing TCP/IP as the reference model for the IoUT network architecture, almost all of the known network topologies can be used in gateway-enabled underwater applications [1, 88]. Two of these topologies (i.e. tree and mesh), which have high potential and are suitable for implementation in IoUT are shown in Fig. 2.4.

The frequency bandwidth is limited undersea and energy is hard to harvest. Therefore, advanced distributed topologies are rarely used in underwater applications, instead, the conventional tree and mesh topologies are dominant. The tree topology is typically used in small networks relying on one-way protocols [4, 18, 32, 88]. The tier (client-server) negotiation in this topology is based on a request and response process. On the other hand, the mesh topology tends to be the option of choice in sophisticated networks to meet the high traffic requirements, while using all available signal routes and frequency bands

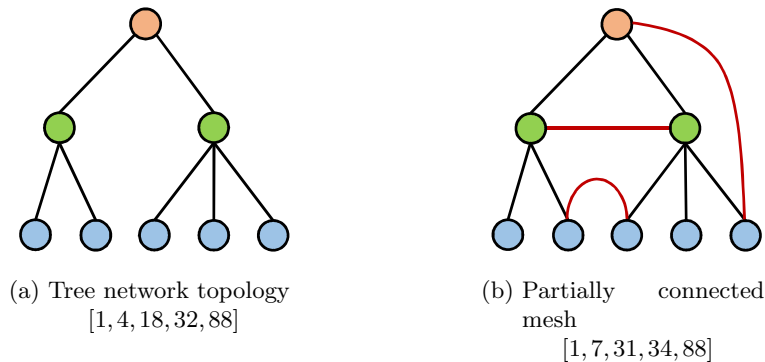


Figure 2.4: Reported network topologies in IoUT applications.

efficiently. This ensures that the limited channel capacity of underwater environments is efficiently exploited [7, 31, 34, 88].

It is worth noting that any signal route of Fig. 2.4 has its own reliability quantified in terms of its overall bit error rate in any typical IoUT network. This assists us in beneficially choosing a suitable network topology for any application, given its specific bit error rate target.

However, choosing a suitable network topology, designing a multi-layer communication network, and selecting an appropriate protocol for each layer of its architecture is not always a straightforward task. This job requires a tedious work to continuously design, test, debug, integrate, and deploy the newer versions of the network. It will get even more challenging in the harsh underwater conditions. To make this process a lot easier, some simulation tools are available, which can be used in underwater network design as well as layer-wise protocol testing.

All the approaches discussed above are introduced to enhance underwater communication and to address its challenges. However, if the devices and nodes in IoUT are capable of local computations at the edge of the network, the underwater network traffic will be significantly reduced and further improvements can be made to the IoUT ecosystem. The concept of these edge computing nodes and their beneficial effects on underwater data traffic will be discussed in the next section.

2.2.6 Edge Computing in IoUT


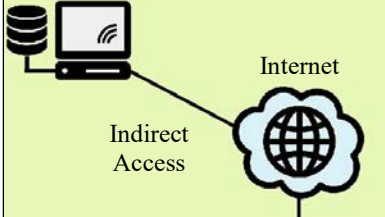

The concept of edge computing was originally introduced in IoT as an alternative to cloud computing. As the terminology suggests, in edge computing, the endpoint (edge) devices perform all or part of the required computations, so the need for data transfer and communication becomes less of a challenge. This sparse data transfer is ideal for the

IoUT ecosystem, which suffers from the hostile communication medium. Within IoUT, edge computing can be defined as a distributed and elastic computing paradigm, in which computing is predominantly carried out in the edge-devices such as, underwater endpoints, mid-layer nodes, and data acquisition tools. In the absence of edge computing capability, the processing should be performed on local computers, servers, or by the centralized clouds [89], all of which require shuttling data back and forth, hence rendering it unsuitable for IoUT [90]. In edge computing, devices have to expand their capabilities and in addition to data collection and communication, engage in data processing as well. This feature will shift the services from a single centralized point to numerous distributed nodes and closer to the physical world. The major advantages of using such a decentralized edge computing technology in the IoUT ecosystem are data-rate reduction, latency reduction, and prompt inner-network decisions making [90–92].

However, to turn these advantages into reality, edge computing has to tackle significant challenges. The energy resources of the edge-devices are limited. Furthermore, sustainable power cannot be readily delivered to the processing units of underwater applications. Even though edge computing has been the subject of extensive research in conventional IoT applications, it has its own specific challenges in IoUT. Below, we briefly review two of the IoT edge computing paradigms presented in [93] that are suitable for IoUT.

- *Mobile Cloud Computing (MCC)*: It is defined in IoT ecosystem, as the combination of cloud computing and mobile nodes to execute computational- as well as storage-heavy mobile applications (e.g. machine learning) in the cloud. MCC can offer rich computational resources to scarce-resource underwater applications. However, its relevance to IoUT is low because of the long propagation distance from the distributed BMD sources to the remote cloud servers [94], as well as the narrow underwater bandwidth and limited access to energy. Another drawback of MCC is its service accessibility, which is via Internet connection only. This is in contrast to other edge computing paradigms that can offer a direct access [93].
- *Mobile Edge Computing (MEC)*: The acronym MEC represents mobile edge computing and multi-access edge computing, with the latter one being more inclusive. According to the descriptions offered by the European Telecommunications Standards Institute (ETSI), the MEC technique provides cloud computing capabilities at the edge of the network, with close proximity to the end-users (i.e. underwater endpoint nodes). In IoUT, the edge of network represents both the land stations at the edge (e.g. cellular towers, data centers, Wi-Fi routers, etc.) [95] as well as the overwater sink nodes (i.e. floating buoys, floating vehicles, exploration platforms, etc.). These edge gateways will carry out some preliminary and short-term tasks,

Table 2.4: Location of The Processing Unit in Different IoUT Edge Computing Paradigms

Edge Paradigms	Edge Processing Locations 	
Mobile Cloud Computing (MCC)		
Mobile Edge Computing (MEC)		

before handing their results to the cloud servers for more sophisticated and resource-intensive analysis [96]. MEC techniques invoked for the IoUT can offer low latency and reliability. Their other advantages in IoUT include their support of wired communications as well as their facilitated direct access (i.e. no Internet connection is required) [93].

Considering the above, MCC and MEC are two of the IoT edge computation techniques that are practically realizable in IoUT, as shown in Table 2.4. Among them, MEC requires lower bandwidth, lower network traffic, low-latency, and low-power underwater operation, and more reliable access to underwater sensors and cameras in the IoUT [91]. Hence, MEC networks constitute the most attractive edge computing paradigm in IoUT applications [91, 92]. They can perform computations in the sink or mid-layer edge-devices that are placed or have access to above water, where for example solar energy may be harvested.

2.2.7 Section Summary

In this section, we studied the essential topics involved in developing the IoUT. We discussed various sensors in the IoUT and learned about their important features and roles in data collection. We then pointed out the major challenges in the IoUT domain and learned that electromagnetic, acoustic and optical signal attenuation significantly affect the IoUT communication and its link reliability. We also learned how to measure and improve the

link reliability. Next, we discussed the IoUT architectural model and its layer-wised protocol stack, and learned how the TCP/IP model can be adapted to the IoUT architecture. In order to be able to gain better insight into the network performance and to predict its reliability, network modeling was discussed. The lessons learned were that various channel models result in different levels of complexity and they should be carefully selected, based on the application concerned.

The IoUT network topology was the next topic we covered, where we learned that the tree and partially connected mesh are the most popular topologies in IoUT applications. We also discussed the underwater network simulation tools and how they can be used to facilitate the implementation of the defined protocol stack. We provided a list of these simulation tools, which can be helpful to the community when designing and analyzing underwater communication networks and protocols. Network security in IoUT was another salient topic discussed. We learned that there is a paucity of literature on this important issue, even though it is essential in critical scenarios such as harbor security. Another important topic studied in this section was software-defined networks in IoUT. We learned how it can help with the management of IoUT networks. Finally, we discussed edge computing in IoUT and learned that it may be even more important for IoUT devices to have edge computing capability than for their IoT counterparts, mainly due to the more challenging communications in underwater scenarios.

The IoUT discussed in this section generates a vast amount of data, which may be referred to in parlance as *big marine data*. In the following section, we discuss big marine data and cover its associated data sources, data collection tools, and data processing methods in the oceanic and underwater domain.

2.3 Big Marine Data

This section is dedicated to underwater data types, which are envisioned to have the lion-share in the IoUT data transactions of the near future. We will also cover data acquisition tools, localization as well as tracking, and will introduce some ready-to-use big ocean sensory, imagery, video, and geographic databases, which are useful to researchers and practitioners in the IoUT Domain.

BMD can be succinctly defined as the vast amount of heterogeneous data collected from marine fields. The main characteristics of BMD are temporally long and spatially vast coverage, diverse nature of the data sources (e.g. sensors, cameras, tags, aerial remote sensing), and multi-disciplinary data types (i.e. physical, chemistry, biological, environmental, economical, etc.) [59].

The BMD system components and processing stages are as follows [57]:

1. *Acquisition*: Involves collecting raw data.
2. *Transportation and security*: Requires the data to be encrypted and transferred across different communication media to its target storage, while considering its reliability and security.
3. *Storage and privacy*: Deals with policies around data storage (legal concerns and users privacy) and its archival requirements (file formats, retention lifecycle, and replication).
4. *Special-purpose processing*: For complex datasets, bespoke software packages are required for searching, pre-processing by filtering and cleaning, recognition and labeling, post-processing and visualization of results, and updating.
5. *Exploitation and leveraging*: Ensures that users gain benefit in terms of increased revenue from their data. Some benefits of underwater data exploitation are that of monitoring the water's vital cleanliness, help new businesses to grow, support experts by providing informative bespoke data, track worldwide maritime transportations, and protect the environment.

From the above five BMD system components, *transportation and security* was already discussed in Sections 2.2. The third component, i.e. *storage and privacy* covers legal policies and data handling issues, which are beyond the scope of this thesis. Below, we review the *data acquisition* stage and later in sections 2.3.3 and 2.3.4 we will discuss some aspects of *data leveraging* stage in more details. Additionally, discussions on *data processing* will be provided in Section 2.4.

2.3.1 Marine Data Acquisition






Data acquisition, which is the first component of any BMD solution can be discussed in three different stages including data gathering, data aggregation, and data fusion.

Data gathering

Data gathering in IoUT can be performed using a variety of tools, some of which are listed in Table 2.5. The data acquisition tools in this table are divided into vehicles and primary data sources. Any vehicle in this table can be considered as a data acquisition tool, if and only if it is equipped with one or more primary data sources and tools (e.g. mounted cameras, hydrophones, sensors).

As we will discuss in more details in the forthcoming sections, BMD processing and data leveraging (as the 4th and 5th BMD system components) are not possible without

Table 2.5: List of Data Acquisition Tools in Underwater Applications with the Percentage of Use of Each Item in Related ML Articles

Tools	Share in ML	Types	Used in ML Articles	
			Before	After 2014
Vehicles	50%	Remotely Operated Vehicles <ul style="list-style-type: none"> • Free Swimming • Bottom Crawling • Structurally Reliant • Towed System 		62%
	45%	Autonomous Underwater Vehicles <ul style="list-style-type: none"> • Underwater AUVs • Underwater Gliders • Autonomous Submersibles 		54%
	5%	Human-Occupied Vehicles <ul style="list-style-type: none"> • Submersibles • Atmospheric Diving Suits 		88%
Primary Data Sources	65%	Sensors <ul style="list-style-type: none"> • Separately Listed in Table 2.2 		50%
	35%	Cameras <ul style="list-style-type: none"> • Fixed Cameras • Baited Remote Videos • Diver Operated Videos • Drifting Underwater Camera 		57%

automatic ML approaches. Therefore, a survey was conducted to extract the contribution of data acquisition tools in state-of-the-art ML articles. The survey result is shown in the second and fourth columns of Table 2.5. In the second column, the inter-category contribution of each item is compared to the others. In the vehicular category, the unmanned ROVs and AUVs together were used in about 95% of the relevant ML publications.

In the fourth column of Table 2.5, an item-specific evaluation is performed to quantify the share of each item both in the old and new ML publications (before and after 2014). As can be seen in the table, more than 50% of all underwater research in the area of ML have been carried out after 2014. There is only one exception for human-occupied vehicles. Perhaps, this is due to the recent wide adoption of automated as well as of remote methods and owing to the reduction in academic usage of costly manned vehicles.

Data aggregation

Data aggregation is a statistical data processing stage before higher level calculations and/or before transmission over a band-limited communication channel [97]. The level of raw data summarization here is to create another set of shortened raw data, by implementing mathematical techniques, such as down-sampling, linear regression, etc.

Data aggregation can be carried out in UWSNs, for example to minimize the bandwidth utilization. It can also reduce the energy consumption in the network-level, by striking a trade-off between low data processing power and high data communication energy. However, using this process in underwater applications may impose some unwanted drawbacks, like [98]:

- Increasing the energy consumption and processing requirements in the node-level;
- Increasing the overall network latency;
- Compromising data accuracy by shortening its volume.

To address these challenges, LEACH, PEGASIS, HEED, and APTEEN, which are some well-known aggregation protocols in wireless sensor networks can be used in UWSNs subject to modest adjustments [99–101]. Additionally, heavy computations and high energy demands in data aggregation processes can be mitigated by using edge computing paradigms, as described in Section 2.2.6.

Data fusion

The BMD gathered and aggregated may be stored in distinct subsystems or in separate databases. Data fusion is defined as the combination of relevant data from different data sources into an integrated dataset, with the objective of searching for more consistent and more accurate information than that provided by any individual database [102].

Data fusion systems in IoUT mainly rely on edge computing devices as well as on cloud computing servers to handle large amounts of heterogeneous BMD (i.e. sensors, audios, videos, commands, etc.) [60]. Three main questions are raised when designing a successful data fusion system in marine applications. The first is the location of data fusing operation, the second is the level of abstraction in the fusion system, and the third is the level of overlapping of the original data.

Considering the first question, data fusion can be performed either in a centralized node or in a distributed network [102]. In a centralized scheme, the data fusing node is usually located on overwater edge devices or in an inland facility with good access to energy resources [60]. The distributed data fusion will be hierarchically conducted over the entire

system, where every cluster-head fuses data from its own cluster nodes, passing them to the next cluster-head. Distributed BMD processing will be discussed in more details in Section 2.3.3. Considering the second question about abstraction level, the answers offered by [103] are:

- *Low level (i.e. sensor fusion)*: If the inputs to our fusion system is the raw data directly gathered from the sensors, cameras, etc.;
- *Medium level (i.e. feature fusion)*: If the inputs to our fusion system is the output of feature extracting blocks, as will be discussed in Section 2.4.4;
- *High level (i.e. decision fusion)*: If the inputs to our fusion system is the output of classification or clustering blocks, as will be discussed in Section 2.4.1.

Low level data fusion is not recommended in BMD, as a result of restricted underwater resources. The only exception is in edge computing, where the raw data is fused together on the edge device, right before extracting their features. On the other hand, medium and high level data fusion can help us with building more sophisticated models or with finding more complex solutions [104].

Considering the third question on the overlapping level, the lessons of [103] are:

- *Redundant*: If we have multiple datasets with the same data type, related to the same subject, in the same time interval, by different data acquisition tools or in different sampling time spots;
- *Complementary*: If we have multiple datasets of the same data type, related to the same subject, from different angles or in different time intervals;
- *Cooperative*: If we have multiple datasets of different data types, related to the same subject.

Fusing redundant or complementary data can increase our confidence over the original datasets. However, based on the expensive data transmission in UWSNs, neither redundant nor complementary levels of overlapping are recommended. In contrast, cooperative data fusion (e.g. couple of sensed parameters from Table 2.2 or associated audio and video data) can obviously increase our knowledge of the subject.

2.3.2 Marine Data Classification

Based on our findings in previous section, which has also been reflected in Table 2.5, cameras and sensors are considered as primary underwater data acquisition tools. Most

maritime industrial, research, or observatory project might integrate these data acquisition tools into a wireless or wired sensor network. The sensor networks that are connected to the Internet, all together play a predominant role in the overall IoUT infrastructure. The data collected in the UWSNs within the IoUT may be classified according to its dimension as discussed below.

One-dimensional time-series marine data

As described earlier in Sections 2.2.1, a sensor is considered as an endpoint node in the application layer, which offers time-series data reception through the Internet according to the end-user preferences. For instance, a biologist might be interested in the water temperature, while an environmentalist might be interested in the water quality.

Conventional marine sensors and marine nano-sensors (as listed in Table 2.2) measure a variety of oceanic signals and processes within a specific duration. Marine nano-sensors are popular owing to their low power consumption [105, 106]. The sequential data provided by sensors and nano-sensors are used in monitoring and surveillance applications to provide long-term and large-scale perception of the environment and to tackle unwanted environmental changes [107].

Implementation of these sensor nodes in small-scale research-based IoUT structures are repeatedly reported in literature. Some of these contributions are discussed below:

- A low-complexity VHF-based IoUT ecosystem is proposed by Al-Zaidi *et al.* [3] for marine data acquisition based on storage devices in the cloud. The proposed structure is equipped with temperature, depth, and wind sensors to produce a near real-time system.
- A network of sensor nodes, based on the IEEE 21451 standard is constructed by Adamo *et al.* [64] for continuous monitoring of the seawater quality. This system is devised as an IoUT network for making strategic decisions concerning a range of environmental issues.
- A low-cost technology for maritime environmental sensing is described by Wright *et al.* [108]. The technology relies on the IoUT for measuring parameters such as the optical properties of water, ocean temperature, and wave dynamics.
- The Great Barrier Reef of Australia is the largest coral system on planet Earth, spanning a distance of over 2300 km. Palaniswami *et al.* [109] constructed an UWSN to capture data from temperature, pressure, and humidity sensors in an IoUT platform, for monitoring the complex ecosystem of the Heron Island in the southern

Great Barrier Reef. The big data of sensory time-series collected for their study, has been then analyzed by ML algorithms to detect underwater anomalies. Therefore, [109] is a notable application of big data analytics in IoUT with prime objective of predicting severe tropical cyclones.

In addition to these small-scale research units, a range of other observatory stations have been established by institutes and organizations, to provide both research-oriented and industrial access to the underwater sensory data. These observatory stations are located all around the planet and they provide a reliable access to BMD through the IoUT.

Here we provide a list of major observatory stations in Table 2.6. The observatories are sorted in an ascending order from the smallest to the largest observatory coverage area. All the entries in this table are open access and indicate active projects with sustainable funding. This guarantees continuous data preparation and update. They are all accessible through a web-form or dedicated software.

There are some other observatory projects, which are not included in Table 2.6, because they are no longer supported and their databases have become obsolete. For instance, the [POSEIDON system](#) in Greece (2008 to 2012) is no longer actively operating, but its atmospheric and marine data are still downloadable through the web. Another example is the [JGOFS](#) (Joint Global Ocean Flux Study) project which was funded by the international science council during 1987 to 2003. JGOFS was an international program with participants from more than 20 nations. The rich multi-disciplinary data of this project is also still available to download.

The data provided by any one of the observatory systems in this section, can be affected by environmental noise, outlier records, misread values, and missing quantities. To deal with these measurement errors, we require a series of techniques, which will be provided in Section 2.4.3.

Two-dimensional underwater image data

Some observatory stations introduced under Table 2.6, are equipped with cameras to provide ready-to-use 2D image data. In addition to those live IoUT images and videos, there exist a variety of other still-image databases. These new databases are listed in Table 2.7, and they are eminently suitable for supervised ML applications (as it will be discussed in Section 2.4.1), as a benefit of their additional expert labels and annotations. These databases are accessible through a web-form or a dedicated software, or even in the form of a downloadable dataset. However, only a few of them are active projects with continuous updates.

Table 2.6: Open Access and Active Observatories Worldwide; Providing Up-to-date Primary Data

Project	Funder	Description
OBSEA	University of UPC	A seafloor observatory in a fishing protected costal area in Spain , which delivers live video , acoustic , and sensor data.
WHOI	Non-profit Organization	An organization in Massachusetts , which owns some observatory stations (like MVCO) and provides sensed data.
MARS	MBARI Institution	Cabled observatory in 891 m below the Monterey Bay in California provides videos , Acoustics , and sensor data.
LoVe	Statoil	This station provides camera , acoustics , and sensors in the cold-water corals in Norway [110].
NeXOS	European Commission	Miscellaneous sensor data from 3 sites around Italy and Spain are gathered [52].
MARNET	BSH [©]	Sensor data for the North sea and Baltic sea of Germany .
CCO	DEFRA Department	The network of six regional sensory data monitoring sites around UK .
IOOS	NOAA	Integrates existing networks of instruments in the western hemisphere (e.g. OOI , NERACOOS , MARACOOS) and provides videos and sensed data from regions around The USA .
AODN and AIMS	Australian Government	Australia 's marine territory is the third largest on Earth. AODN and AIMS portals provide access to Australian marine and climate sensed data.
ONC	University of Victoria	Delivers cabled observatories and sensors to monitor the west and east coasts of Canada.
EMSO	European Union	Consists of 8 observatories and 3 test sites around Europe , with biological, chemical, and physical sensors .
EMODNet	European Union	This European data network includes broad disciplinary themes of bathymetry , geology , physics , chemistry , biology , seafloor habitats , and human activities .
JCOMM in GOOS	UNESCO	An Intergovernmental sensor network that integrates many famous observatory programs worldwide, including ARGO , DBCP , OceanSITES , SOT , GO-SHIP , and GLOSS .

The multidisciplinary data provided by these image databases can be used in different branches of science. For example, while an ecologist may be interested in counting and

Table 2.7: Open Access Datasets of Still-images, Taken from Underwater Organisms

Project Name	Description
Inst. for Marine and Antarctic Studies (IMAS)	This Australian institute hosts three major image, taxonomy, and atlas database projects including Reef Life Survey (RLS), Temperate Reef Base (TRB), and Zooplankton .
Sea Life Base and Fish Base	FishBase is a project of FIN [©] (Philippines) with around 59,000 images of fish. SeaLifeBase is a joint project of the University of British Columbia (Canada) and The FIN [©] (Philippines), which includes 12,500 images of different marine species.
World Register of Marine Species (WoRMS)	Provides an authoritative list of valid and vernacular names for 250,000 global marine organisms. This Belgium -funded infrastructure contains extra information such as 32,000 images , literature, biogeographic data, and parent taxon. It is generally used by scientists for classification of marine species.
Fish 4 Knowledge	Provides three different datasets with annotated ground-truth for fish recognition (30,000 images of 23 species), fish trajectory (9000 trajectory of 23 species), and benchmark for complex backgrounds (14 videos of 7 backgrounds). It also offers access to live video feeds from 10 underwater cameras in reefs of Taiwan .
Encyclopedia of Life (EOL)	Provides 4,000,000 Images of all life-forms on Earth (e.g. animals, plants, bacteria), including 15,000 images and videos from marine environment. It is financially supported by institutions from Australia , The UK , The USA , Mexico , and Egypt .
Shape Queries Using Image Databases (SQUID)	Includes 1100 images of marine creatures in a smooth background (one creature in each picture). The ground-truth coordination of boundaries of animal's body in all images is annotated (256 to 1,653 points for each creature). This database is located in Tje UK and will not be updated anymore.
RSMAS, EILAT, EILAT II, and SDMRI [42, 111]	These independent datasets contain 766, 1123, 303, and 100 images of corals, which have been categorized by experts into 14, 8, 5, and 20 different classes of coral genera and non-coral, respectively.
Moorea Labeled Corals (MLC)	This is a subset of the MCR-LTER dataset and has 2055 coral reef images with almost 200 human expert point-annotations over each one of them. These images are taken from the island of Moorea in French Polynesia and contain 5 coral and 4 non-coral distinct classes. They also contain pallets with known reference colors; making them suitable for image correction and color restoration.

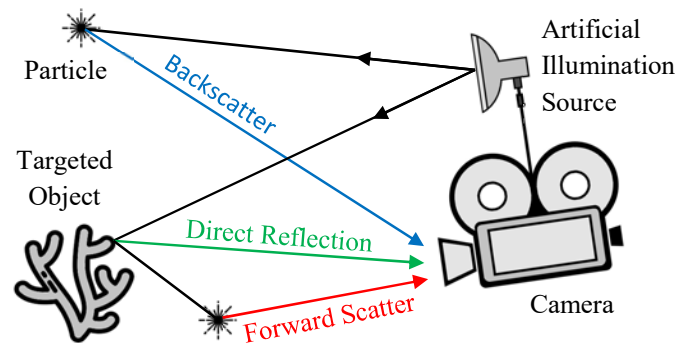


Figure 2.5: Schematic view of three different optical rays, which are reflected back to the camera, in active underwater imaging.

tracking of individual inhabitants, a data scientist or computer programmer might need these data to train a ML algorithm and to verify its performance.

When considering the usage of these imagery data, one should be aware of the lighting conditions in the underwater environments. This can be significantly different from over-water photography and it is expected to affect the images taken undersea. To understand the nature of light in undersea imaging systems, we have to discriminate passive and active images, i.e. whether the imaging equipment creates its own light or not. Although the physics of optics are the same in both cases, different sets of parameters have to be considered to provide a better understanding of the context of underwater optical imaging.

Active imaging relies on the explicit usage of artificial light in the process of underwater imaging. This type of photography benefits from a substantial improvement in image quality, especially when the light is appropriately controlled by an optimized hardware configuration. However, active imaging suffers from high underwater energy loss (especially in case of long-term illumination), reduced portability, and unpleasant inhomogeneous intensity and color of the final picture [112].

As depicted in Fig. 2.5, undersea active imaging always encounters 3 sources of light rays into the camera. Direct reflection from the target object is the desired signal, while the two other reflections, termed as *backscatter* (which has not interacted with the target) and *forward scatter* (or blur component) are both undesired.

Backscatter or Volume Scatter: In active imaging scenarios back scattered light is defined as the one which has never interacted with the target object and usually appears as bright points in the output image. To accurately calculate the magnitude of light that is backscattered to the camera, we have to calculate or measure the light intensity ($I[\text{W}/\text{m}^2]$) first. Light intensity (or irradiance) is the power received by an illuminated surface perpendicular to the direction of propagation, per unit area. This parameter has

to be evaluated for the specific volume that the camera is viewing. One of the popular methods to achieve this goal is proposed in [70], which discretizes the entire volume to small cubic cells and then calculates the Volume Scattering Function (VSF)¹ parameter for each cubic cell.

The VSF number of each unit cell is then used as a weighting function in the next step. Finally, the method of [70] considers the VSF, the light intensity I , and the angle between the incident and the reflected light for numerically estimating the magnitude of light that is backscattered to the camera. Again, this is an efficient and accurate algorithm of estimating the magnitude of the backscattered light. However, it can be used as a fundamental step of any image enhancement procedure, especially in underwater applications, where the presence of backscattered light is indeed a significant challenge.

Forward Scatter or Blur: Forward scattered light is defined as a light beam, which interacts with the target and then it is indirectly reflected back to the camera. To approximate the output image of $E_{\text{blur}}(x, y)$ in a scattering environment, a common method is to convolve the original image $E_o(x, y)$ with the Point Spread Function (PSF(r)) as [70]:

$$E_{\text{blur}} = E_o \otimes \text{PSF}(r), \quad (2.13)$$

where r is the distance from the camera to the object and \otimes is the convolution operator.

PSF represents the spatial impulse response of a scattering environment between a light source and the point of observation (e.g. camera), as already shown in Fig. 2.5. It is used in Fourier optics to calculate the output image of a linear imaging system. The formulation of PSF is simplified to [70]:

$$\text{PSF}(r) = \left[(e^{-Gr} - e^{-cr}) F^{-1} \left\{ e^{-Brf} \right\} \right], \quad (2.14)$$

where c is defined in (2.4), G is an empirical constant ($|G| \leq c$), B is an empirical damping factor, and $F^{-1}(\cdot)$ is the inverse Fourier transform.

Both e^{-Gr} and e^{-cr} in (2.14) represent the forward scattering amplitude attenuation, while e^{-Brf} stands for the frequency-dependent damping. The use of the convolution operator in (2.13) indicates that this equation is only valid in linear optics, which is the case in the majority of underwater light-based experiments [70]. Although PSF in (2.13) is used for the forward scattering formulation, it can be employed to calculate the VSF in backscattering as well. The PSF formulation as well as its relationship to the VSF has been studied in great detail in [70].

The forward- and back-scattering process encountered in seawater disperse the light

¹VSF is an inherent optical property of water, which does not vary with the incident light field. A number of experimental methods are introduced in [70] to measure the VSF.

beam, hence resulting in blurred underwater images. Observe from (2.14) that the image gets more blurred, when the distance increases. One can exploit this equation to mitigate the blurring problem with the aid of image deblurring methods, which will be covered in Section 2.4.3.

By contrast to active imaging, in passive imaging, i.e. where no artificial light is generated by the image capturer, even though the power consumption will be significantly lower, other concerns may be present. These concerns include limited visibility, contrast, and color distortion.

Visibility: The first concern in passive underwater imaging is always the visibility. In clear sunlit water, ambient sunlight provides a clear vision in relatively shallow water. In ordinary line-of-sight underwater applications, the distance of visibility can be modeled as,

$$d_{\text{Visible}} \approx \frac{5}{c - K(\theta, \varphi, z) \cos(\theta)}, \quad (2.15)$$

where K is the diffuse attenuation coefficient, which is an apparent optical parameter, while θ , φ , and z are the relative spherical coordinates of the subject of interest [113]. Based on this formula, underwater visibility decreases rapidly after a few meters. The only known remedy for this physical constraint in deep-waters is to use an artificial light source in active imaging [112].

Contrast: Contrast is defined as the color or gray-level difference between adjacent areas in the presence of light attenuation, optical noise, and vision blurring. If we consider a target object (O) against its background (B), the contrast of the object and its surrounding areas can be formulated as [114],

$$C_{OB} \triangleq \frac{L_O(\theta, \varphi, z) - L_B(\theta, \varphi, z)}{L_B(\theta, \varphi, z)}, \quad (2.16)$$

where L_O is the light level (radiance) of the object and L_B is that of background. The knowledge of C_{OB} will help us design contrast-enhancement algorithms for improving the contrast in underwater environments [115].

Color Distortion: As previously discussed in Section 2.2.2 and based on (2.4), light absorption (or attenuation) in underwater propagation strongly depends on the wavelength. Accordingly, all of the visible light wavelengths present in sea surface, provide a crystal clear view. By contrast, red light hardly penetrates below 10m, which is the reason of the greenish color of seawater. Colors having shorter wavelengths (i.e. blue color family) penetrate even deeper to 20 and up to 50 m in sea water, resulting in low-contrast bluish images.

Color distortion and the visibility are the main reasons why passive imaging is inefficient

bellow 10 meters and also why the use of artificial light sources is inevitable in deep-water explorations. To address this important issue, the image enhancement or color restoration methods of Section 2.4.3 have to be involved.

Having surveyed some of the 1D signals generated by sensors and the 2D image data types encountered in IoUT, let us now focus our attention on video or 3D data, that forms a large subset of IoUT datasets.

Three-dimensional underwater video data

As discussed, some of the observatory stations listed in Table 2.6 are capable of delivering live or archival ready-to-use video data. However, the main challenges in underwater video streaming are related to the limitations of undersea communication, discussed in Section 2.2.2, aggravated by the underwater imaging difficulties studied in Section 2.3.2.

Although the underwater optical parameters and imaging constraints are all the same as in 2D imagery (i.e. contrast, color distortion, visibility, backscatter, and forward scatter), underwater video imaging has an extra intrinsic impediment that has to be addressed. This barrier, is of course, the continuously growing data volume produced by cameras. A typical camera operating at 1 frame/sec may generate 30 million frames (equal to almost 3 Tera bytes of data) per year [116]. This data volume is then multiplied by the number of cameras, mounted in a single observatory system.

As a rule of thumb, in the current version of the semi-automatic marine image-annotation software, every minute of video requires an expert to spend about 15 minutes for manual annotation [116]. Therefore, to analyze the video produced by a single camera in a single month, we approximately need 10,800 man-hour. Hence, there is a growing necessity to develop automatic video processing methods to deal with this excessive data volume. These automatic methods can be used in a variety of underwater video data applications such as, visible light video tracking [117], sonar video tracking [118], photo mosaicing underwater [46], and marine life studies [40].

In addition to the above-mentioned potential use cases of automatic video processing, the growing field of depth-based video may also significantly benefit from automatic processing. This technology uses optical multi-camera systems [119, 120], acoustic arrays [121], Time of Flight (ToF) depth sensors [122], and laser beams [123] to provide rich 3D information about the scene. This stereoscopic video technology has been used in underwater vehicles (e.g. ROVs, submersibles, etc.) [124, 125], in tethered underwater platforms [126], towed systems [117], and baited stations [127]. They are used in a wide variety of applications, including photogrammetric bundle adjustment [128, 129], 3D scene and organism reconstruction [120, 122], underwater 3D live tracking [117, 129],

quantitative analysis and sizing of targets [117, 126], counting and measurement of marine creatures [127], as well as improving the segmentation and classification capabilities of traditional 2D algorithms [127, 130].

However, the sophisticated depth-based vision technology has its own challenges both in the context of acoustic and optical recording methods. While acoustic depth vision always suffers from variable sound velocity, reverberation, as well as unwanted noise and echoes, optical stereoscopic cameras encounter blur and haziness in turbid water, aggravated by unstable illumination, and light refraction [117]. Perhaps, the most direct way of addressing this problem is to use improved acoustic and optical channel models for compensating the deficiencies of each technology [128]. Another solution is to combine both technologies into a unified opti-acoustic 3D imaging system. The latter solution is capable of offering improved precision, if both technologies cooperatively calibrate each other [125].

So far in this section, we have provided a detailed evaluation of the family of IoUT sensors, as well as of image and video data sources. Additionally, some ready-to-use databases were introduced. Once the raw big data has been gathered, a high-performance data processing platform is required for inferring knowledge. Due to the scarcity of resources and owing to the limited power budget at the data collection points, processing cannot be performed locally. Therefore, the system has to rely on cloud-based or on distributed data processing platforms, which will be studied in the next section.

2.3.3 Distributed and Cloud-based BMD Processing

To meet the growing demand for big data processing, many high-quality software and services have been created to offer big data analytics. Some of these platforms are Apache, Amazon EMR, Microsoft Azure HDInsight, Cloudera, Hortonworks (which has recently been merged with Cloudera), SAP-Hana, HP-HAVEn, 1010data, Pivotal data suite, Infobright, etc. [38, 57]. Each of these services responds to the big data processing demands by either providing distributed processing frameworks or cloud computing services.

Distributed processing systems consist of networked computers, which cooperate with each other to offer high performance data processing [131]. As seen in Fig. 2.6, among all distributed big data analytics frameworks, the Apache is the dominant platform, which has been used in about 95% of all reported scientific articles. Apache software foundation is a not-for-profit corporation, founded in 1999 to support more than 350 open-source Apache software projects, and 47 of them (about 13%) are directly related to big data analytics. More than 80% of the Apache platforms have been developed in only five languages (i.e. Java ($\approx 60\%$), different types of C ($\approx 10\%$), Python ($\approx 5\%$), JavaScript ($\approx 3\%$), and Scala ($\approx 3\%$)) [132].

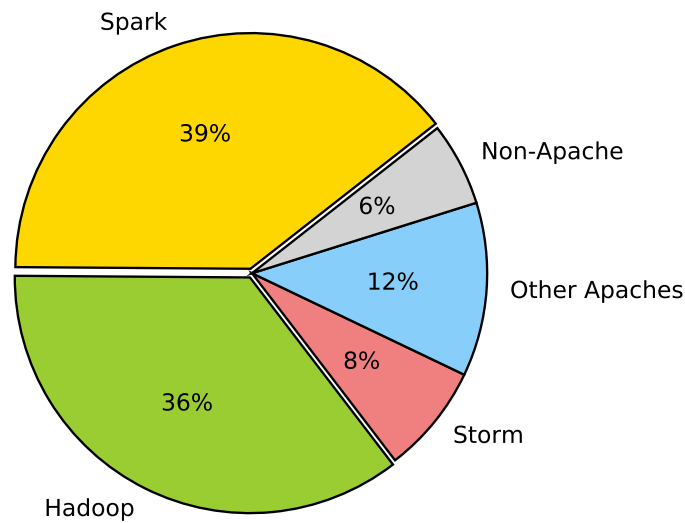


Figure 2.6: Distributed big data frameworks used in data analytics papers published in IEEE and Web of Science databases.

Table 2.8 introduces the advantages and disadvantages of the four widely used Apache distributed processing frameworks including Spark, Hadoop, Storm, and Flink. All of these open-source projects are supported by Apache software foundation and they are eminently suitable for high-speed processing of IoUT-generated BMD analytics.

The term *cluster* in Table 2.8 refers to any computer system or embedded system in the parallelized network of computers, each having its own processor, memory, and network IO. Furthermore, the directed acyclic graph scheduler is based on a specific type of directed mathematical graph having no cycles, in which it is impossible for the data to emanate from a vertex (i.e. any standalone computer) and pass over a non-zero number of graph edges (i.e. inter-computer connection cables), and to eventually loop back to the starting vertex again.

In addition to the locally distributed processing systems, there exist companies who offer cloud-based services for all aspects of big data storage, integration, streaming, ML, and *ad hoc* data analysis. According to the [Statista](#), the share of software and services in overall big data revenue will double in the 6-year period spanning from 2019 to 2025. This means that the organizations will rely more on cloud services to handle their sophisticated use cases [133]. In other words, big data processing is expected to gradually shift from distributed to cloud-based services. This is an ongoing trend, as the number of businesses performing their big data analysis in the cloud has increased from 58% in 2017 to 73% in 2018 [134].

As already mentioned in Section 2.2.6, BMD can partly be processed in the IoUT-edge,

Table 2.8: Major Apache Frameworks Suggested for Distributed Processing of BMD

Project Name	Interface Languages	Description	Advantages
			Disadvantages
Spark	-Scala -Java -Python -R -SQL	Spark uses a state-of-the-art directed acyclic graph scheduler along with a query optimizer. It also internally includes around 80 high-level mathematical operators (especially written in favor of building parallel applications) and many applicable libraries (e.g. SQL, MLlib, GraphX, Spark Streaming, etc.). All these operators and libraries can be combined seamlessly in every single application.	<ul style="list-style-type: none"> • Runs in about 100× faster than Hadoop • Multiple compatible developing languages • Capable of arbitrary complex processing like machine learning • Suitable for real-time processing of batch data and streaming data <hr/> <ul style="list-style-type: none"> • High memory requirement
Hadoop	-Java	Hadoop is a distributed batch-processor with 4 modules: Common (supports the other Hadoop modules), HDFS (distributed file system that provides data access), YARN (job scheduling and cluster management), and MapReduce (distributed parallel processing of large data sets).	<ul style="list-style-type: none"> • Suitable for real-time processing of batch data <hr/> <ul style="list-style-type: none"> • High processing latency • Low encryption capabilities • High consumption of resources (i.e. IO and Memory)
Storm	-Java -Clojure	Storm is a distributed real-time computation system which is designed as a topology in the shape of a directed acyclic graph scheduler. It reliably processes unbounded streams of data, because it always schedules the streams in a resource-aware manner.	<ul style="list-style-type: none"> • Capable of arbitrary complex processing • Stable integration with any database system • Suitable for real-time processing of data <hr/> <ul style="list-style-type: none"> • More dependency on system components
Flink	-Java -Scala	Flink is a distributed processing engine for state-based computations, which parallelizes applications (in any scale) into possibly thousands of tasks that are distributed and concurrently executed in clusters.	<ul style="list-style-type: none"> • Operable over unbounded data streams • Suitable for real-time processing of pipelines <hr/> <ul style="list-style-type: none"> • Possibility of latency over network communications

before being transmitted to land. After transmission, there is no difference between the conventional IoT and IoUT-based distributed or cloud-based data processing. Nevertheless, all available cloud services today are offered in accordance with one of the following models [135]:

- *Software as a Service (SaaS)*: This service model relies on centrally hosted software, which delivers specific services to licensed or subscribed clients and usually offers its dedicated functionality through a browser.
- *Platform as a Service (PaaS)*: This service model targets developers by providing them with operating systems, databases, software packages, application services, etc. It helps developers to focus on the development of their diverse applications, instead of software resource management and hardware maintenance of the underlying infrastructure.
- *Infrastructure as a Service (IaaS)*: This flexible low-level service model targets both developers and businesses by providing access to the underlying infrastructure (e.g.

processors, data partitions, security, backup, etc.). This service tends to rely on high-level Application Programming Interfaces (API) in support of the network operations. While the service provider is responsible for the hardware maintenance, the users are required to configure and maintain both operating system and the required software packages.

According to the aforementioned definitions, cloud computing is capable of processing BMD with the aid of all the above service models. Among them, PaaS strikes an attractive balance between convenience and cost, undertaking the management of the operating system, software packages, and hardware elements. It is worth mentioning that, Big Data as a Service (BDaaS), which has recently appeared in technical glossaries may indeed be classified as a specific form of PaaS, where the statistical analysis software packages are offered alongside the required databases and APIs.

On the other hand, IaaS constitutes a cost-effective solution, where the service providers are responsible for maintaining the underlying hardware. Some of the top Paas and IaaS service providers are Amazon Web Services, Microsoft Azure, Google Compute Engine, DigitalOcean, Oracle Cloud Compute, Rackspace Cloud, IBM Cloud, Linode, HP Enterprise Converged Systems, Green Cloud Technologies, etc.

Despite the fact that SaaS providers tend to be user-oriented and application-specific, some of them also support underwater applications. For instance, marine image-annotation software are ready-to-use software products for semi-automatic annotation of videos and still-images. These software packages are comprehensively reviewed by Gomes-Pereira *et al.* [136], where 23 software products from more than 500 publications are summarized.

The semi-automatic solutions offered by marine image-annotation software are rarely applicable in real-time BMD analytics and they tend to rely on data analytic platforms as well as self-developed ML algorithms. As a result, every practical use case of BMD, requires a distributed on-premise platform or cloud-based data processing software for gleaning knowledge from BMD. Next, some of these IoUT data extraction applications are studied.

2.3.4 Marine Data Applications

The marine data collected by various sensors and devices of the IoUT ecosystem can be exploited by data processing platforms for compelling applications. These applications offer businesses benefits based on their own data [57]. In the following subsections, we discuss and evaluate some of these applications in the context of IoUT, including maritime applications, underwater localization, and marine life tracking.

Coastal monitoring and GIS data

By appropriately processing the available maritime sensory data, some secondary parameters can be collated with the aid of the IoUT. These include the accurate localization of marine vehicles, the provision of weather and climate data for specific oceanic locations, accessing biogeographic data such as the recognition, counting, and distribution of underwater species, etc. Hereafter, we will refer to all these secondary parameters as IoUT Geographic Data, because they are devoted to studying the physical environment, the inhabitants, locations, and things in the particular area of the sea.

Having accurate geographical data is essential in IoUT applications. For this type of IoUT data, there exist a number of open-source on-line geographical databases. These databases are mainly focused on maritime information systems, a particular type of Geographic Information System (GIS), which is indispensable in efficient international transportation [137]. Fortunately, most of these GISs can be readily merged with the IoUT infrastructure in support of maritime organizations with the aid of tracking and routing information, etc. They can also provide up-to-date access for researchers in order to monitor the global ecosystems.

Kalyvas *et al.* [56] surveyed more than 180 free real-world GISs having open access databases based on the most trustworthy on-line data sources. They have categorized the GISs into 19 classes, which are distilled into as few as four classes here for simplicity. These classes and their applications include:

- *IoUT tracking data*: For vessel monitoring, marine accidents, anti-shipping activities, and navigational aid systems;
- *Marine cartographic data*: For essential naval data (like borders), protected and sensitive areas, port status, port locations and facilities, coastline as well as land areas, and bathymetry maps;
- *Oceanic climate data*: For nautical weather, natural hazards, tides as well as eddies, and satellite imagery;
- *IoUT commerce data*: For shipping companies, flags of convenience, and marine conservation organizations.

An important category that could be added to the above-mentioned classes of Kalyvas *et al.* [56] is Biogeographical Data for addressing the geospatial distribution of underwater species. In order to define this new category, the open-access [OBIS](#) project (Ocean Biogeographic Information System), funded by UNESCO can be used. OBIS is connected

to more than 500 databases in 56 countries and provides observation of 120,000 marine species down to 11,000 m depth, from bacteria to whales.

In addition to the aforementioned need for Biogeographical data, under the Marine Cartographic Data category detailed in [56], there is a missing sub-category for Georeferenced Locations. To elaborate a little further, georeferencing in GIS is a subcategory of navigational assistance that aims for precisely associating locations with their equivalent points on the physical maps, which is achieved by the [Marine Regions](#) project. This project is composed of a list of 55,000 georeferenced marine places, gathered from the [VLIMAR](#) Gazetteer and [MARBOUND](#) databases.

In addition to the GIS databases, there should also be a directory of applications, which require access to precise undersea locations. This is called underwater positioning. In the following sections, undersea localization of sensors and vehicles, along with the marine life tracking methods will be discussed in more details.

Underwater localization

A very useful data type that is included in IoUT is related to the positioning of undersea devices, systems, animal species, and data sources. This is even more critical when there is a need to geo-tag IoUT sensory and imagery data.

Underwater positioning is a challenging task, because the otherwise ubiquitous navigation signals of the GPS satellites do not penetrate seawater. Therefore, other underwater navigation methods should be used, including blind positioning relying on miscellaneous inertial sensors, acoustic transponders (with separate transmitters and receivers), ranging sonars (with only a single standalone acoustic transceiver), image-based positioning (using cameras to localize), and Simultaneous Localization and Mapping (SLAM).

Due to the associated challenges including the long latencies, multipath fading, Doppler shifts, frequency limitations, sparse deployment of the nodes, and their high mobility in underwater networks, a single navigation technique will never offer a flawless performance [138, 139]. Therefore, all the vehicles, fixed stations, sinks, and nodes in IoUT applications usually combine some of these localization methods to achieve improved performance in underwater environments. Here, we briefly review the aforementioned localization techniques and provide insights concerning their advantages and limitations.

Blind positioning: which is also known as inertial navigation, is based on the knowledge of a device's relative orientation, acceleration, velocity, and gravity anomaly (i.e. difference between the observed gravity and the predicted value [140]). In this localization method, the underwater device has to determine its position automatically, i.e., without any positioning support from a ship or transponder. In this method, a combination of

sensors is used to estimate the current location.

These sensors include a typical magnetic compass for direction detection, a pressure sensor for underwater depth estimation, a doppler velocity log for velocity measurement, a ring laser gyroscope or Micro Electro-Mechanical Systems (MEMS) gyroscope for angular velocity detection, and a pendulum or MEMS accelerometer for force and acceleration calculations [54].

However, these sophisticated blind positioning methods suffer from propagating errors. Even a small positioning error remains in the memory of the system and aggravates the future measurement errors, leading to an unbounded error propagation. Nevertheless, the method's estimation of the exact position can be improved by a variety of integrated sensors. Furthermore, blind positioning is a power-efficient method, compared to other positioning techniques [141]. Hence these positioning techniques contribute to almost all modern underwater positioning systems [138–142].

Acoustic transponders: This localization technique uses a transmitter and receiver pair for measuring the ToF of a ping signal in order to perform navigation. ToF [sec] is applicable both to the adjacent transmitter/receiver (i.e. where the transmitter and receiver are closely located) and separate transmitter/receiver (i.e. where transmitter and receiver are distantly located). It is also known as the time of arrival in separate transmitter/receiver scenarios [139]. By measuring the ToF and the acoustic wave speed, one can precisely measure the distance in [m]. This positioning method has diverse categories, including:

- *Acoustic array:* Similar to the concept of GPS satellites, and by using more than one beacon transponders, the system will be able to determine the position in any underwater location based on the phase-difference of the signals arriving at the transceivers. Short- as well as ultrashort-baseline [35] and long-baseline [143] techniques may be used in this category. In short- and ultrashort-baseline relying on ship-mounted transponders, the undersea system localizes itself relative to the ship floating at the surface; while in long-baseline that uses GPS-intelligence as well as dispersed buoys and beacons, the location can be determined.
- *Single fixed transponder:* The idea behind fixed beacon based positioning is shown in Fig. 2.7. The AUV in this picture has an uncertain prior knowledge about its position (region A). However, it knows its distance from a fixed beacon subject to a degree of uncertainty (region B). With the advent of combining these two pieces of information, the AUV finds its position subject to a reduced uncertainty (region C). Due to the fact that it uses only a single geo-referenced beacon, this positioning system is cheaper and easier to install than multi-beacon long-baseline techniques [144].

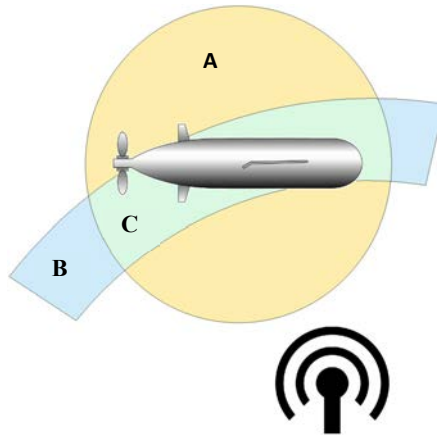


Figure 2.7: Using a fixed beacon at known location to decrease the location-uncertainty of an AUV from ‘A’ (intersecting with ‘B’) to ‘C’.

- *Cooperative navigation*: This method is also known as modem-based navigation, because in recent advanced positioning systems, modem transponders are also used to send beacons in support of navigation based on the ToF [145]. These modems do not have to be stationary, hence they can be installed on a moving vessel having a known geo-position or a swarm of underwater vehicles to communicate and to cooperatively localize each other.

Range sonars: Sonars are robust, well-established standalone acoustic transceivers that were originally used for imaging and ranging. The main ranging sonars used in underwater navigation and mapping include [54]:

- *Echo sounder*: A single narrow beam is used for determining the distance from obstacles [146] or from the seafloor.
- *Multi-beam*: Measures the ToF for each beam to assemble a bathymetric map [147].

Image-based positioning: This technique uses environmental images, taken either by monocular or stereoscopic cameras (or even by imaging sonars) for navigation. In this positioning method, substantial processing power is required for feature extraction and for processing in order to detect and identify key points, objects, and regions of interest.

The main idea behind the vision-based navigation, also known as visual odometry, is to capture images of the seabed and then to match subsequent images in order to navigate [148, 149]. stereoscopic cameras have the advantage of capturing 3D transformations between consecutive image pairs.

It is plausible that errors can propagate and accumulate in the context of this technique. To address this problem and achieve a bounded positioning error, the SLAM tactic can be merged with image-based positioning for improved localization.

SLAM: Concurrent or Simultaneous Localization and Mapping is defined as the process in which an autonomous vehicle builds a map of a specific area and also localizes itself within it [54]. The mapping process of SLAM may rely on a variety of devices such as cameras, sonars, or inertial sensors, respectively, leading to vision-based, sonar-based, or sensor-based SLAM techniques [150, 151]. In all cases, the features of the sensed data or captured image are extracted. Then, based on those features, the position is detected and stored.

Marine life tracking

At first glance, underwater animal tracking seems to be nothing more than a memory-aided localization method in IoUT, which was the subject of the previous section. However, undersea animals may be quite small and they cannot carry relatively heavy inertial sensors, transponders, sonars, or cameras. Therefore, sophisticated new tracking methods have to be devised.

Additionally, as discussed in Section 2.3.4 and also bearing in mind the penetration depth formula of (2.1), the GPS signals having a frequency of 1.2 and 1.5 GHz cannot penetrate seawater beyond a few meters. Accordingly, alternative marine life tracking methods have to be implemented in IoUT applications, some of which are evaluated in this section. All these methods can make use of a data storage tag to archive data on a memory chip for future retrieval [152].

RFID tags: This tracking technology uses Radio Frequency Identification (RFID) patches and has a limited range of about 10 m in freshwater. However, it does not work well in seawater owing to its high salinity. RFID tags are equipped with a unique Identification code and need an external energy source in the form of a low frequency signal, in order to become activated and to retrieve data. These tags are also available in passive integrated transponder form, which are specific implantable RFID devices [32].

Acoustic tags: This tracking technology emits signals in the form of acoustic waves and has a reasonable transmission range both in fresh and seawater. In this method, a pinging sound with an embedded ID is periodically transmitted to an array of remote acoustic sensors (hydrophones). This ping is used to locate the animals and the ID is used to identify them [152].

Image-based tracking: Perhaps image-based tracking is one of the most advanced underwater tracking techniques reported to date. It uses images of a moving or fixed camera

system to track, recognize, count, measure, and also study the animal's behaviors. In a fixed camera based system, usually a baited or trawled station is used for attracting intended species [117]. On the other hand, the tracking of animals in a moving camera based system uses cameras installed on a moving platform. The main challenge in these systems is to promptly process the large variety of images [47].

2.3.5 Section Summary

In this section, we have studied the essential topics of big marine data. At the beginning, we introduced the five BMD system components, followed by further elaborations on the first component, i.e. on data acquisition. Thereat, we reviewed the underwater data gathering tools and techniques, along with the data aggregation protocols for more compactly representing the gathered data. We also studied data fusion methodologies, conceived for the fusion of data from distinct datasets into an integrated database, for the sake of having more generally representative information. Our discussions in this section were then continued by classifying BMD into 1D sensor signals, 2D image data, and 3D video streams. For all these categories, we surveyed the typical IoUT infrastructures and observatory systems, which freely offer their BMD for researchers. The visibility, contrast, color distortion, and light scattering in 2D underwater images as well as 3D video data gathering were also addressed. Then, different distributed as well as cloud-based BMD processing frameworks were introduced and a couple of essential BMD applications were studied. These applications covered the oceanic GIS, underwater localization, and marine life tracking, which all play essential roles in IoUT contexts.

All of the tools and methods that have been discussed in this section, including the sensory and imagery data sources, GIS-based ready-to-use data in IoUT, and underwater object tracking as well as geo-tagging methods will generate BMD, which has to be processed and analysed. In the following section, we will focus our attention on big data processing techniques. Thereby, we will review state-of-the-art technologies in data cleansing and data processing using ML techniques for underwater applications.

2.4 Machine Learning for BMD Analytics

As mentioned in Section 2.3, *data processing* is one of the five critical system components in the chain of wealth-creation from data. To address the commercial and industrial demands, ML and its Deep Learning (DL) variant constitute promising solution. In this section, as visualized in Fig. 2.8, we focus our attention on the use of ML strategies including the classical ML approaches, as well as the more traditional and emerging NN-based learning methods for BMD processing.

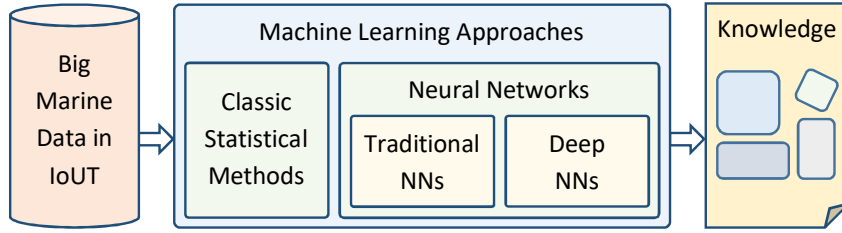


Figure 2.8: Conceptual categorization of the main topics in Section 2.4.

2.4.1 Machine Learning Techniques

Considering the large amount of data generated both in IoT and IoUT, there is a growing need for powerful tools and techniques capable of distilling and visualizing knowledge gleaned from data. These tools and techniques have evolved over the past century! As a benefit, ML is readily available for the analysis of BMD [58]. ML can be viewed as a collection of software algorithms that empower intelligent machines to improve their performance on an accomplishing pre-configured specific task. This particular task in turn, may be categorized under descriptive, predictive, and prescriptive models.

The ML approaches may be divided into supervised and unsupervised methods. The unsupervised methods are mainly used for data clustering, according to the features embedded in the data itself. These methods do not rely on an expert for labeling and for entering inputs. On the other hand, in supervised learning, data will be labeled, prior to its exploitation, by an expert. In this case, the resultant ML solution is referred to as a classifier, which is then involved for the ensuing data classification phase.

Classification methods are devised for categorizing data, representing for example a measurement from a sensor or a pixel in an image into one of the legitimate predefined output classes. When there are only two possible answers to a given question (i.e. yes or no), the classification problem is often termed as *detection*. In cases of three or more legitimate output classes associated with multiple choices, classification is also often referred to as recognition. Methodologically, classification algorithms can be divided into statistical methods (also known as conventional data analytics) and Neural Networks (NN). The neural-based methods may themselves be divided into traditional NNs and deep NNs [153].

All ML approaches are designed to assign a given input data (X) to a set of predefined classes (C_i). This is a well-known classification/clustering problem, which can be solved by either generative or discriminative models. The generative models are those which solve this problem by using the joint probability distribution function of [154],

$$i = \arg \max_C P(C)P(X|C). \quad (2.17)$$

By contrast, the family of discriminative models uses the conditional probability distribution function of [154],

$$i = \arg \max_C P(C|X). \quad (2.18)$$

According to the Bayes' theorem from a theoretical perspective, these methods are identical. In practice however, it is usually easier to calculate $P(C|X)$; especially when we have a large amount of data (i.e. X) for training our model. The conventional data analytic methods tend to use generative models, while most NNs rely on discriminative models [155].

In generative models, the behavior of both C and X should be known. However, for discriminative models, we directly deal with the unknown C , based on a given X . Discriminative models are generally simpler, faster, and have less parameters to adjust. Accordingly, while the family of conventional statistic approaches may be good enough for handling moderate-dimensional situations, traditional ML and modern DL based approaches (with discriminative models) are needed for processing big marine data problems (i.e. image, video, and other sensory information generated within the IoUT infrastructure).

The concept of DL is built around the idea that artificial neurons are capable of automatically extracting features and learn a pattern, provided that there are enough hidden layers and unweighted neurons in their networks. Based on this concept, deep networks have evolved from the traditional NNs by invoking more than one hidden layer (technically, dozens of layers). Using these extra layers, deep networks become capable of extracting features and reduce the learning dimension as it will be introduced later in Section 2.4.4.

The growth rate of DL usage in all ML publications in the post-2010 era is demonstrated in Fig. 2.9. This figure shows an astonishing factor 25 increase in the percentage of publications in as few as four years. These publications have substantially advanced the field by proposing new algorithms, networks, and strategies for improving the performance of deep networks. Table 2.9 presents the most recognized supervised and unsupervised deep learning networks. All the tandem arrows and circular nodes in this table ($\rightarrow\bigcirc$) are representative of an artificial neuron processing the sum of weighted inputs and a subsequent activation function (e.g. Sigmoid, tanh, etc.). Please note that every NN has a bias input. These inputs are deliberately eliminated in the shown diagrams, for presentation simplicity.

Again, deep networks have several hidden layers. This is shown in Table 2.9 by utilizing a dash-dot line ($-\cdot-\cdot-$), whenever the network is capable of incorporating multiple hidden layers. However, it should also be considered that, as these networks grow deeper, they also require more training data. Further details on various types of networks and on their advantages/disadvantages are provided in Table 2.9 and will be discussed throughout

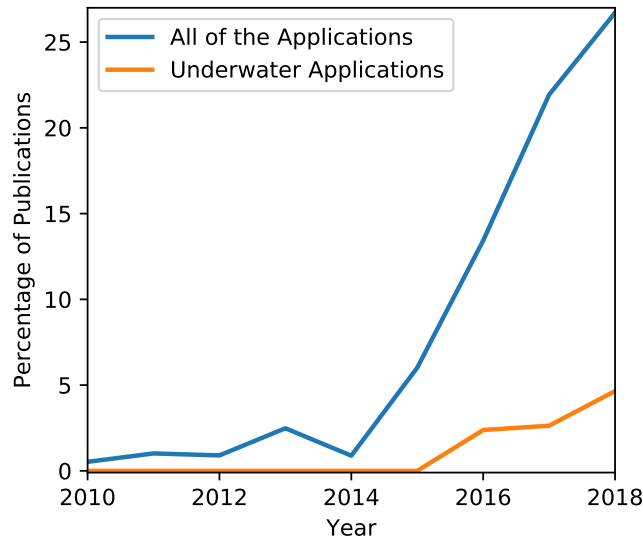


Figure 2.9: Percentage of deep learning usage in all machine learning publications, as searched between IEEE Explore as well as Web of Science databases.

this section. However, we study these networks mainly from an IoUT as well as BMD perspective and discuss what specific network types and algorithms would be particularly suitable both for BMD analysis and for IoUT applications. In doing so, we appraise many reported use cases of ML in underwater applications in Section 2.4.6, but before that, let us discuss the various software frameworks for architectural deep network design, suitable for big marine data types.

2.4.2 Deep Learning Frameworks and Libraries

To facilitate the development of various DL architectures, such as those listed in Tables 2.9 and 2.14 for different applications, numerous software frameworks and libraries have been developed for open access by the ever-growing deep learning community [156]. Table 2.10 lists and compares a range of open-source DL tools and frameworks. Some of the abbreviations used in this table are Deep Learning for Java (DL4j), Microsoft Cognitive Toolkit (previously known as CNTK), and TensorFlow (TFlow).

All the frameworks seen in Table 2.10 are capable of operating on NVidia[®] CUDA-supported GPUs relying on parallel processing. By contrast, OpenMP is another shared-memory based multiprocessing programming interface, which is only supported by some of the frameworks. As it will be discussed in Section 2.4.5, shared-memory based methods are critically important for speeding up the deep network's training process.

So far we have discussed various architectures capable of processing BMD and marine data. Once the data has been collected, some pre-processing steps have to be performed

Table 2.9: Deep Networks with Input, Hidden, Mid-output, and Output Layer Neural Nodes to Evaluate Static, Dynamic, Sequential, or Hierarchical Input Data Types [9–13]

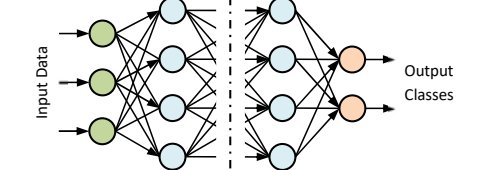
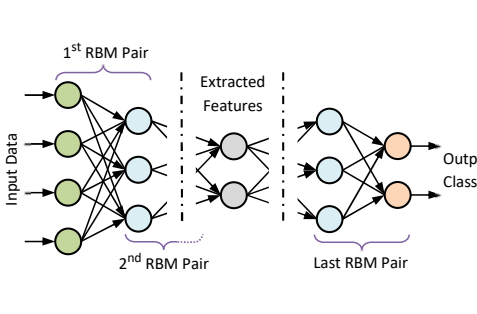
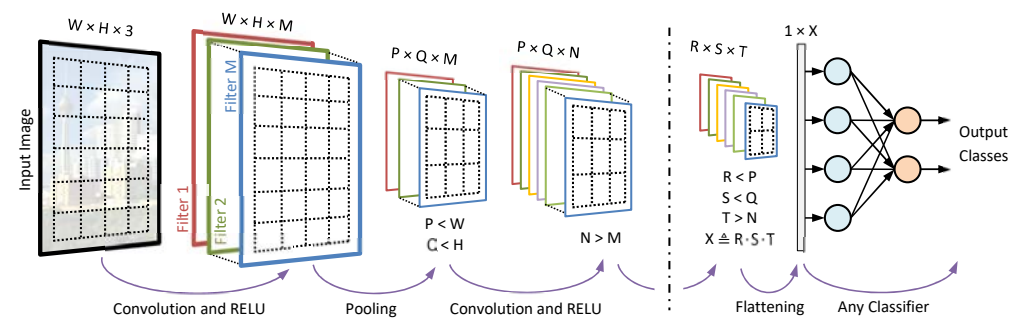
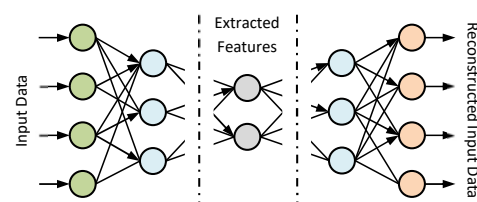
Fully-Connected (FC) or Dense Network		<p>Description: This is a nonlinear multipurpose data classifier. The input data passes through the weighted vertices and nonlinear activation functions, to the output layer, where its class will be decided.</p>	<p>Pros:</p> <ul style="list-style-type: none"> • It is widely used and well documented. <p>Cons:</p> <ul style="list-style-type: none"> • Gradient vanishing and gradient explosion (for unstable derivatives) and slow convergence during the BP training process are expected.
Deep Belief Network (DBN)		<p>Description: DBN is structurally similar to FC networks, but they differ in their training. DBNs are trained by exploiting the concept of successive RBM pairs. Every layer in DBN is a visible RBM layer for the next neighbor and a hidden RBM layer for the previous one. This pairwise training is a beneficial substitution for the BP training scheme in FC networks. DBN as a supervised method can use labeled data to fine-tune the weights, after RBM initialization.</p>	<p>Pros:</p> <ul style="list-style-type: none"> • It overcomes the gradient vanishing and gradient explosion problems in BP method, by employing the RBM training scheme. • By using the Kullback-Leibler (KL) divergence metric, the information loss (or relative entropy) is considered as the cost function. • Can be used in both supervised and unsupervised learning. <p>Cons:</p> <ul style="list-style-type: none"> • It usually requires larger training time, especially when it undergoes a fine-tuning step.
Convolutional Neural Network (CNN)	 <p>Description: CNN is probably the most popular data and image processing deep learning network by far. The input, for example an image of size $W \times H$ and 3 color channels, is partitioned with overlapping channels at first. Then a series of filters (or kernels) are convolved with each window to extract features. Despite the filters can be initialized by the kernels of Table 2.14, but their weight might change during the train process. The filtered values are then guided to pass a nonlinear activation function (i.e. rectifier (RELU), \tanh, sigmoid, etc.). Afterward, the 1st and 2nd dimensions of the resultant matrix will be reduced in the pooling step. The number of consecutive convolution and pooling steps can be adjusted. Finally, the multi-dimensional matrix will be flattened to a 1-dimensional vector and will be passed to a classifier network (e.g. a FC network) to decide about the appropriate output class.</p>	<p>Pros:</p> <ul style="list-style-type: none"> • CNNs have a variety of architectures, many of which are widely used and well documented. • It has already been embedded in many software libraries and frameworks, making it available in high-level programming. • Less neural connections are required in CNN, because neurons are not fully connected. • Every neuron in the CNN learns a patch of the image. This contrasts with other methods (e.g. DBN) where every neuron has a contribution in learning the entire image. <p>Cons:</p> <ul style="list-style-type: none"> • CNN usually requires a lot of consecutive convolution and pooling layers to deal with a simple image. • It usually requires many labeled data to be appropriately trained. 	<p>Pros:</p> <ul style="list-style-type: none"> • It offers a wide range of applications including data denoising, dimensionality reduction, data compression and decompression, etc. <p>Cons:</p> <ul style="list-style-type: none"> • The network structure is robust and the feature vector size cannot be adjusted, after training process.
Autoencoder (AE) Network		<p>Description: This network consists of two separate parts. The first part extracts the feature vector from the input data, while the second part reconstructs the same data, from the extracted features. After training the network, these 2 parts can be used concurrently or separately.</p>	<p>Pros:</p> <ul style="list-style-type: none"> • It offers a wide range of applications including data denoising, dimensionality reduction, data compression and decompression, etc. <p>Cons:</p> <ul style="list-style-type: none"> • The network structure is robust and the feature vector size cannot be adjusted, after training process.

Table 2.9: (Continued)

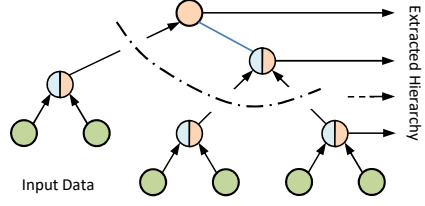
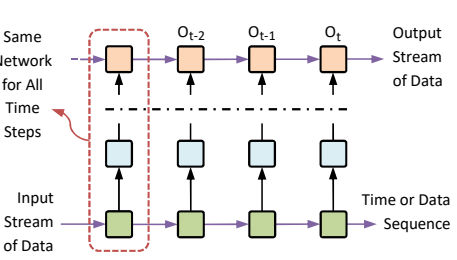
<p>Recursive Neural Tensor Network (RNTN)</p>		<p>Description: The RNTN is a deep version of shallow autoregressive models. It is suitable to extract hierarchy in almost any sort of input data (i.e. syntactic parsing, image segmentation, etc.). It can also be used in sequential data analysis (e.g. sentiment analysis in text messages).</p>	<p>Pros:</p> <ul style="list-style-type: none"> • The tree structure of RNTN is completely customizable and can be tailored to any application. • Any neural node of the network can also be used as an output. <p>Cons:</p> <ul style="list-style-type: none"> • RNTN is a memoryless network, which is not recommended in dynamic system evaluation.
<p>Recurrent Neural Network (RNN)</p>		<p>Description: RNN is a deep and deterministic version of the well-known Hidden Markov Model (HMM) of predicting any sequential data or time-series. In RNN, HMM layers are stacking on top of each other to create a deep network with intrinsic memory. The concept of point attractors in their state space acts as settling points to recall memories from the past.</p>	<p>Pros:</p> <ul style="list-style-type: none"> • RNNs are very suitable in modeling complicated and nonlinear dynamic systems. • Their distributed hidden state (i.e. the intrinsic memory in hidden layers) helps them to effectively store information about the past. <p>Cons:</p> <ul style="list-style-type: none"> • They might show a degree of chaotic behavior. • RNNs are hard to train.

Table 2.10: Comparing the Best Open-source Software Frameworks and Libraries for Deep Learning

Framework Name	Initial Release	Operating Systems	Interface Languages	Parallel Support		Deep Networks Support					Description
				GPU	OpenMP	FC / AE	DBN	CNN	RNN	RNTN	
TFflow	2015	Linux, Win, Android	Python, C++, Java	✓	✗	✓	✓	✓	✓	✗	<ul style="list-style-type: none"> • Most used framework • Great documentation and community support
Keras	2015	Linux, Win, Android	Python, C++, Java	✓	✗	✓	✓	✓	✓	✗	<ul style="list-style-type: none"> • Operates on top of TFflow or Theano • Offers a high-level interface
CNTK	2016	Linux, Win	Python, C++	✓	✓	✓	✗	✓	✓	✗	<ul style="list-style-type: none"> • High performance on big datasets • Easy training
DL4j	2016	Linux, Win, Android	Python, Java, Scala	✓	✓	✓	✓	✓	✓	✓	<ul style="list-style-type: none"> • Used in industrial big data applications • Has a rich set of deep networks support
MXNET	2016	Linux, Win, iOS, Android	Python, C++, Scala	✓	✓	✓	✓	✓	✓	✗	<ul style="list-style-type: none"> • High efficiency and flexibility • Suitable for image classification
Neon	2016	Linux, iOS	Python	✓	✓	✓	✓	✓	✓	✗	<ul style="list-style-type: none"> • Optimized for Intel CPUs • One of the fastest frameworks
PyTorch	2016	Linux, Win	Python	✓	✓	✓	✓	✓	✓	✗	<ul style="list-style-type: none"> • Python version of the Torch framework • High-level object-oriented programming
Caffe2	2017	Linux, Win, iOS, Android	Python	✓	✗	✓	✗	✓	✓	✗	<ul style="list-style-type: none"> • Efficient for CNN applications • Fast operation

for ensuring that the data used to train the ML algorithm is clean and does not result in future learning problems. In the next section, this important issue is discussed in more

details.

2.4.3 Marine Data Cleaning for Machine Learning

Data preparation and cleansing is a critical step in almost every ML project and it usually takes over half of the entire project duration to collect and clean the data. When dealing with the inaccessible harsh underwater environment of the IoUT, data cleansing becomes even more important. This is because the underwater environment is affected by many environmental factors, which make it almost impossible to acquire clean data. Additionally, data sources tend to be quite vulnerable to the hostile underwater environment. Therefore, data preprocessing and cleaning, before training is crucial for the success of any ML aided BMD processing in IoUT. Below, various IoUT data types and their main preparatory considerations are discussed.

Underwater sensor data cleaning

Sensors are inevitable parts of every IoUT subsystem and are used in nearly all underwater exploration and surveillance applications [157]. They continuously measure physical, chemical, as well as biological parameters and generate a huge volume of data. Examples of these sensor and data are listed in Table 2.2.

Data cleansing for sensory data sources in underwater BMD applications is typically concerned with missing values, contaminated measurements, and detecting outliers. There are a variety of techniques for these preparatory steps. To deal with a missing value in any sensor data, one can [158],

- Delete the entire record;
- Use a global constant;
- Use a statistical value (e.g. mean or median);
- Use an inter-class statistical value;
- Use the most likelihood (i.e. more probable) value;
- Use the most likelihood inter-class value.

Noise is another phenomenon affecting almost any sensory data. The process of de-noising a sensory measurement is termed as *noise cancellation*. Although no universally applicable de-noising methods exist, one should find the one meeting the SNR criteria. The SNR is a very common parameter of characterizing the performance of different noise cancellation methods. Some of these methods include:

- Using a low pass filter [159];
- Using regression models [160];
- Applying a data binning method [161];
- Using wavelet methods in time-series [158].

Among these methods, low pass filter is the most common one and wavelet-based methods are the most complex ones to implement. Additionally, data binning is not particularly popular for sensory data, but it is widely used for images as it will be discussed in the next section.

The final step in underwater sensory data cleansing is the outlier detection. This step can be readily carried out by considering three well-known criteria, named the IQR, T^2 , and Q criteria [158]. There are also some more advanced methodologies like, probabilistic models, clustering methods, distance-based detection, density-based detection, etc. [158].

Underwater image and video data cleaning and quality enhancement

As we have already discussed in Section 2.2.2, underwater image acquisition suffers from strong absorption and scattering. Additionally, there are two other undersea signal degradation factors, namely chromatic aberrations [162–164] (i.e. color distortion or chromatic distortion) and noise imposed by both natural and artificial light sources [165]. These destructive factors significantly affect the quality of the captured video or image and should be mitigated by a data cleansing algorithm.

Any image and video preparation algorithm has to assess the image quality first. Automatic image quality evaluation algorithms assign an objective metric, which is a weighted sum of the image colorfulness, contrast, sharpness, etc. [166, 167]. This metric is then used as an objective function to be maximized by other image enhancement procedures. Therefore, underwater image and video quality assessment constitutes an essential step before image retrieval, image quality optimization, video compression, and other visual signal processing steps. It can also act as a guide for determining the data bandwidth required by the underwater communication algorithms, as well as by other decision-making processes [168].

After the automatic quality assessment of the underwater images and videos, software-based image preparation algorithms are used for enhancing the image quality. As a general rule, it is always cheaper to improve the image quality in software, instead of implementing bespoke high-cost imaging devices.

In a comprehensive review paper by Han *et al.* [112], the image preparation methods are divided into image dehazing and color enhancement. The authors then introduce and

compare numerous methods for each category. However, they have missed the innovative method proposed by Ancuti *et al.* [169] that addresses both image dehazing and image enhancement at the same time. This method improves the global contrast that was degraded by light absorption, increases the edge sharpness impaired by light scattering, and exposes the dark regions with the aid of color balancing. The proposed method relies on a single-image camera-independent technique that can be applied to both photos and full-motion videos. Furthermore, it does not require any prior knowledge about the environmental conditions.

Underwater image data binning

In addition to image enhancement, image data binning is another data cleansing method that can be used in IoUT, which relies on grouping pixels into distinct partitions based on the similarity of their characteristics [161]. The goal of constructing these partitions is both to reduce the amount of noise and to resolve data complexity. After binning, every pixel has a label to explicitly indicate its parent partition, based on its membership index. Although image binning is not an essential preparation step, it is recommended before invoking feature extraction for reducing the computational complexity and to speed up the ML process [161].

Image clustering methods, such as image binning, were developed much earlier than the state-of-the-art ML and DL algorithms and date back to the age of statistical mathematics. However, none of these methods are universally accepted and it is still a challenging task to select the most appropriate image partitioning method for a given application. A list of image clustering methods, that are beneficial in underwater applications is provided in Table 2.11. These methods either tend to exploit image discontinuities (like edges) or similar regions to partition a given image.

In addition to the algorithms introduced in Table 2.11, there are many others that have however been more rarely used in the literature. These methods are reviewed by Flake *et al.* [186] as well as Wang *et al.* [161] and might be worth investigating in future studies. Some of them are the Minimum Cut, Mean-Shift, Turbo-Pixels, Lattice Cut, Compact Super-Pixels, Constant Intensity Super-Pixels, Entropy Rate Super-Pixels, Homogeneous Super-Pixels, Topology Preserved Regular Super-Pixel, SEEDS, VCells, Depth-Adaptive Super-Pixels, Voxel Cloud Connectivity Segmentation, Structure Sensitive Super-Pixels, Saliency-based Super-Pixel, and Linear Spectral Clustering.

In addition to data cleansing, feature extraction is another essential step for any ML applications, which rely on statistical methods and traditional NNs. This substantial step in ML is discussed in the next section.

Table 2.11: Image Clustering Methods and Algorithms in Underwater Applications

Method	Algorithms	Underwater Applications
Edge-detection (based on the image derivatives)	<ul style="list-style-type: none"> • Watershed • Snakes • Blob detection 	<ul style="list-style-type: none"> • Jellyfish Detection [170] • Plankton Recognition [171] • Megafauna Recognition [172]
Thresholding (based on a threshold value (clip-level) in feature space)	<ul style="list-style-type: none"> • Otsu's Method • Expectation-Max • Maximum Entropy • Hist. Thresholding 	<ul style="list-style-type: none"> • Fish Detection [173] • Jellyfish Detection [170] • Megafauna Recognition [174] • Plankton Recognition [51, 175]
Distance- or Region-based (based on the similarity in attributes)	<ul style="list-style-type: none"> • Region Growing • SLIC • Region Splitting • MSPA • Statistical Clustering • Neural Clustering • Fuzzy Clustering • Wavelet Transform • Compression 	<ul style="list-style-type: none"> • Shrimp Detection [49] • Plant Detection [46] • Plant Recognition [130] • Plankton Recognition [50, 176] • Coral Detection [177] • Object Detection [44] • Object Detection [43] • Object Detection [45] • Underwater Clustering [178] • Underwater Clustering [179] • Coral Recognition [111] • Coral Recognition [180]
Math Calculus (based on solving a Differential or Integral Equation)	<ul style="list-style-type: none"> • Level Set • Mumford Shah • Chan-Vese 	To the best of authors' knowledge, this method has never been used in underwater applications.
Graph Cut (based on the undirected graph partitioning to model the impact of pixel neighborhoods)	<ul style="list-style-type: none"> • Normalized Cuts • Random Walks • Iso. Partitioning • Min. Spanning Tree • GrabCut • Model Based 	<ul style="list-style-type: none"> • Fish Recognition [181, 182]
Video Motion (based on the object movement)	<ul style="list-style-type: none"> • Subtracting Images 	<ul style="list-style-type: none"> • Fish Detection [183, 184] • Fish Recognition [116] • Lobster Detection [185]

2.4.4 Feature Extraction for Marine Data Analytics

Feature extraction is a mathematical step, in which raw data is replaced by its numerical descriptors. This step is responsible for transforming large vectors of sensory data and

large matrices of image data into their low-dimensional vector-based representatives. It is usually the most intricate part of almost any ML-aided computer vision problem and the solution should be tailored for the particular needs of the specific project at hand. Although all of the modern deep learning algorithms extract their own feature vectors automatically, these deep networks are hard to train, when relying on limited training datasets.

Finding a series of useful features is even harder in underwater image processing applications in high-dynamic environments subject to non-uniform light illumination, variable scene brightness, and degraded colors. Table 2.12 lists a number of salient descriptor routinely used in underwater applications. These features are categorized into four classes, including color, texture, shape (boundary), and other descriptors. The papers listed in Table 2.12 tend to use a series of descriptors, depending on their specific target applications. In one case, the number of descriptors used has been as high as 66 features in the Fish Recognition project carried out by Huang *et al.* [182].

After selecting a number of features, it is recommended to mathematically evaluate their correlation and then reduce the number of features to the number of truly independent descriptors. This process is termed as *data dimensionality reduction* and can be performed by feature reduction techniques, feature selection, and feature aggregation [194].

Some of the popular dimensionality reduction techniques found in literature are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), independent component analysis, non-negative matrix factorization, Self-Organizing Map (SOM), sequential forward search, sequential backward search, bag of words, etc. [194–198].

So far in this section, the software components suitable for ML and DL aided IoUT have been studied. However, these ML-aided solutions also rely on appropriate high-performance hardware platforms.

2.4.5 Hardware Platforms for ML in IoUT

The implementation of any ML solution, from its data cleansing and algorithm development to its final deployment, may rely on a variety of hardware platforms. These hardware platforms typically boost the overall throughput by parallel processing. These processing methods can be broadly divided into the following two main categories:

- *Shared-memory multi-processors*: They rely on multiple processors that all share a memory unit [199] and the ML algorithm resides within this memory unit. Some of the more popular shared-memory methods are based on Application-Specific Integrated Circuits (ASIC) [200], Field-Programmable Gate Arrays (FPGA) [201], multi-purpose and multi-core CPUs, and Graphics Processing Units (GPU) [202].

Table 2.12: Data Processing and Machine Learning Feature-sets in Published Underwater Applications

Color Descriptors	Underwater Applications
• Image statistics (e.g. Hu’s 7 invariant moments, etc.)	[49–51, 116, 127, 176, 180, 182, 187]
• Segment contrast	[49]
• Histogram descriptor	[45, 49, 111, 172, 181, 182, 188]
• Five MPEG7 color features	[172, 174]
• Transparency ratio	[175]
Texture Descriptors	
• Three MPEG7 texture features	[172, 174]
• Wavelet transform	[177]
• Gabor filter	[46, 111, 116, 130, 172, 182, 188]
• Filter banks (e.g. Schmid, maximum response, Leung and Malik, root filter set)	To the best of the authors’ knowledge, these descriptors have never been used in underwater applications.
• Grey level co-occurrence matrices	[48, 50, 51, 111, 116, 176, 182, 187, 189]
Shape Descriptors and Key Points	
• Hough transform	[53]
• HOG (e.g. SIFT [©] , SURF [©] , GLOH)	[43, 172, 181, 182, 190]
• Binary descriptors	[41, 42, 111, 191, 192]
• Weber local descriptor	[193]
• Convexity	[50, 127, 175]
• Fourier descriptors	[50, 51, 116, 176, 182]
• Frequency domain descriptors	[193]
• Curvature scale space transform	[116]
Other Descriptors	
• Granulometrics (size, area, and orientation) to recognize an already detected object	[48, 50, 51, 117, 175, 176, 182]
• Eigenvalues and covariance matrix to recognize an already detected object	[175]
• Motion related	[48]
• 3D shape, surfaces, and texture descriptors	[130]

- *System of networked processors:* As already introduced in Section 2.3.3, a Distributed Computing System (DCS) is a system of networked processors, which coordinates the processors' actions by passing messages to each other [131, 203]. They can be used to parallelize ML algorithms in three different ways, including:
 - *Data parallelization:* This is performed by running the same ML algorithm in all of the distributed computers and then dividing the data between them. Every computer estimates all parameters based on a separate dataset, before they exchange their estimates for formulating a final estimate. MapReduce is one of the most popular data parallelization methods [204, 205].
 - *Model parallelization:* In this paradigm, the ML algorithm will have to be decomposed into different functions and operands. The algorithmic parts are then shared among multiple computers and every part has its own specific set of parameters. The input data, however, will be simultaneously fed to all of them, where every computer is responsible for estimating the set of parameters assigned. During the parameter training process, those parallelized computers exchange their partial error vectors back and forth to meet the convergence criteria and to come up with the final parameter estimations [206, 207].
 - *Pipelined parallelization:* In this method, the algorithm is shared among distributed computers, similar to the above-mentioned model parallelization method. These parallelized algorithm parts are chained together from one input block to one (or more) output blocks, termed as *pipelines*. In contrast to model parallelization, data will also split into a series of records. At the beginning, the first record is passed to the first computer in the DCS, to carry out its own task (i.e. local parameter adjustment). The output of this computer is then relayed to the next computer to carry out its own task over the first data record. Meanwhile, the first computer starts processing the second data record. This pipe-lined process continues until the last record in the database exits the DCS and consequently, the local parameters of the last block in the pipeline are updated [208].

A rudimentary qualitative comparison of different parallel processing hardware platforms used for ML-aided and BMD processing is offered in Table 2.13. Here, the strength of each platform in terms of a specific criterion is indicated by the + and – signs, where more +s represents better performance in the context of that specific benchmark. The last two rows of this table summarize the entire table with respect to the associated research or industrial applications. For instance, as the Table shows, the CPUs require the lowest

Table 2.13: Comparing Different Parallel Processing Hardware Platforms to Carry out a Typical Machine Learning Algorithm

Criteria	ASIC	FPGA	CPU	GPU	DCS
Inference Speed	++	++	--	+	-
Product Unit Cost	++	+	+	-	--
Design Complexity	--	-	++	+	-
Occupied Space	++	++	-	+ ^a	--
Research Friendly	--	-	+	++	+ ^b
Market Friendly	++	+	--	-	+ ^c

^a New GPU platforms such as NVIDIA Jetson occupy small spaces.

^{b, c} Cloud-based DCS platforms such as AWS EMR are readily available.

design time and impose the lowest design complexity, but they suffer from the lowest operational speed in all design and application phases, i.e. data cleansing, NN training, and final inference. Therefore, they may be used for research, but they are less suitable for industrial applications.

Perhaps the most prominent choice for typical scientific application would be the GPU [209]. The hardware and software resources required for GPUs are affordable and their speed is high enough to cover almost any application. Both the research-based as well as the industrial-scale employment of GPUs in underwater data processing and ML applications have been frequently reported in the literature. These applications range from obstacle detection and collision avoidance, to image-based SLAM localization, and even further to underwater object detection (e.g., coral detection) [202, 210].

However, the major problem with GPUs is their dependence on a bulky host computer. This has limited their implementation in low-power and lightweight IoUT platforms as well as underwater vehicles. To address this problem, new compact GPU designs have emerged to offer AI-ready computing resources. For example, [NVIDIA Jetson](#) is a standalone GPU-accelerated embedded system, which has a small volume. The high performance, low power, and compact form-factor of the Jetson family makes them ideal for example for deep learning aided computer vision applications [210].

Since the Jetson embedded systems are empowered by the Linux Ubuntu operating system, they are eminently suitable for deep NN frameworks of Table 2.10, e.g., TensorFlow. Thus, the low-latency inference capability of machine learning algorithms can be readily realized in underwater vehicles and platforms [202].

Additionally, NVIDIA's Compute Unified Device Architecture (**CUDA**) library unleashes the GPU's parallel processing capabilities for applications other than machine vision. As previously mentioned in Section 2.4.2, CUDA is an API model that allows engineers to use a CUDA-enabled GPU for general purpose processing, this is also referred to in parlance as a *General Purpose GPU* (GPGPU). For instance, Pallayil *et al.* [211] have used a GPGPU for performing high-complexity real-time beamforming in their underwater acoustic phased array. They simply used the popular C Language in the Linux environment to harness the fast Fourier transform in the CUDA library, to implement their underwater frequency-domain beamformer.

ASICs and FPGAs are listed in Table 2.13, which have the lowest form-factor. They also offer a high throughput and high power efficiency, which makes them eminently suitable for industry-scale IoUT projects [212]. Given these compelling attributes, FPGAs are repeatedly featured in real-time and low-power underwater applications. For example, Karabchevsky *et al.* [212] have implemented a standalone FPGA architecture for noise suppression in underwater sonars. Their proposed signal processing implementation is claimed to overcome the sonar-based underwater visibility problems.

Following the above survey of various data processing methods, platforms, and their hardware realizations, diverse ML algorithms used for underwater applications will be discussed in the next section.

2.4.6 ML Techniques in BMD Applications

The advantages of ML techniques make them eminently suitable for most underwater applications. The categorical investigation of these techniques as well as their implementation in BMD applications will be carried out in this section.

Deep NNs for static IoUT data

Static data is exemplified by still images in contrast to full-motion video clips. In the context of Table 2.9, the fully-connected, Deep Belief Network (DBN) [9], CNN [10], Autoencoder [11], and Recursive Neural Tensor Network (RNTN) [12] networks are eminently suitable for static marine data processing. In the case of RNTNs, they are usually fed by a single static underwater image in every single data entry step. This image consists of multiple segments, and RNTN is supposed to determine the hierarchy of every segment inside the given image (e.g. background, coral, plant, fish, etc.). The double-colored nodes of RNTN in Table 2.9 are hidden neurons that can also be designated by the network designer to be an output neuron, representing those expected hierarchical segments.

Among the static data processing networks seen in Table 2.9, Autoencoders are the only

unsupervised clustering NNs. Every Autoencoder consists of two parts, which can be used either separately or simultaneously. The first part processes a static input data such as an underwater image and extracts its features. The second part on the other hand, takes the feature vector and tries to reconstruct the input image again. If the inherent image features were adequately captured, the reconstructed input data appearing at the output will be similar to the input data itself. From an application-oriented perspective, one can use the first part of the Autoencoder for dimensionality reduction and data compression, while the second part is suitable for data decompression. Meanwhile, by using both parts simultaneously together, Autoencoder will act like a noise reducing NN.

Another network architecture in Table 2.9 is the family of DBNs. These networks consist of consecutive shallow learning Restricted Boltzmann Machine (RBM) pairs, which gradually reduces the dimension of input data from the top-level of the entire search/classification towards the final unique classifier. Any mid-layer in DBN may act as output layer of the feature extraction. Again, by continuously decreasing the feature vector size, the procedure culminates by unambiguously classifying the input data. Therefore, DBN can act as an unsupervised clustering NN, provided that it is terminated somewhere at a mid-layer. By contrast, it can also act as a supervised classifier, if the number of nodes is reduced to the specific number of classes and if labeled data is used to train the network.

The final deep network for static data that will be discussed here is the convolutional neural network. This network relies on multiple stages of convolution and pooling, as seen in Table 2.9. It is considered to be the best deep classification method, especially when dealing with static images or previously recorded outputs of sensors and of hydrophones. Several architectures of this popular network have been designed for scientific use, some of which are also supported by Google and Microsoft (as exemplified by Inception [213] and ResNet [214]). Several of these CNN architectures are compared in Table 2.14, based on the excellent review article by Canziani *et al.* [14]. The Top-1 accuracy² listed in this table is measured with the aid of the single central-crop sampling technique³ of [15] for all of the networks.

The number of network parameters in Table 2.14 is proportional both to their memory footprint and to their required training time. On the other hand, the number of operations required for a single forward pass, as shown in the table is capable of indicating the overall inference speed of the network. The lower the number of operations, the higher the inference speed. Here, ENet is not a CNN, but it is directly inspired by CNNs. By

²Classifiers usually assign a probability value to all of their output classes. Thereafter, the class with the highest probability (top-1) will be considered as the final answer, which is not always true. Sometimes the correct answer is among the top-N classes. Using top-5 accuracy is common, when comparing different classifiers.

³A given image may have four corner crops and one central crop.

Table 2.14: Comparison of a selected number of CNN architectures submitted to the Annual ImageNet Challenge [14, 15]

Architecture Name	Top-1 Accuracy	No. of Parameters	No. of Operations
AlexNet [215]	> 56%	≈ 50 M	≈ 2 G
NIN [216]	> 62%	≈ 6 M	≈ 3 G
VGGNet-19 [217]	> 70%	≈ 150 M	≈ 39 G
Inception-v4 [213]	> 80%	≈ 35 M	≈ 18 G
ResNet-152 [214]	> 77%	≈ 50 M	≈ 23 G
ENet [218]	> 67%	≈ 5 M	≈ 2 G

the same token, Network in Network (NIN) is not a CNN, but it relies on convolution operations. Some of these architectures are also included in the deep learning libraries of Section 2.4.2 and can be readily involved for any project. Let us now extend our discussions to cover deep NNs in dynamic systems.

Deep NNs for dynamic IoT data

Recurrent Neural Network (RNN) [13, 219] and its variants (i.e. long short-term memory, gated recurrent unit, etc.) constitute the only deep NN architecture in Table 2.9 that can be used for nonlinear dynamic systems, as exemplified by continuous underwater sensor outputs. RNNs are capable of using both time-series and sequential data streams to construct supervised classifiers.

RNNs constitute a deep version of Hidden Markov Models (HMMs) [219], which represent a simple form of the broader family of dynamic Bayesian networks. Markov models (i.e. Markov chains) are stochastic models, in which the output of a NN in response to the current state, only depends on the output of certain selected neurons in the previous state. The values of all neurons in all states are visible and are considered as the outputs.

Hidden Markov models are similar to Markov models, but they rely on non-observable or hidden neurons. In these networks, the visible output is directly calculated by applying a nonlinear function to the output of the hidden neurons. As illustrated in Table 2.9, an RNN is constructed by stacking HMMs on top of each other. The rectangular shape of the nodes in this picture indicates that in contrast to the circles of the other networks, these nodes are not constituted by a single neuron, but rather they are a combination of neurons in the form of a HMM. After studying the DNNs suitable for static and dynamic data

analysis, let us study their capabilities to solve real-life problems in marine environment.

ML solutions in underwater image applications

The concepts of image and video quality assessment as well as image restoration were studied previously in Section 2.4.3. We have also stated the fact that underwater imaging applications heavily rely on image enhancement algorithms to cope with the destructive effects of visible-light environments. To address this problem, deep learning techniques have been employed for enhancing images suffering from noise, absorption, scattering, and color distortion effects [162, 220, 221]. Furthermore, the benefits of DNNs in underwater image quality assessment can be assessed in future studies with reference to their overwater counterparts [222]. These applications, while in their infancy, are rapidly expanding.

Other applications that can benefit from deep learning are underwater image clustering and binning. We briefly reviewed these concepts in Section 2.4.3, noting that some underwater clustering algorithms are also introduced in Table 2.11. By the way, using novel deep learning techniques in underwater image clustering has not as yet attracted the attention it deserves. For instance, one can beneficially exploit the embedded image clustering technique of [223] as an unsupervised DNN methodology in underwater image segmentation.

Additionally, using ML techniques in underwater object recognition have been previously used for various underwater applications and have shown different performance. Here, we provide a summary of these techniques applied to some common underwater object of interest recognition and compare their performance. In order to have a fair comparison, the Average Precision (AP) parameter, which is a widely accepted performance measure, can be used. The average precision in any statistical or ML-based classifier can be calculated as [40],

$$\text{AP} \triangleq \frac{1}{N_C} \sum_{i=1}^{N_C} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \quad (2.19)$$

where the True Positive (TP) and False Positive (FP) values are calculated for N_C number of classes. The AP parameter of many ML techniques published in the underwater plant, fish, and coral recognition literature are compared in Table 2.15. All the algorithms referenced in this table are based on the well-known Fish4Knowledge and EILAT datasets introduced in Table 2.7 and are also based on the video footages recorded by AUV and ROV vehicles.

It is quite common in ML to cascade different algorithms into a single method to attain an improved performance (e.g. decision tree with SVM, and CNN with fully-connected).

Table 2.15: Comparing the Precision of Multiple Machine Learning Techniques to Recognize Three Different Underwater Species

Application	Fish Recognition												
Dataset	Fish4Knowledge Dataset [40, 41, 181, 182, 224]												
Methods	Regression Tree	LeNet	LDA and SVM	Flat SVM	BEOTR	DT and SVM	Softmax	KNN	DeepFish	GoogLeNet	VGGNet	DeepFish and SVM	AlexNet
Average Precision (%)	75.4	76.8	80.1	82.9	84.5	84.6	87.6	89.8	90.1	93.9	96.6	98.2	99.7
Application	Plant Recognition							Coral Recognition					
Dataset	ROV and AUV Video [187, 191]							EILAT Dataset [111, 225]					
Methods	Self-Organizing Map	LVQ	Decision Tree	KNN	Random Forest	Fully-Connected	SVM	Decision Tree	Bayes	KNN	Random Forest	Fully-Connected	SVM
Average Precision (%)	93.8	94.1	95.1	95.3	95.8	96.1	96.1	69.6	82.1	83.3	84.6	85.2	90.8

As an instance of algorithm merging in underwater applications, Faillettaz *et al.* [51] as well as Hu and Davis [176] cascaded a fully-connected classifier over shape-based feature-sets with an SVM classifier over texture-based feature-sets. By combining the results from these two classifiers, they claimed to have achieved an improved average precision, as defined in (2.19).

Another noteworthy cascaded solution was conceived by Schoening *et al.* [174], where they employ multiple cascaded SVM binary detectors to construct a deep sea megafauna recognizer. The binary SVM detectors have object-specific operation, which makes them more accurate. Accordingly, by combining these binary detectors, high precision multi-object recognition was achieved. However, combining those object-specific binary detectors requires more hardware resources than a single classifier.

Table 2.16: Comparing the Precision of Multiple Machine Learning Techniques to Recognize Objects in Sonar Imagery

Application	Object Recognition				
Dataset	Echoscope Sonar Image Dataset [227]				
Methods	K-Nearest Neighbors	Multilayer Perceptron	SVM	CNN (AlexNet)	CNN (LeNet)
Average Precision (%)	72.0	89.3	92.7	94.1	97.0

Finally, it is worth mentioning that underwater imaging applications are not limited at all to the visible light domain. For example, undersea sonar imagery can provide high-resolution images of the seabed, even in turbid water with low visibility. Some of the reported applications of sonar images include ocean mapping, mine-countermeasures, oil prospecting, and underwater search and rescue (e.g. finding the drowned corpses, wrecks, and airplanes) [226]. In this regard, relying on human operators in sonar-based underwater object recognition applications is not recommended, since they will experience fatigue by staring at the display screen, and they might consequently miss the object of interest. Therefore, intelligent image recognition methods can be trained to replace human operators, especially in long-duration search scenarios.

To highlight the accuracy of machine learning techniques in underwater sonar-based object recognition, the AP metric of (2.19) is used here in Table 2.16. This table compares the performance of deep NNs to that of several statistical methods as well as to that of traditional NNs. While the statistical and traditional networks seen in this table are trained using HOG features from Table 2.12, CNN deep networks do not need any feature extraction based preprocessing. Additionally, as seen in this table, appropriately designed and well-trained DNNs may outperform traditional models.

ML solutions in underwater video applications

As already discussed in Section 2.4.5, the capability of hardware platforms to train and analyze DNNs was impressively improved recently. This improvement has attracted increasing attention to the subject area of real-time IoUT video applications. It was also pointed out in Section 2.3.2 that the rapid growth in underwater video data volumes will require the development of automatic video processing, which can be carried out by machine learning techniques. These automatic solutions will be used in a variety of

underwater video data processing applications such as:

- *Visible light video tracking*: This application is designed for scanning video sequences to follow a specific element of interest [129]. A pair of basic problems in a video object tracking solution is how to predict the location of a moving element in the next frame and how to detect the element within this predicted region. Both of these can be handled by deep NNs [117]. This application was also discussed in more detail in Section 2.3.4.
- *Sonar video tracking*: Sonar videos are recorded with the aid of acoustic waves, instead of visible light. Within turbid undersea environments, sonar systems offer a significant advantage by supporting long-range and low data-rate imaging [118]. Similar to visible light video object tracking, sonar-based video object tracking can also be carried out with the aid of deep NNs [228].
- *Underwater photo mosaicing*: This is the act of combining separate visible light or sonar video frames, for capturing a wider perspective of the region of interest. Machine learning based photo mosaicing of underwater images is now routinely performed by state-of-the-art underwater vehicles both for exploration and for navigation [46,54,229]. Automatic localization and positioning of submersible vehicles with the aid of photo mosaicing is termed as SLAM [151] and it is studied in Section 2.3.4.
- *Marine life studies*: The analysis of underwater species is an indispensable part of any observatory video system. Thanks to the wide availability of machine learning based data processing toolsets, marine biologists are now capable of analyzing the high-volume video data captured for extracting the desired information. Scientific studies based on marine life video data applications have been published in different areas, including underwater species behavior understanding (ethology) [116,185,230], abundance and counting [48, 127, 171, 231], size measurement [127], detection and recognition [40,41,171,182,184,231], and tracking [47,117,170,171]. We have covered some of these aspects of marine life analysis in Section 2.3.4.

ML solutions in underwater sensor applications

Despite the unique benefits of applying RNNs to the processing of sensor outputs and in nonlinear dynamic systems control, there is a paucity of reported use cases in real-world IoUT applications. Their underwater applications are limited to a few scenarios, such as IoUT sensor data forecasting [17], underwater vehicle sensor read and fault diagnosis [232], and the dynamic control of underwater movements [233].

Table 2.17: Comparing the Mean Squared Error of ML Techniques in Sea Surface Temperature Prediction at Philippine Sea

Application	Sensor Timeseries Forecasting				
Dataset	NOAA Dataset in Table 2.6				
Methods	SVM [236]	CFCC-LSTM [236]	SVR [16]	FC-LSTM [16]	GRU-ED [16]
1 Day Forecasting	0.434	0.166	0.095	0.061	0.063
1 Week Forecasting	N/A	N/A	0.214	0.168	0.162
1 Month Forecasting	1.478	1.145	0.212	0.343	0.207

Among these applications, RNN-based predictive models conceived for IoUT sensory data forecasting as well as for missed sensory data implantation are better investigated [234]. More specifically, sea water temperature and salinity predictions are claimed to be important, because:

- Water temperature and salinity have a direct effect on the acoustic communications between IoUT nodes [235],
- Oceanic temperature has a substantial impact on both the land and the marine ecosystems by regulating the global climate [17].

Hence, the employment of machine learning techniques to design accurate predictive models is promising in BMD. For example, the influence of Long Short-Term Memory (LSTM) [236] and Gated Recurrent Unit (GRU) [16], as a pair of common variants of RNNs on highly accurate water temperature prediction is presented in Table 2.17. Even though sea surface temperature forecasting is challenging due to the influence of numerous complex and nonlinear thermodynamic factors, data-driven DNNs are capable of learning these dynamic behaviors. By comparing RNN-powered networks to traditional machine learning solutions in Table 2.17, the efficiency benefit of DNNs in IoUT sensor data prediction becomes explicit.

To recap, the current usage of RNN and its variants in IoUT applications are rare. Nonetheless, their versatile applications in IoT in smart homes [237], smart cities [238], weather forecasting [239], and other areas promise a similar growth for IoUT applications in the near future.

2.4.7 Section Summary

In this section, the employment of ML-based techniques in BMD processing was studied. We started this section by the definition of machine learning, classifying it into: classic statistical methods, traditional neural networks, and modern (i.e. deep) neural networks. The associated chronological perspective was also presented. Then, we discussed NNs by surveying the major deep networks in the literature and their potential applications both in static and dynamic underwater data processing. Despite the rapid development of sophisticated, but complex deep NNs, light-weight traditional NNs might remain typical in limited-complexity underwater applications. Therefore, more detailed discussions were provided in this section to cover data cleaning and feature extraction techniques in BMD. We then surveyed the available software frameworks and hardware platforms, including a collection of freely available libraries and frameworks. We also provided a comparison of the hardware infrastructures suitable for the software products discussed. Finally, the average precision and accuracy of diverse machine learning approaches suitable for underwater applications was studied.

2.5 Challenges and Future Directions in IoUT and BMD

Having reviewed the state-of-the-art research in the areas of IoUT, BMD, and machine intelligence, the challenges and opportunities in these growing fields will be discussed. We also propose solutions and future research directions to address the challenges and to pursue the opportunities.

One of the main obstacles that has been hindering further advances in the IoUT domain is that well-known terrestrial technologies, which perform well in the IoT domain, tend to be unsuitable in underwater applications. Many issues in the oversea application domain can be readily solved, easily, while they pose a significant challenge in underwater scenarios. Below we will continue with a list of challenges, opportunities, and future trends in the IoUT, BMD, and ML fields. Some of these challenges may be mitigated with the aid of big data processing and analytics, while some others require research efforts from the broader engineering community, hardware vendors, and policy makers.

2.5.1 Underwater Network Management System

Due to the significant growth in the number of Internet-connected underwater devices, the IoUT infrastructure tends to exhibit increased complexity. Consequently, improved Network Management Systems (NMS) are required, which represent the process of monitoring and controlling every aspects of the underlying network, for ensuring its seamless

operation [240]. The monitoring must be automatic and prompt in locating, measuring, and reporting faults. Additionally, their control should be capable of efficient and reliable resource allocation or troubleshooting [61].

While the concepts of network management in IoUT are somewhat similar to those in IoT, the methodologies are different, as discussed in Section 2.2.4 owing to the differences between overwater and underwater communication channels. Hence, the extension of NMS in IoT to Underwater NMS (U-NMS) in IoUT requires further research [241] for each of the six aspects of U-NMS in the FCAPSC model, namely the Fault-, Configuration-, Account-, Performance-, Security-, and Constraint-managements [241]. It is worth mentioning that FCAPSC of IoUT was derived from the original five elements in FCAPS of IoT, which was introduced by the International Organization for Standardization (ISO) in the 1990s. The constraint-management element requires U-NMS to deliver continuous connectivity even in the face of the hostile underwater channel, node mobility, device fragility, environmental dynamics, and technological heterogeneity.

To facilitate the implementation of the FCAPSC model, U-NMS protocols tend to divide their influence domains into the family of network functionalities (e.g. routing management, protocol assignment, security checks, etc.) and of device operations (e.g. UWSN maintenance, energy conservation, device positioning, time synchronization, etc.) [61]. The relevant studies around these domains were comprehensively surveyed in Section 2.2. To the best of our knowledge, the Underwater Simple Network Management Protocol (U-SNMP) is the only U-NMS protocol, which covers the first domain of influence (i.e. the network functionalities). U-SNMP in IoUT is again, an extension of SNMP in IoT, and it is a manager-agent-based protocol, which is used in communication between devices [241]. On the other hand, Lightweight Machine to Machine (LWM2M) is an IoT protocol that covers the second domain of influence (i.e. the device operations). LWM2M is applicable to IoUT as well, subject to some modest adjustments [61].

Despite its popularity, U-SNMP lacks facilities for network configuration, and LWM2M suffers from challenges owing to the associated heterogeneous network support [61]. By contrast, there are other network management protocols in IoT that offer better performance. Future studies might consider appropriately adopting those overwater protocols to underwater applications. For example, the Common Management Information Protocol (CMIP) offers better security features and it is suitable for wide area networks [242]. Another alternative might be the LoWPAN Network Management Protocol (LNMP) in IoT, which has low data rates, low power consumption, low cost, and supports flexible topologies [243].

While the above protocols are better suited for distributed network managements, the centralized Software-Defined Network (SDN) management techniques are also in need for

underwater management systems. Here, the Open vSwitch Database (OVSDB) protocol at overwater SDNs constitutes a promising base for designing its underwater counterpart. However, any attempt to adopt OVSDB to underwater SDNs requires further research for addressing its current security issues [244].

Even after dividing the U-NMS responsibilities into two major domains of influence and limiting the U-NMS protocols correspondingly, the number of challenges in each domain will remain significant [61]. This is a direct consequence of the broad nature of the U-NMS topic itself. To better understand these challenges and study their future directions, we have discussed each challenge in a dedicated subsection. The following subsections will cover multiple aspects of U-NMS, ranging from energy conservation to device maintenance, to its security issues and communications. Some of these subsections discuss using BMD analytics as well as powerful deep learning techniques.

2.5.2 Energy Conservation and Harvesting in IoUT Devices

Energy conservation and harvesting in IoUT devices are of prime concern in almost all underwater applications, while they can be readily addressed in overwater scenarios, where energy can be harvested from the sun and preserved in the system [245]. Some innovative methods of gleaning energy in underwater environments include:

- *Overwater solar energy*: As already discussed in Section 2.2.6, by performing computations in above-water edge-devices, such as surface-floated buoys and vehicles, solar energy can be harvested. Indeed, solar powered buoys are amongst the oldest methods of environmental energy harvesting techniques [246, 247].
- *Underwater solar energy*: It was stated in Section 2.2.2 that light is strongly absorbed in water and it additionally suffers from color distortion. However, as mentioned in Section 2.3.2, blue light will penetrate water deeper than other visible light frequencies. Some previous studies suggested the use of solar cells to harvest the blue frequency band of the solar energy underwater. These photovoltaics are claimed to provide useful power at the depth of more than 9 m [248]. However, many underwater applications are invoked at depths well beyond the light penetration domain.
- *Tidal-wave energy*: Harvesting kinetic energy from waves in the littoral tidal basin can be readily achieved by using today's technology. This can also offer a source of energy in the underwater benthic zone. These systems are based on piezoelectric elements and the energy generated is high enough to power UWSNs and their devices [249]. Similarly, some contributions report the employment of the same technique to harvest energy from fluid-flow in pipelines [250].

- *Wireless energy transfer*: This method was introduced for RFIDs and acoustic tags in Section 2.3.4. Additionally, a remotely powered acoustic UWSN was reported in [251] as another energy acquisition alternative for the IoUT. Thereby, sensor nodes harvest the mechanical wave power supplied by an external acoustic source. Another use case of wireless energy transfer may rely on in-situ magnetic charging stations, where underwater vehicles (e.g. AUV, ROV, etc.) can use these stations to recharge their batteries [252].
- *Wired energy transfer*: In a clear contrast to the previous items, every close-to-shore IoUT application may rely on energy transmission through a cabled network. In this method, the energy arrive from a solar, wind, or urban power network and it is directly transferred to the UWSN. These systems, however, are costly because of the cabled infrastructures on both land and at sea [4].

The above methods have the potential to be used in marine type projects. They can extend the lifetime of IoUT networks and boost their QoS. However, except for the *wired energy transfer* method, they tend to be unpredictable, hence none of them guarantees the uninterrupted delivery of energy. To address this issue, the following pair of solutions may assist:

- *Using rechargeable batteries*: All the aforementioned methods can be accompanied by tandem batteries to store energy. These batteries are recharged during the instances of energy acquisition and deliver their stored energy afterwards. This solution obviously prolongs the sensor network's lifetime, but requires maintenance and increases the cost of the system [4].
- *Managing energy consumption*: Just like any other electrical grid worldwide, the IoUT has to manage its energy demands by optimizing the power allocation to guarantee uninterrupted data collection and transmission [253]. Indeed, all components of the IoUT should be energy-aware. In this context, an innovative energy-aware robot was proposed by Wu *et al.* [254], which had the shape of a killer-whale. This robot has had reduced energy consumption for its propulsion as a benefit of its excellent lift-to-drag ratio, which is important for effortless gliding in water. Compared to the Seaglidors introduced in Table 2.5, the controllable flukes of this robot offer substantial energy savings, better maneuverability as well as enhanced endurance.

2.5.3 Development of Low-cost and Affordable Sensors

The underwater sensors listed in Table 2.2 and multi-sensor buoys are usually very costly compared to their overwater counterparts. To address this multi-disciplinary challenge,

the following approaches can be considered:

- *Quality vs. cost trade-off*: Low-cost sensing devices having lower precision measurements could be purchased to strike a quality vs. cost trade-off.
- *Transferring specimen for inland assessment*: The establishment and operation of an underwater in-situ sensing unit is generally more expensive than a laboratory-based experiment. Therefore, whenever possible, samples could be transferred to an inland lab, to avoid the need for costly in-situ processing and evaluation.
- *Inferential measurements*: It is a common technique in industrial instrumentation to estimate a parameter from the values of other parameters, which are easier to measure. For example, calculating the total amount of ions dissolved in water is always easier by measuring its electrical resistance instead of utilizing costly electrochemical sensors [255].

In addition to the above-mentioned general recommendations, designing specific cost-effective sensors can help the evolution of IoUT technology. In order to design such sensors, scientists from different research backgrounds have to cooperate. The result of this cooperation will be *ad hoc* solutions, which are tailored to the predefined need of any project. For instance, here we list several contributions involving low-cost underwater sensors.

- Islam *et al.* [256] proposed a low-profile and low-cost microstrip patch antenna to measure the salinity of water. They found that the antenna's reflection coefficient is proportional to the amount of salt or sugar dissolved in water.
- Vorathin *et al.* [257] constructed a high-resolution hydrostatic pressure and depth sensor by attaching a fiber Bragg grating on a rubber diaphragm. Their sensor is claimed to enhance the sensitivity and to compensate the temperature effects.
- Wang *et al.* [258] conceived a low-cost turbidity sensor, based on the 90° scattered light detection principles. To elaborate, they used off-the-shelf infrared LEDs having controlled light emission to construct a low-cost, yet accurate product.
- Kirkey *et al.* [259] proposed an inexpensive fluorometer based on an optical backscatter transducer. Explicitly, their idea is to use low-frequency circuitry for modulating the light source. Using this technique, their product will be very cost-effective.
- By relying on the fundamental concepts of Time-Domain Reflectometry (TDR), Time-Domain Transmissometry (TDT), or Fiber Bragg Grating (FBG) in optical

physics, optical fibers can act as a sensor to detect a wide range of underwater physical parameters. Examples include leak detection in pipelines and estimating its location [260], stress response of the offshore platforms (i.e. legs of the jacket structure) and detecting its deformation [261], bending moments of the flexible risers in a hang-off position to avoid exceeding its absolute maximum ratings [261], temperature and pressure measurement as well as eliminating the temperature-pressure cross-sensitivity [262], etc.

In addition to the above techniques of reducing the sensors costs, reducing the physical size of underwater sensors also tends to reduce their production cost. Examples of low-cost smaller sensors include miniature underwater robots [263] and low-power nano-sensors [105].

Furthermore, using the edge computing capabilities discussed in Section 2.2.6, it is possible to establish a laboratory on board of an oceanic exploration platform [60]. This limited-capability on-board lab may conduct preliminary experiments on samples, before transmitting the numerical results through the web. Additionally, by using a video-empowered command and control system, there would be no need for an expert to be present in-situ. This laboratory on the edge will also eliminate the need for sending the specimen to an inland lab for evaluation, which would be both expensive and time-consuming.

2.5.4 Large-scale IoUT Underwater Communications

In terrestrial telecommunication, electromagnetic waves, copper cables, and optical fibers are the mainstream transmission media. However, as studied in Sections 2.2.2, fiber-optics are expensive to deploy and maintain under water, hence typically acoustic, electromagnetic, and optical technologies are deployed, which do not propagate well, hence making IoUT telecommunication challenging.

Electromagnetic and optical technologies only cover short communication distances and are therefore unsuitable for long-range IoUT communications. Acoustic technologies tend to be more amenable to long-range IoUT networks, but they have a narrow frequency bandwidth and are prone to cross-talk with other local acoustic applications [264]. Hence they are also unsuitable for large-scale networks.

A promising, but costly technique of addressing the communication challenge in IoUT is to combine heterogeneous communication technologies [265]. However, the design of multi-technology multi-mode gateways for undersea applications is a challenging task, especially when considering the energy harvesting difficulties. These gateways may be combined with the SDN and cognitive radio concepts for efficiently sharing the limited spectrum undersea [265]. Future research should conceive energy-aware software and low-

power hardware solutions for these gateways to improve the quality of communication in IoUT networks.

Another promising solution to underwater communication problems, especially in the case of long-range inter-continental telecommunications, is the inter-connection of the IoUT and aeronautical technology. In this technique, the data in the application layer of underwater networks that is produced by sensors and imagery equipment can be transferred to the overwater buoys and to the floating ships. Therefrom, data will be handed to the low-earth orbit or medium earth orbit communication satellites, or to the aerial vehicles. A similar example of this technique is provided by NCEI, Landsat, [Aquarius](#), SARAL, [CryoSat](#), [Jason](#), [HY2-A](#), and IRS satellites in Table 2.6 for remote sensing of ocean surface parameters. Another example is offered by [92] for the connectivity of unmanned aerial vehicles and IoUT.

Additionally, reducing the data volume to make it suitable for transmission using narrow-bandwidth acoustic technology could be used for mitigating the communication challenges in large-scale IoUT networks. This was discussed under the concept of MCC and MEC edge computing paradigms in Section 2.2.6. According to this method, edge-processing is capable of reducing the volume of raw data [90]. This consequently reduces the bandwidth requirement.

In this regard, the combination of the MEC paradigm with smart unmanned vehicles can offer an alternative solution to the problem of long-distance underwater communications. In this solution, autonomous underwater vehicles [91], unmanned aerial vehicles [92], etc. can be used for IoUT data collection. Here, edge computing can undertake some essential computations, so that big data collection will be mitigated, while latency-sensitive situations are handled promptly [92]. However, the lack of energy in undersea environments, makes the employment of MEC a challenge. In closing we note that MEC has a similar architecture to the previously suggested inter-connection between the IoT and IoUT.

2.5.5 Dynamic IoUT Signal Routing and Traffic Control

The underwater propagation environment is quite hostile; hence it is of low channel capacity, which makes even point-to-point single-link data transmission challenging in the IoUT. This becomes even more challenging when a network of concurrently communicating nodes is considered. For example, in the presence of a network supporting multiple transmitters, the tele-traffic escalates, and traffic control becomes a challenging task, when aiming for a reasonable QoS [29].

To avoid any tele-traffic congestion, efficient routing management is crucial. To elaborate a little further, in *ad hoc* networks the design-dilemma is whether to use more short

hops at the cost of an increased delay or fewer longer hops. It is beneficial to use low-complexity non-coherent transceiver techniques and take into account the battery-charge during routing.

Additionally, intelligent traffic control systems, using both the deep learning approaches [266] and the SDNs [267, 268] may be devised in underwater applications. These systems are capable of efficiently handling concurrent data transfers to avoid congestion.

To employ deep learning in traffic control systems, we first have to define the action space. To do so, consider a heterogeneous IoUT network constructed of both wired and wireless connections. As detailed in Section 2.2.3, all the nodes of such a network may connect with one another, using multiple hops. As a consequent of both the limited underwater transmission range [269] or the large-scale infrastructural size of the network [265], the number of possible hops escalates, leading to numerous potential paths for a data packet to travel between a pair of nodes. In this context, every path can be considered as an action in the action space. After taking an action, the system's feedback (a.k.a. reward) can be quantified by the traffic load level of the nodes in the following timeslot, which is formatted as an award value matrix. The combination of the actions and their consequent rewards provides the required training data for semi-supervised deep reinforcement learning, as advocated in [266]. To the best of the authors' knowledge, such datasets for IoT, do not exist for IoUT. Once collected, these datasets of IoUT routing management and traffic control can then be used to train ML-based models. After successfully passing the training phase, this deep model can be used to infer the best overall path for a given data packet.

Although the above DNN can be conveniently used in most of the IoUT structures, it might falter as the IoUT architecture gets wider. By expanding the network scale, the number of possible paths will increase exponentially, and the deep NN can no longer learn the patterns in the data flow. To address this problem, Fadlullah *et al.* [266] have proposed a solution, which can also be employed in underwater applications. To elaborate a little further, they suggested to:

1. Change the action space from the entire set of all path combinations to simply the next hop destination;
2. Replace the semi-supervised reinforcement learning by a cascaded combination of supervised CNN and DBN;
3. 3) Predict the award value matrix (by CNN) before deciding on the best action (by DBN).

These recommendations for using DNNs for wide-scale IoUT traffic control will reduce

the packet loss rate as well as improve the network throughput. As a result, the routing performance will improve compared to that of the conventional methods.

The other technique of improving the network throughput is that of adhering to the SDN methodology. This type of network, which relies on a centralized control system, for monitoring the network's traffic flow. According to Xie *et al.* [267], the centralized management of SDNs substantially benefits from using DNNs. For example, the functions of routing optimization, traffic prediction, path load prediction, node deployment optimization, delay prediction, QoS prediction, content delivery optimization, resource allocation, SDN reconfiguration, optimized spectrum sharing, number of active nodes estimation, and intrusion detection would all benefit from using DNNs in overwater SDNs. Hence it is promising to critically appraise their synergies also in underwater applications.

2.5.6 Deleterious Effects of Imprecise Channel Modeling

Naturally, the channel plays an essential role in designing the underwater deployment of endpoint nodes, relay nodes, and sinks. The strategic deployment of nodes is capable of increasing the entire system's battery life and improving the QoS. Almost all of the underwater acoustic, electromagnetic, and optical channel models as described in Sections 2.2.2 and 2.2.4 and also the underwater magnetic induction channel model of [270] rely on approximations to simplify the overall model. For example, many channel models assume straight signal propagation undersea, which simplifies numerical calculations [34]. However, approximations and idealized simplifying assumptions will result in imprecise models and inaccurate communication.

Imprecise channel modeling can also lead to inaccurate simulation-based modeling of underwater communications. Below, we provide a list of recent publications, which showed the impact of imperfect channel modelling in the simulations of underwater communication, and therefore used precise channel models for their simulations.

- Using a preamble alerts the receiver about the reception of an incoming data burst and switches it from its low-power dormant mode to its high-power active mode. Therefore, both missing the detection of a preamble and declaring its reception, when it is actually absent reduces the receiver's battery life. A beneficial preamble detection method was proposed by Li *et al.* [85], for an underwater digital communication system, which coexists with other deployed networks. Using an accurate channel model is critical in this identification method in order to prevent a receiver from being triggered by other systems and consequently extends the underwater battery life.

- IoUT communication is affected by numerous signal impairments, such as a high propagation delay and high signal attenuation. These reduce the link reliability, which can be mitigated by sophisticated Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ) techniques. However, at low SNRs the throughput may be reduced by excessive data retransmission. Liang *et al.* [86] optimized the overall transmission redundancy to be used in UWSNs by relying on an accurate channel model.
- In the simulation-based investigation of adaptive transmission in time varying underwater acoustic channels conducted by Wang *et al.* [87], the transmitter's data queue length and the predicted channel conditions were relied upon for designing the adaptive transmitter parameter values. Although their method relies on reinforcement learning to yield considerable energy saving, it cannot correctly learn the accurate parameter values without a precise channel model.

To address the need for precise channel modeling, further research is required for devising precise yet computationally efficient models, for ensuring that the simulation of underwater communications in context of the emerging IoUT is as reliable as possible. This would mitigate the need for and the likelihood of future network upgrades.

2.5.7 Sparse and High-maintenance Sensing Devices in IoUT

In contrast to the IoT, the sensing devices of the IoUT are sparsely deployed and exposed to severe environmental effects [32]. The sparse configuration and the harsh environmental conditions make the maintenance of the IoUT, costly. Explicitly, maintenance should mitigate the effects of erosion, corrosion, sediments, pollutions, and other phenomena imposed by seawater.

To address these issues and to reduce the maintenance cost of sparse high-maintenance nodes in the IoUT, a compelling solution is to incorporate self-management capability [271], including self-evaluation, self-configuration, and automatic reports to human operators. Therefore, developing intelligent ML-based hardware nodes for the IoUT, which have a self-management and decision-making capability and conceiving their required software are promising avenues for research in IoUT, which need further attention.

2.5.8 Poor Underwater Positioning and Navigation

As mentioned earlier, GPS signals do not penetrate the sea, hence other navigation techniques have to be used. A number of navigation methods such as blind positioning, acoustic transponders, ranging sonars, image-based positioning, and SLAM were introduced in

Section 2.3.4. However, none of these stand-alone techniques offer a non-accumulating positioning error, therefore, none of them are adequate [138].

Hence, further research is required for improving underwater navigation systems. The challenges to address are:

- Selecting and combining the large amount of data from the aforementioned stand-alone navigation techniques is a challenging task, while considering the system's cost and complexity as well as precision. A research opportunity to address this issue is to study different combinations of navigation techniques, while striking a trade-off between cost and accuracy. For instance, Bonin-Font *et al.* [46] have combined image-based positioning with SLAM to achieve improved navigation. Another study has combined blind positioning with a long-baseline acoustic transponder to reduce the positioning errors caused by acoustic ray bending and variable sound velocities [139].
- Preliminary environmental survey for acquiring offline data is a challenging mandatory step, which has to be conducted before scene analysis based localization techniques (i.e. image-based positioning systems as well as acoustic transponders). This step is required for extracting positional fingerprints (i.e. features), which will be subsequently used for accurately training a localization ML algorithm. However, gathering these big datasets to train ML algorithms is not trivial in underwater environments.

Furthermore, traditional ML models can be easily misled by any variation in the underlying high-dynamic underwater scenes. To address this, using deep NNs having automatic feature extraction capability is highly recommended. The benefits of diverse statistical methods, traditional NNs, and deep NNs in scene-based indoor positioning systems are reported in [272]. These quick lines can be adapted for underwater applications.

- Finding and implementing new natural phenomena for improving underwater positioning as well as navigation is another challenge. In this context, a gravity-aided navigation system was conceived in [140], which is based on exploiting the difference between the observed and the predicted gravity. These methods however, suffer from biases and error accumulation, which has to be addressed in future research.
- Adapting the existing Low-Power Wide Area Network (LPWAN) technologies to carry out IoUT localization is a potential opportunity. In a review article of Zafari *et al.* [272], a collection of these wireless technologies (i.e. SigFox, LoRaWAN, Weightless, etc.) was studied in IoT localization. Using the same approach would

be beneficial for IoUT, as they all consume extremely low energy and operate in a wide reception range.

2.5.9 Non-Destructive Testing in Underwater Applications

Non-destructive testing (NDT) is the process of inspecting a system, device, or component, without imposing any changes on its shape or material. NDT can be performed while the Device Under Test (DUT) continues its normal operation. For example, both the TDR and TDT methods discussed in Section 2.5.3, are variants of NDT methodologies relying on optical fibers, which are undertaken without interrupting the normal operation of their DUT [260].

Apart from optical fibers, NDT can also be carried out by other equipment. Some of these NDT tools include visible light cameras, electromagnetic flux detectors, ultrasonic transceivers, and magnetic inductors (i.e. eddy current sensors). Using these equipment in underwater NDT are reported in many diverse applications, such as:

- Leakage detection [260], vibration recognition [273], and non-stationary disturbances as well as strain sensing [274] in pipelines, by fiber-optics;
- Bending and deformation inspection in flexible risers, by fiber-optics [261];
- Temperature and pressure monitoring in downhole tools, by fiber-optics [262];
- Determining water-level in unmanned water resource management systems, by visual cameras [275];
- Inspecting the outer surface of large ship hulls, by electromagnetic flux detectors [276];
- High-sensitivity hydrophones for opto-acoustic imaging, by ultrasonic [277];
- Welding inspection and defect characterization in offshore platforms, by magnetic induction [278].

Developing a classic NDT algorithm for a typical component is not an easy task, requiring knowledge about both the DUT and the underlying physics of the NDT itself (i.e. ultrasonic, electromagnetic, or optical wave scattering). Nevertheless, in the modern age of ML techniques, one can gather a big dataset from any DUT. These data samples can then be fed to a deep NN for training purposes. By implementing this deep learning approach, developing NDT algorithms can be carried out faster, with minimal knowledge about the underlying physical concepts [273–275].

Another opportunity in underwater NDT is to use the recent developments in the field of distributed and cloud-based BMD Processing tools, which has been discussed in Section 2.3.3. Relying on the frameworks listed in Table 2.8, one can readily glean data from multiple independent NDT equipment, and then employ a data fusion technique from Section 2.3.1, to combine and process the gathered big data. It is proven by Bayes' theorem that the uncertainty in the final test result will dramatically decrease by fusing data from independent data sources [278].

2.5.10 Lack of Strong Data Leveraging Tools

The processing of BMD requires powerful hardware and software tools that can automatically extract knowledge from large databases. Some of these hardware tools were introduced throughout this chapter, specially in Sections 2.3.3 and 2.4.5. However, more advanced software techniques are required for automatic long-term data-gathering and data-monitoring applications. For instance, even though unmanned auto-annotation based industrial software is in very high demand for classification and labeling underwater objects, plants, or creatures, the existing marine image-annotation software packages are only semi-automated, at best [136].

The shortages in automatic data leveraging techniques is partially due to the uncertainty in selecting feature-set. Ambiguity in feature selection and in the ensuing feature reduction is a consistent challenge in automated ML-aided projects. In a conventional neural network, there are a variety of features and descriptors, hence selecting the most useful ones is challenging. These diverse underwater feature-sets have previously been discussed in Section 2.4.4 and summarized in Table 2.12. Recall that relying on a single feature is usually inadequate for accurate underwater classification and clustering [182].

Although modern deep learning approaches (e.g. fully-connected, CNN, Autoencoder, etc.) are promising in terms of overcoming this challenge, the advantages of classical ML methods may nonetheless provide better results. These benefits of classical ML methods include having fewer parameters, more rapid convergence during their training, better insights into the tangible physical interpretation of their operation, and much easier debugging as well as tuning the network. Further research is required for developing techniques and algorithms to infer useful features from a dataset, or even to automate the feature selection process. This will be invaluable for the ML community, because feature selection has a direct impact on the performance of ML-based solutions.

2.5.11 Training Deep Networks

When using deep networks, the feature extraction will be automatically handled by the hidden layers of the network. Despite this automatic feature extraction, the deep networks have the disadvantage of requiring large amounts of data for tuning their weights and biases, during the training step. The deeper and wider these networks become, the more useful features can be extracted [279]. Ultimately, we have to strike a trade-off between the training data volume required, the network size/power and its overall performance.

To address this trade-off, a pair of general solutions could be offered. The first solution is to satisfy the deep network's hunger by more data. The extra data can be generated automatically (for example by various data augmentation techniques or by employing generative neural networks, such as the Generative Adversarial Network (GAN) [280,281]) or manually (by using web-based technologies to enlist the assistance of international experts and to produce a large volume of user-generated contents).

The second solution is to enhance the deep network's efficiency by modifying their building blocks and the inter-connection of neurons (like the convolution, pooling, and activation operations of a typical CNN [282] or pruning a deep NN to ease their operation on mobile devices [283]), in order to reduce the number of network parameters. The vibrant deep learning community will no doubt continue to improve the deep network training and tackle the challenges. This will definitely improve the IoUT applications and advance BMD analytics.

2.5.12 Degraded Underwater Images

Undersea photography is always affected by environmental factors. Although the contributions surveyed in Section 2.4.3 have addressed some of these issues, there is a lot of room for improvement. For instance, developing an imaging system capable of both real-time forward- and back-scattering elimination is critically needed.

Furthermore, the underwater image quality could be significantly improved by applying some well-known hardware methods and techniques, such as light polarization [284], multi-spectral imaging [285], or stereoscopic imaging [286]. Additional improvements may also be attained by software methods, such as wavelength compensation and color reconstruction [112,169] both in active and passive underwater photography. Finally, the new concept of image reconstruction with DL [162,220,221] has promise for future research.

2.6 Conclusion

The recent advances in IoT technology and the extension of its influence both to coastal and open sea areas has led to the proliferation of the number of Internet-connected objects both in over- and underwater applications. This technological evolution inspired the new scientific concept of IoUT constituted by marine sensors, cameras, hydrophones, etc. This concept opens many new research directions for undersea data acquisition, data communication, BMD handling, and oceanic data processing.

In this chapter, we commenced by surveying the state-of-the-art in underwater communications. Given the harsh underwater propagation environment, data communication is quite a challenging task in the IoUT. The families of advanced underwater communication models of acoustic, electromagnetic, and optical technologies were introduced and innovative solutions were proposed for increasing the overall link reliability by topology and routing optimization, security improvement, and protocol enhancements. Furthermore, the underwater channel modeling was studied along with software tools to simulate both those channel models as well as the communication protocols. Both SDN and edge computing techniques were also reviewed as a promising technique of improving underwater communications. We also reviewed the IoUT network architecture, based on the well-known 5-layer TCP/IP standard model.

Naturally, the IoUT leads to BMD generation and the associated challenges include data storage, transportation, preparation, and analysis. Because manual and semi-automatic data processing methods are no longer appropriate in the new era of the IoUT, the five system components of BMD solutions were discussed and the most recent frontier-research and a range of practical solutions were discussed for each component. These solutions covered the areas of sensor, image, and video data sources, marine geographic data, localization and tracking, open access databases, distributed data processing, and cloud-based services. A complete section was also dedicated to machine intelligence (i.e. ML and DL) and its applications to marine data processing. The most recent research articles in both the hardware and software aspects of the IoUT and BMD processing were also surveyed, along with the critical appraisal of these works. Finally, numerous open research issues and future study directions were presented to provide an insight into the prospective applications, trends, and challenges. Do join this vibrant interdisciplinary research community, valued colleagues.

Chapter 3

Sea Surface Temperature Timeseries Forecasting

The timeseries forecasting of sea surface temperature is described in this chapter. The structure of our proposed ensemble of stacked DNNs is also explained and its performance for highly-accurate sea surface temperature prediction is analysed. The results are compared with the state-of-the-art in the literature. This chapter is published in *IEEE Geoscience and Remote Sensing Letters* as

- [25] M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sea surface temperature forecasting with ensemble of stacked deep neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1502605–1502609, Aug. 2021.

3.1 Introduction

Seasonal weather forecasting and SST prediction has attracted increasing attention in scientific literature [16, 17]. Because oceans cover approximately three quarters of the surface of our planet, accurate SST prediction can provide noticeable benefits to many environmental-related studies and applications. To address the SST forecasting problem, various prediction algorithms have been introduced in the literature [287–289]. These diverse predictive models can be categorized into: (i) physics-based numerical models, (ii) classic statistical methods, (iii) traditional neural networks, and (iv) deep neural networks.

Physics-based numerical models use complex kinetic and thermodynamic equations along with exciting parameters and boundary conditions. General circulation models [287] and regional ocean modeling system [290] are commonly used methods in this category. The second category of the SST predictive models, uses classic statistical methods to create a mathematical model that embodies numerical relationship between one or more random variables. Markov model [288] and linear regression [289] are two repeatedly reported statistical methods. Traditional (a.k.a shallow) neural networks including Support Vector

Regression (SVR) [16] and wavelet neural networks [290], are the third type of methods used for SST forecasting.

The fourth category of SST predication models are Deep Neural Networks (DNN), which incorporate multiple hidden layers to extract data features and to automatically learn SST variation rules. These methods are very popular due to easy access to big and up-to-date collections of in-situ and remotely sensed SST data collected by various organizations and made publicly available. Some well-established models in this category include Fully-Connected Long Short-Term Memory (FC-LSTM) [16], Gated Recurrent Unit Encoder–Decoder (GED) [16], and Convolutional LSTM [17], which are shown to have higher accuracy compared to the first three categories [16].

The SST forecasting DNN models can be divided into timeseries and next-frame predictors. While a timeseries predictor works with spatial averaged SST [17], next-frame predictors use SST distribution matrix in an area [236]. From a functionality perspective, a next-frame predictor results in a higher mean squared error, compared to its timeseries counterparts. This is simply because the next-frame predictor needs to predict a 2D matrix, while timeseries predictors predict a single value. Consequently, various practical applications need to choose between next-frame or timeseries predictors, depending on the trade-off between spatial coverage or prediction accuracy.

Here, we propose a new SST timeseries prediction model, which consists of two stacked DNNs. While previous DNN models have only used water temperature as their input [16, 17], we also use readily available air temperature data for improving the efficiency of our model. Two correlated air and water temperature variables are fed separately to two stacked DNNs. These two networks then form an ensemble to create our highly accurate model. Simulation results show that our model outperforms the state-of-the-art, even without using the air temperature data. However, the accuracy is enhanced by introducing an ensemble node, which merges the SST and air temperature timeseries into the final prediction output.

3.2 Background and Problem Formulation

Various factors can affect SST variations. These include solar radiation, surrounding air temperature, heat exchange with atmosphere, wind speed at sea surface, evaporation, ocean internal processes, etc. [16]. Due to the diverse nature of these factors, accessing them in a desired location is not always possible. Among these factors, the air temperature is a common observation that can be found in almost any weather station worldwide. Therefore, the air temperature can be easily obtained to be used in conjunction with historical SST to devise a better SST prediction model. If the air temperature is not

Table 3.1: Temporospatial Coordinates of the Sea Areas Under Study and Their Corresponding Land Weather Stations

Location Name	Type of Timeseries	Data Source	Spatial Coordinates	Sampling Duration
Bohai Sea	Endogenous Averaged SST	NOAA PSL	117.5°- 121.5°E, 36.5°- 40.5°N	1998-2020
Dalian City	Exogenous Air Temp	NOAA NCEI	121.633°E, 38.9°N	1998-2020
South China Sea	Endogenous Averaged SST	NOAA PSL	112.5°- 119.5°E, 6.5°- 21.5°N	1998-2020
Dagupan City	Exogenous Air Temp	NOAA NCEI	120.35°E, 16.083°N	1998-2020
North Pacific Ocean	Endogenous Averaged SST	Argo	130°- 190°E, 10°- 50°N	2004-2019
Amami Island	Exogenous Air Temp	NOAA NCEI	129.5°E, 28.383°N	2004-2019

available for a desired point, it can be obtained from an iso-latitude station to act as an exogenous variable.

In order to perform precise SST forecasting, here we propose to utilize SST and air temperature, simultaneously. The SST timeseries are extracted from the National Oceanic and Atmospheric Administration (NOAA) Physical Science Laboratory (PSL) and the [Argo](#) data sources, while the air temperature timeseries come from NOAA National Centers for Environmental Information (NCEI) [291]. Geolocations of these timeseries are presented in Table 3.1. The SST values in this table are area averaged over the corresponding spatial coordinates. The choice of these referenced coordinates are in accordance with other published works [16, 17], which enable us to compare the accuracy of our approach with literature.

The bilateral relationship between the SST and air temperature has already been studied in climatological research. More specifically, SST has been used as a reliable predictor of weather anomalies [292], while air temperature is used to assist in predicting SST [293].

A close inspection of Fig. 3.1 reveals the close relationship between SST and air temperature factors in two typical geolocations of our datasets (i.e., Bohai Sea and Dalian City). However, the air temperature usually has higher dynamics compared to the SST, which results in a lower statistical correlation between them. To address this problem, a moving average window can be slid over the air temperature data sequence. This window acts as a

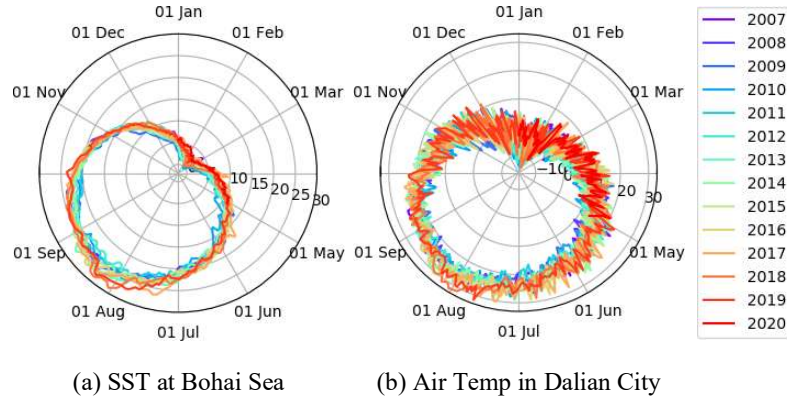


Figure 3.1: Polar plot of (a) the area-averaged SST at Bohai Sea and (b) the air temperature at its iso-latitude Dalian City, both from the NOAA data source.

low-pass filter (LPF), which smooths the high dynamics of data. This smoothing process will increase the accuracy of our proposed SST forecasting algorithm. To elaborate, by applying an LPF to the air temperature, the Pearson correlation coefficient increases to over 0.8 for all the geolocations in Table 3.1. This indicates the existence of a high linear relationship between the smoothed air temperature and SST.

In this chapter, the SST forecasting problem is represented by function $F_1(\text{SST})$, which returns a prediction value. Similarly, predicting SST from historical air temperature values can be formulated as function $F_2(\text{Air})$, which returns another SST prediction value. The predictive outputs of these two functions can then be merged to build an enhanced SST predictor, i.e.,

$$F^+(\text{SST}, \text{Air}) = w_{\text{SST}} F_1(\text{SST}) + w_{\text{text Air}} F_2(\text{Air}), \quad (3.1)$$

where w_{SST} and w_{Air} coefficients adjust the relative contributions of $F_1(\text{SST})$ and $F_2(\text{Air})$ in the final prediction.

It is worth mentioning that the SST observations for the North Pacific Ocean in the Argo data source, along with its corresponding air temperature reads (i.e., the air temperature from the Amami Island) are monthly averaged throughout this chapter. Consequently, any further smoothing is unnecessary for this geolocation.

3.3 Network Architecture

The high-level block diagram of the proposed voting ensemble of stacked DNNs is shown in Fig. 3.2(a). This model consists of two separate stacked DNN branches that are trained with different datasets. The first stage of the top branch ($F_1(\text{SST})$) takes the SST time-series as input and performs general preprocessings e.g. outlier detection and missed data

interpolation. It then passes the clean data to the seasonality decomposition block. This block decomposes the year seasonality from its raw input timeseries, outputting trend and residual.

Trend is a straight line that matches as closely as possible to the original timeseries. It can be easily found by linear regression. Year seasonality on the other hand, is the repeating one-year-long cycle in data. It can be calculated by yearly averaging SST, after subtraction of the trend line. Calculation of both the year seasonality and trend must be carried out with the training dataset only, leaving the test dataset completely unseen to the system. Finally, the residual is the remaining random variation in the SST timeseries, which has not been taken into account in the trend and seasonality [294]. The sum of the trend and residual outputs are then fed to the core processing block, which has an stacked architecture [295] of an LSTM and a Multilayer Perceptron (MLP) network. This block is illustrated in Fig. 3.2(b). It consists of two cascaded LSTM layers and one fully-connected MLP layer.

The two cascaded LSTM layers are referred to in parlance as *encoder-decoder* or *Seq2Seq* [234]. The first LSTM layer translates the dynamicity of the input sequence into a higher dimensional representation, whilst the second LSTM layer extracts useful features to feed to the next MLP layer. The MLP layer then combines all the automatically discovered features in the data, into its predictive output. Additionally, two dropout layers are placed between each consecutive DNN pairs to prevent the model from overfitting.

The second DNN branch in Fig. 3.2(a) (i.e., $F_2(\text{Air})$) takes the air temperature as input and performs the same preprocessing as that on the SST. The next block smooths the given high-dynamic data with a moving average window, which acts like an LPF to increase the correlation between the air temperature and SST. Similar to the first branch, the smoothed data is then fed to the core stacked processing block to produce the final outcome of the second branch.

Later in Section 3.4, we will show the prediction efficiency of $F_1(\text{SST})$ as a standalone branch. However, to further improve the prediction accuracy of our model, we implement (3.1) by placing an additional block to combine the outputs of $F_1(\text{SST})$ and $F_2(\text{Air})$, at the end of the network architecture in Fig. 3.2(a). This block takes $F_1(\text{SST})$ and $F_2(\text{Air})$ as the inputs to a McCulloch-Pitts neuron with zero biasing and with a linear activation function. This single neuron is referred to in parlance as the *voting ensemble* [296]. As shown in (3.1), the main objective of this block is to merge the independently predicted results to form the final output of the network, i.e., $F^+(\text{SST}, \text{Air})$. Nevertheless, the weights of this single neuron would be optimized by backpropagation through gradient descent algorithm to produce the ultimate outcome.

The proposed voting ensemble model in Fig. 3.2 includes some hyper-parameters that

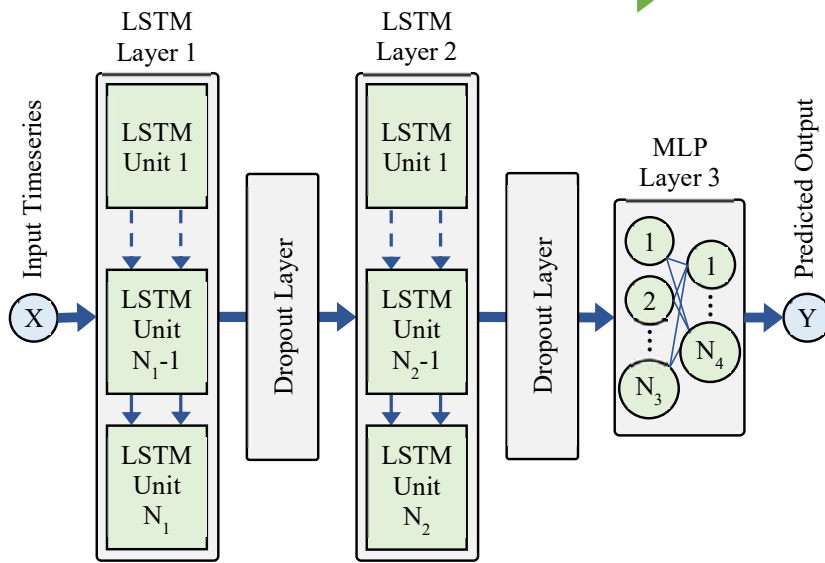
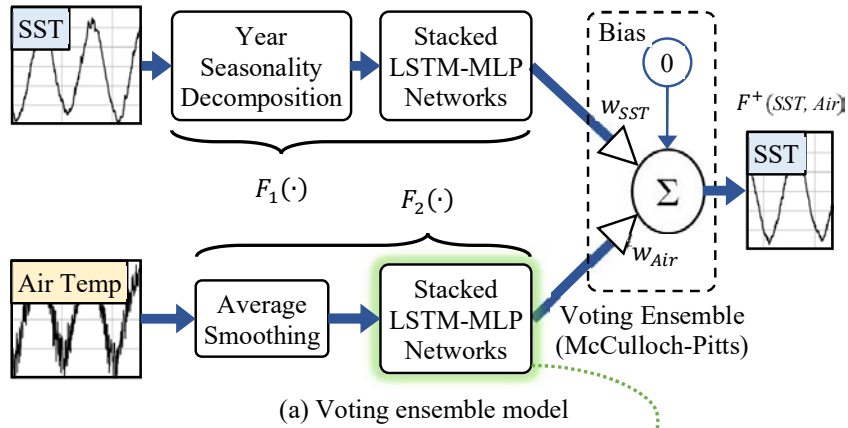


Figure 3.2: Block diagram of (a) the voting ensemble model for SST forecasting, which consists of two (b) stacked LSTM-MLP deep neural networks.

require optimization. The first one is M_1 , which represents the number of data elements inside the input timeseries to the first LSTM layer in Fig. 3.2(b). This parameter can be defined as,

$$M_1 = \frac{\text{Input Horizon}}{\text{Sampling Period}}, \quad (3.2)$$

where the *Input Horizon* refers to the length of time, in which we look back in our historical data to provide future predictions. Additionally, N_1 and N_2 are the number of LSTM units in the first and second layers of the stacked LSTM-MLP network shown in Fig. 3.2(b).

Table 3.2: Mean Squared Error of Area-Averaged SST Forecasting at the Bohai Sea, Compared with the Different Schemes Used in [16]

Model Name	Daily Mean			Weekly Mean		Monthly Mean
	1 Day	3 Days	7 Days	1 Week	3 Weeks	1 Month
GED [16]	0.166	0.415	0.742	0.350	0.514	0.581
FC-LSTM [16]	0.170	0.424	0.787	0.382	0.592	0.687
SVR [16]	0.472	0.692	1.005	0.578	0.627	0.711

Our $F_1(\text{SST})$	0.166	0.322	0.514	0.310	0.696	0.272
Our $F_2(\text{Air})$	2.289	2.829	1.633	0.835	1.135	0.242
Our $F^+(\text{SST}, \text{Air})$	0.157	0.318	0.508	0.294	0.696	0.194

These two parameters should be *adequately* large. Here '*adequately*' means, firstly,

$$N_2 > N_1 > M_1. \quad (3.3)$$

Secondly, N_1 and N_2 should be large enough so that the dropped-out neurons in Fig. 3.2(b) do not adversely affect the network performance.

The output of the second LSTM layer is fed into the first and second dense layers of the following MLP network with N_3 and N_4 neurons, respectively. It is worth noting that very large or small values of N_i will respectively result in overfitting and underfitting, which consequently reduces the overall accuracy. To achieve the best performance, we performed a hyperparameter optimization process that resulted in $N_i \forall i \in \{1, 2, 3\}$ values with 200, 300, and 100 neurons. Finally, the number of output neurons from the MLP layer (N_4) should be equivalent to the number of elements in the predicted output timeseries. This parameter can be defined as the ratio of the *forecasting horizon* to the sampling period, where the *forecasting horizon* simply indicates the length of time of the prospective forecasting period.

3.4 Results and Discussions

In this section, we introduce and discuss the utilized air and water temperature data sources. We also evaluate the efficacy of our proposed model compared to literature.

Table 3.3: Mean Squared Error of Area-Averaged SST Forecasting at the South China Sea, Compared with the Different Schemes Used in [16]

Model Name	Daily Mean			Weekly Mean		Monthly Mean
	1 Day	3 Days	7 Days	1 Week	3 Weeks	1 Month
GED [16]	0.063	0.125	0.211	0.162	0.267	0.207
FC-LSTM [16]	0.061	0.140	0.218	0.168	0.285	0.343
SVR [16]	0.095	0.157	0.242	0.214	0.285	0.212

Our $F_1(\text{SST})$	0.055	0.084	0.131	0.078	0.135	0.104
Our $F_2(\text{Air})$	0.692	0.738	0.802	0.676	0.812	0.874
Our $F^+(\text{SST}, \text{Air})$	0.055	0.084	0.131	0.077	0.135	0.099

Table 3.4: Mean Squared Error of Area-Averaged SST Forecasting at the North Pacific Ocean, Compared with the Proposed Model in [17]

Model Name	Monthly Mean					
	1 Month	2 Months	3 Months	4 Months	5 Months	6 Months
Convolutional LSTM [17]	0.038	0.042	0.040	0.035	0.102	0.072

Our $F_1(\text{SST})$	0.015	0.013	0.017	0.019	0.016	0.018
Our $F_2(\text{Air})$	0.283	0.456	0.294	0.288	0.440	0.488
Our $F^+(\text{SST}, \text{Air})$	0.014	0.013	0.017	0.017	0.016	0.012

3.4.1 Description of Data Sources

The conducted experiments in this chapter are carried out using two major data sources. The first data source is NOAA, which is a USA national scientific agency. NOAA focuses and monitors the conditions of both the oceans and atmosphere, facilitated by its various centers and laboratories. The NOAA PSL contains SST timeseries from September 1981 to present. This data covers the global oceans in daily sampling period, with 0.25° spatial

resolution. In addition, the NOAA NCEI has a global information system tool, which enlists global weather stations. For each station, daily sampled atmospheric measurements such as air temperature, precipitation, wind speed, etc. are publicly available [297].

The second data source used in our experiments is Argo, which is dedicated to oceanographic research and is collected and made publicly available through an international program since the early 2000s. The floating buoys of Argo record temperature, salinity, oceanic currents, bio-optical properties, etc. In contrast to NOAA, Argo does not measure atmospheric parameters, and it has as low as 1° spatial resolution [291].

3.4.2 Prediction Accuracy and Comparison

The prediction results of our voting ensemble model are compared with other published works in Tables 3.2, 3.3, and 3.4. The last three rows in these tables correspond to the three distinct nodes in our proposed model architecture, namely $F_1(\text{SST})$, which is the output of the top branch in Fig. 3.2(a) (SST data only); $F_2(\text{Air})$, which is the output of the bottom branch in Fig. 3.2(a) (air temperature only); and $F^+(\text{SST, Air})$ that represents the output of our voting ensemble model in Fig. 3.2(a) (both SST and air temperature data).

Similar to [16], our models in Tables 3.2 and 3.3 are separately trained with the daily mean, weekly mean, and monthly mean data. Therefore, the Mean Squared Error (MSE) values for 7 days in the daily mean category is different from the MSE values for one week in the weekly mean category.

As can be seen from Tables 3.2 to 3.4, the prediction capability of the standalone first branch in Fig. 3.2(a) (i.e., $F_1(\text{SST})$) is very strong. Among all the comparisons made, this single branch provides better SST forecasting in 16 cases out of the total 18 compared to literature. On average, our $F_1(\text{SST})$ offers 15% and 39% better MSEs, compared to GED [16] in Tables 3.2 and 3.3, respectively. It also provides 65% better MSE, compared to convolutional LSTM [17] in Table 3.4. Note that, GED is the best performing model presented in [16].

After combining the $F_1(\text{SST})$ and $F_2(\text{Air})$ branches using a McCulloch-Pitts neuron to build a voting ensemble, our model outperforms $F_1(\text{SST})$ results and all previously reported works in 17 out of the 18 comparisons. To summarize, on average our proposed $F^+(\text{SST, Air})$ ensemble model provides 19% and 40% better MSEs, compared to GED [16] in Tables 3.2 and 3.3, respectively. It also provides 68% better MSE, compared to convolutional LSTM [17] in Table 3.4.

These results show that our model outperforms previous works, when only using SST data in most of the cases, while it achieves a slight accuracy improvement, if an ensemble

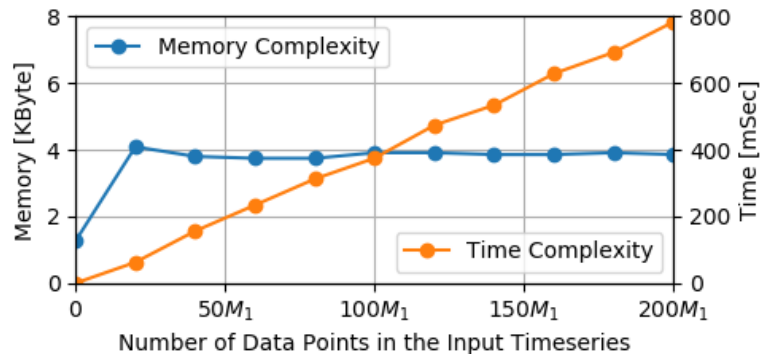


Figure 3.3: Time and memory complexities with respect to the number of data points for one-day $F^+(\text{SST}, \text{Air})$ forecasting in Bohai Sea.

is used to include some readily available air temperature data.

To further analyze the efficiency of our proposed model, we used the big-O approach to evaluate our model’s demand when changing the input size [298]. The analysis of both memory and time complexities for one-day SST prediction at the Bohai Sea are illustrated in Fig. 3.3. These plots quantify the growth in required computational resources, against the number of data points M_1 in the input timeseries in (3.2). The results are obtained during inference, where the weights and biases are fixed. The plots reveal an $O(n)$ linear growth for simulation time, and an $O(1)$ constant growth for the memory demand. These suggest our model is efficient.

When comparing our model to the state-of-the-art for SST prediction, it offers the following advantages.

- To the best of our knowledge, the proposed simple and efficient ensemble of stacked DNNs for SST prediction is unique and has not been reported previously in literature.
- Creatively incorporating the air temperature data from close-by iso-latitude weather stations makes the model more accurate and versatile. As a result, our model outperforms the previous works by 19% to 68% better prediction accuracy by adding readily available air temperature data. This improvement is more prominent in some geolocations than others. For instance, the accuracy improvement for the Bohai Sea increases from 15% to 19% when including air data.
- The innovative injection of air temperature is beneficial to our model not only for accuracy enhancement, but also in reliability improvement. Our model is more resilient to missing SST values and outliers, as the air temperature data is present. By inspecting Tables 3.2, 3.3, and 3.4, it is obvious that we can predict the SST,

using the air temperature only. This prediction has less than $\pm 1^\circ\text{C}$ error in most of the cases. That is, we can use the second $F_2(\text{Air})$ branch as a missing SST value estimator.

One may further improve our model by adding extra branches into its modular design to incorporate more exogenous factors, e.g., solar radiation or wind speed. Besides, the stacked LSTM-MLP DNN in our model can be replaced by novel Transformer networks, which may better learn long-term dependencies compared to an LSTM.

3.5 Conclusion

Accurate long-term SST prediction is challenging. To address this problem, we proposed a light-weight and highly-accurate new DNN structure that leverages the correlation between SST datasets and air temperature at nearby iso-latitude weather stations. We devised two stacked LSTM-MLP networks and trained them with the correlated SST and air temperature datasets. We then integrated the outputs of the two stacked networks in a voting ensemble to form a highly accurate model. We used the two well-known NOAA and Argo data sources to train and test our models. We demonstrated that our model significantly outperforms the state-of-the-art SST prediction algorithms.

Chapter 4

Finite Element DNN for Nitrogen Distribution Prediction in the Great Barrier Reef

The prediction of nitrogen distribution in the Great Barrier Reef (GBR) is proposed in this chapter. The proposed technique considers the whole GBR as a frame and treats forecasting of nitrogen as a next-frame prediction task, to produce spatial maps of nitrogen over the whole GBR at forecast time-steps. To achieve this, an innovative Deep Neural Network (DNN) is designed inspired by the Finite Element (FE) analysis concept. This chapter is published in *Environmental Modelling & Software* as

- [26] M. Jahanbakht, W. Xiang, B. Robson, and M. R. Azghadi, “Nitrogen prediction in the Great Barrier Reef using finite element analysis with deep neural networks,” *Environmental Modelling & Software*, vol. 150, pp. 105311–105327, Apr. 2022.

4.1 Introduction

The Great Barrier Reef (GBR) is the world’s largest coral reef system, located off the east coast of Queensland, Australia. This world heritage site is facing severe threats that challenge its resilience, including extreme weather events and climate change, agricultural pollutants, coastal activities, surface runoff associated with the catchment areas, etc. Among these threats, land and agricultural activities are the main sources of pollutants from GBR catchments [299].

Nutrients, fine sediments, and pesticides are considered to be the primary land-based pollutants that significantly reduce ocean water quality [300]. According to the Australian and Queensland Government’s long-term sustainability plan for the GBR (Reef 2050 Plan) [301], excess nitrogen is particularly challenging in the GBR. High rainfall, flash floods, numerous short river basins, and the close proximity of the reef to the Wet Tropics of Queensland mean nutrients are flushed to the reef lagoon quickly.

Accordingly, the total nitrogen is amongst the most commonly measured and monitored water quality variables worldwide. In coastal and marine waters, nitrogen is usually considered the primary limiting nutrient. In other words, there is a strong consensus that it is the limited supply of nitrogen that limits marine ecosystem productivity in most cases, although phosphorus, silica, and iron may co-limit productivity in some situations [302]. When the total nitrogen increases, the growth and productivity of marine algae and other photosynthesising organisms increases, often to the detriment of marine ecosystems. This process is known as *eutrophication* and there is an extensive literature assessing its prevalence, causes, and management [303].

There is extensive evidence that the coastal waters of the GBR have been subject to some degree of eutrophication due to changes in its catchment land use since European settlement [304–306] and that this has had a negative effect on GBR ecosystems [307,308], though the offshore GBR and much of the midshelf remain oligotrophic (i.e., has low nitrogen and phosphorus concentrations) in absolute terms [309].

Management of nitrogen loads to the GBR in order to improve GBR water quality has been the focus of major investments by state and federal governments, not-for-profit organizations and farmers for many years [310–312]. Towards this end, a greater focus on experimentation, evaluation, and modelling to understand future nitrogen scenarios could further support water quality programs [313]. In particular, predictive models can be used to forecast and manage the high risk areas in the coral reef ecosystems [300].

However, implementing an accurate nitrogen predictor for the vast areas of the GBR is a challenging task. Nitrogen values in the GBR form a big frame (matrix) that vary with both spatial coordinate (x, y) and the time. One technique to handle this giant time-varying frame is to transform it into a timeseries by averaging all nitrogen values on each day.

This technique has been employed by many predictive models for a variety of target parameters, e.g., physical, chemical, and biochemical characteristics of water [314], water quality index [315], nitrogen uptake in crops [316], marine environment salinity, O_2 , NO_3 , phosphorus, silicon, chlorophyll, and alkalinity [317], etc. The employed timeseries forecasting models in these published works range from decision tree and multivariate regression in statistical models to support vector regression in shallow neural networks, and further to the Long Short-Term Memory (LSTM) in deep neural networks. For example, one of the most recent models that has used this averaging technique is the fuzzy partitioning LSTM model introduced by [317]. In this model, the data attributes are partitioned by fuzzy c-means before feeding to an LSTM network for supervised learning. This architecture makes the model ready for high-speed distributed learning, as well as inference.

As opposed to the above technique, there is a second approach to design a next-frame predictor. In this approach, nitrogen values of each day across the GBR form a frame. The goal is to forecast future frames from the historical frames. This approach is referred to as *next-frame prediction* in parlance [318]. While time series forecasting could be applied to predict a value for each pixel separately, next frame forecasting has the great advantage of incorporating both spatial and time-series information rather than considering the history of each pixel in isolation. This provides a much richer source of information for each prediction.

It is worth mentioning that next-frame prediction is a type of forecasting problem, which is different from simulation problems widely carried out by hydrodynamic models [319]. Standalone hydrodynamic models cannot forecast unless future boundary conditions can be reliably predicted, except by coupling with a data-driven surrogate model. For this reason, hydrodynamic models for water quality forecasting are rarely reported in the literature, are mainly timeseries forecasting models, and typically have high errors [320].

To the best of our knowledge, all existing data-driven next-frame predictors in the literature treat each frame as a whole. In other words, they simply stack up historical 2D frames, making a 3D matrix, and then feed the resulting 3D matrix to their Deep Neural Network (DNN) models to output a 2D prediction frame. Some of the commonly used DNNs are recurrent neural networks [321], 3D Convolutional Neural Network (Conv3D) [322], Convolutional Long Short-Term Memory (ConvLSTM) [323,324], etc.

One of the most successful next-frame predicting models in the literature is PhyDNet proposed by [324]. PhyDNet disentangles physical knowledge described by partial differential equations from data, before feeding it to the ConvLSTM model. The experiments with sea surface temperature data showed the ability of PhyDNet to outperform state-of-the-art methods. In ensuing sections, we will apply PhyDNet to our nitrogen distribution dataset for comparison. We will show that the main disadvantage of these next-frame predictors is their low coefficient of determination (R^2). In other words, frames predicted by these models are blurred (i.e., reduced R^2) to reduce their overall prediction error (as measured by the mean squared error (MSE)). To address this problem, we propose a new DNN inspired by the Finite Element analysis (FE-DNN). By dividing the GBR study area into small elements, and by introducing the so-called stiffness matrices concept from the finite element analysis into the proposed FE-DNN model, prediction accuracy is increased, while the details of data variations are preserved.

To investigate the performance of the proposed FE-DNN model, we employ it to forecast nitrogen distribution frames in the GBR from hindcast distributions provided by an existing Partial Differential Equations (PDE) based simulation model. This distribution follows a complicated set of PDEs [325]. The *eReefs* modelling suite [299] provides plenty

of simulated nitrogen distribution data based on biogeochemical transformations and the spatial distribution of total nitrogen across the GBR but does not forecast future values. In addition, there are some sparsely collected nitrogen measurements across the GBR, which are useful in understanding and predicting nitrogen distribution in the GBR. These criteria make nitrogen prediction a good case study for FE-DNN implementation.

The rest of this chapter is organized as follows. In Section 4.2, nitrogen in the GBR will be defined, and challenges in high-resolution nitrogen prediction will be discussed. Section 4.3 will describe FE-DNN as our proposed solution to the problem of next-frame nitrogen prediction in the GBR. This data is introduced in Subsection 4.4.1. We will then evaluate the accuracy of the FE-DNN model for nitrogen distribution forecasting in the rest of Section 4.4, where a detailed investigation of both the computational complexity and the ablation properties of our model is also provided. The chapter is concluded in Section 4.5.

4.2 Background and Problem Definition

The GBR is recognized by [UNESCO](#) as a World Heritage Area of “Outstanding Universal Value” due to its great cultural and natural significance and unmatched biodiversity. As stated in the previous section, reduced water quality since European settlement has been identified as a key threat to the health and resilience of GBR ecosystems [307]. While climate change is the single greatest threat to the world heritage status of the GBR, water quality adds cumulative pressure, reduces the resilience of reef ecosystems to climate change [308], and may be more readily subject to improvement through local management actions. To support such actions, it is important to be able to monitor and predict water quality on the scale of the whole GBR.

In this regard, land-sourced pollutants from farms and runoff in GBR catchments cause major damage to coral reefs. Among all the primary pollutants, the greatest water quality risks to the GBR are from nitrogen discharges [301]. Accordingly, the key component of the Australia’s Reef 2050 Plan is to manage all nitrogen forms in GBR waters, including nitrogen oxides (NO_x), ammonia ions (NH₄⁺), detrital particulates, dissolved organic nitrogen, living biological forms of nitrogen, etc. [325].

There is continual recycling between all the above forms of nitrogen. Therefore, the Total Nitrogen (TN) is defined as the sum of all the nitrogen forms and employed in this chapter as the parameter of interest.

TN distribution in the GBR can be simulated through a complicated set of PDEs which are solved by the *eReefs* modelling suite. The *eReefs* marine models are process-based simulation models that predict (in hindcast) the past spatial and temporal distribution

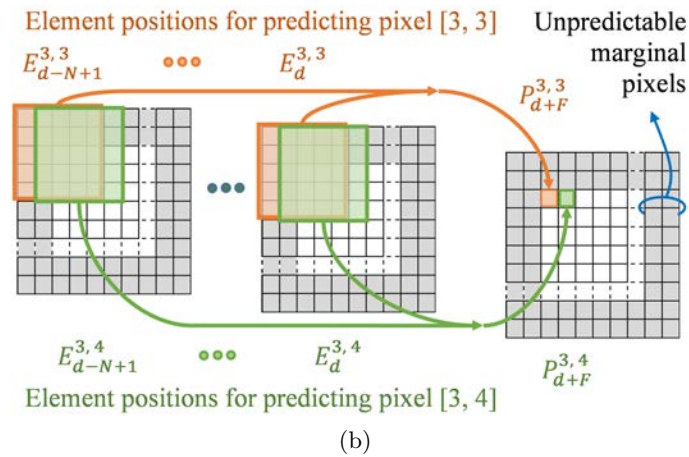
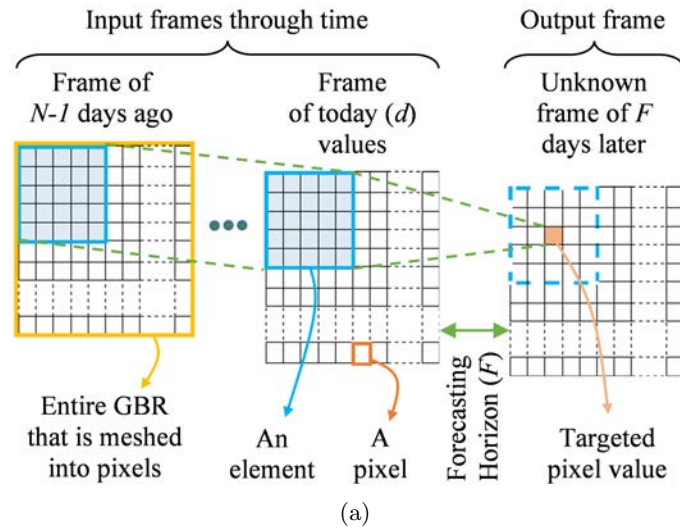


Figure 4.1: (a) The study area frame is meshed into small pixels that hold the temporal TN values of each of the past days until today. The pixels of each of the N historical frames are grouped into 5×5 elements, which are then used to predict the TN value of a target pixel F days after today. (b) The elements are then swept across the study area to predict all the target pixels one-by-one, while leaving a narrow 2-pixel margin of unpredictable TN values.

of physical and water quality variables in space and time as a function of environmental drivers including river discharges, meteorological conditions, and global ocean currents.

This suite of models includes components that simulate hydrodynamics, sediment dynamics, biogeochemical transformations of water quality variables, and optical conditions in the water by numerical integration of a set of PDEs [299]. The models are used to supplement sparse *in situ* water quality observations to support monitoring of the Great Barrier Reef, and have also been used to project how water quality might change under

alternative land management scenarios, to support policy decisions for GBR catchments. Hence, they provide two types of predictions:

- *Hindcasting* (i.e., prediction of past conditions); and
- *Projection* (i.e., counter-factual scenario analysis).

However, *eReefs* does not currently provide forecasting predictions (i.e., prediction of conditions at a specific point in future time). Prediction of water quality in lakes, rivers and marine ecosystems has long been a focus of research and pragmatic modelling efforts. This began with simple empirical and physics-based models of phosphorus and nitrogen dynamics in aquatic systems and has steadily progressed over time to more and more complex coupled hydrodynamic-biogeochemical-ecosystem modelling systems. The range and development of these models has been reviewed by [326], [327], and [328]. Hydrodynamic water quality models in current use are generally complex, computationally intensive, and have limited application in forecasting contexts.

More recently, machine learning approaches have been adopted in water quality modelling. For example, [329] applied the Adaptive Neuro-Fuzzy Inference System, Radial Basis Function Neural Network, and Multi-Layer Perceptron Neural Network to forecast time-series of three water quality parameters in a river basin as a function of a range of other water quality observations, while [330] compared the performance of an artificial neural network, a support vector machine, and a group method of data handling in a similar context. Most machine learning applications to date have been limited to forecasting time-series of water quality at one or a few discrete locations. Other works have focused on hybrid approaches such as using machine learning models as surrogate or meta-models for physics-based water quality models, or the use of machine learning to support data assimilation to improve the performance of physics-based models. One example of this approach is the work of [331], who used error-subspace emulators to assimilate remote sensing ocean color data into the *eReefs* marine sediment dynamics models.

In contrast to all previous works, we employ the DNN to forecast TN distribution in the wide GBR. Based on the *eReefs* findings, the TN in nearshore regions of the GBR is due mostly to river discharges. On the other hand, in the midshelf we can also see marine sources for TN. These various nitrogen sources in the wide coverage of the GBR, make the TN distribution a challenge task to predict. In the current study, we show how hindcast results from a process-based environmental model can be used to train an FE-DNN model to provide forecast predictions. Measured TN values in the GBR are scarce, making data-hungry DNN training unfeasible. To address this, a physics-informed neural network will be designed.

4.3 Proposed Model

As discussed in Section 4.1, there is no high-resolution model in the literature that is able to forecast TN distribution over the GBR. In our proposed model, shown in Fig. 4.1a, we solve the TN forecasting problem by meshing the GBR study area into small overlapping *elements*. To elaborate, each day in N days of the input frames consists of a frame of TN values of all the meshes. The TN value for each mesh is termed a pixel, which represents the average TN in a 16 km^2 mesh area. Several pixels are then grouped into a matrix to create a square element.

The historical element-wise TN values, until the present day, are used to predict a pixel value for F days later, where F is known as the forecasting horizon. As illustrated in Fig. 4.1b, the element is slid across the entire frame so that all possible pixel values can be predicted. The element size shown in Fig. 4.1 is 5×5 . However, this size is a hyperparameter in our model that needs to be optimized. As shown, there is a narrow margin of pixels where TN values cannot be predicted. The size of this unpredictable margin is equal to half of the element size, e.g., 2 pixels for 5×5 elements and 3 pixels for 7×7 elements.

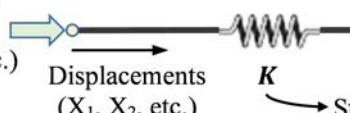
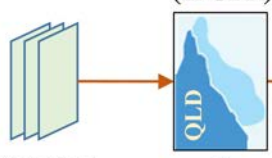
It is worth noting that classic image processing techniques for filling the marginal pixels (e.g., padding, flipping) are not suitable for our TN distribution prediction problem. This is mainly because every pixel in the GBR is highly dependent on its neighbors' historical TN values, which are chaotic, asymmetric, and highly dynamic [332]. Therefore, we cannot simply fill these pixels by techniques such as constant padding or symmetrical flipping. Alternative solutions are needed to predict these marginal pixels, e.g., linear regression, convolutional neural network, LSTM, etc.

4.3.1 FE-DNN Model

To accurately predict the output TN frame from the timed input frames, we design a novel DNN. This network is inspired by the concept of Finite Element Analysis (FEA), and thus is dubbed FE-DNN. FEA is a well-known numerical method for solving boundary value problems in engineering. This method is extensively used in mathematical physics simulations, i.e., current transient response to the current transient inputs. However, FEA cannot be readily applied to forecasting scenarios for obtaining future response to the historical inputs, unless future inputs (boundary conditions) can be accurately estimated. To address this problem, our proposed FE-DNN integrates the concept of FEA with modern machine learning techniques to predict unknown future values.

Table 4.1 provides a conceptual comparison between FE-DNN and FEA, by using the well-known linear spring problem. FEA starts by dividing a study area into small elements

Table 4.1: Conceptual comparison between FE-DNN and FEA

Finite Element Analysis			
Abstract of the governing equation: $\mathbf{X} = f(\mathbf{F}, \mathbf{K})$		Processing technique: Recursive approach in linear algebra	
Excitations	Target values	An element (linear spring)	Boundary condition
External Forces (F_1, F_2 , etc.)	Displacements (X_1, X_2 , etc.)		\mathbf{K} Stiffness matrix
FE-DNN			
Abstract of the governing equation: $\mathbf{P} = \mathbf{W} f(\mathbf{E}, \mathbf{K})$		Processing technique: Gradient descent in deep learning	
Excitations	An element (in GBR)	Target values	Boundary condition
Historical values (\mathbf{E})		\mathbf{P}	Forthcoming frames as an open boundary condition

[333]. In each element, the governing equation in FEA (i.e., $\mathbf{X} = f(\mathbf{F}, \mathbf{K})$) takes both the stiffness matrix \mathbf{K} and the excitations as input, and yields target values. Similarly, our DNN can be expressed as a system of linear equations (i.e., $\mathbf{P} = \mathbf{W} f(\mathbf{E}, \mathbf{K})$), where \mathbf{P} represents the unknown output pixel values in the predicted frame, \mathbf{W} is the known weights of the neurons, \mathbf{E} is the known input frames, and \mathbf{K} is the known stiffness matrix¹. Additionally, the recursive approach to solve the FEA is similar to the recursive gradient descent approach to solve the system of linear equations in FE-DNN.

The stiffness matrix \mathbf{K} in FEA for the spring problem represents the elastic behavior of the underlying material. We will discuss \mathbf{K} in more detail in Section 4.3.2. However, in contrast to FEA that multiplies the inversed stiffness matrix into the excitation, we feed \mathbf{K} as a separate input to our DNN (i.e., $f(\mathbf{E}, \mathbf{K})$). This way, the FE-DNN will learn the behavior of the stiffness matrix, in conjunction with input TN values variations across the historical frames.

¹During the training phase in our supervised learning, \mathbf{P} is known and \mathbf{W} is unknown.

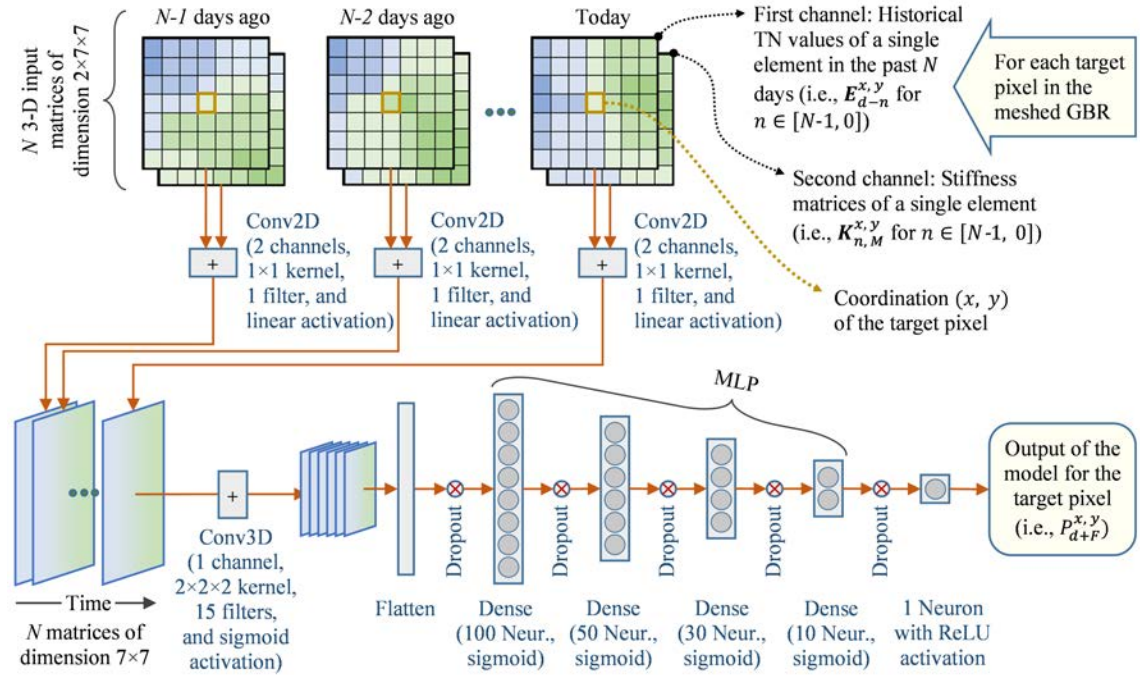


Figure 4.2: The architecture of the proposed FE-DNN network to predict a TN frame in F days later, where d represents today. Here, N input elements (as shown in Fig. 4.1) surrounding a target pixel with coordinates of (x, y) as well as N stiffness matrices (calculated for month M by (4.3)) are required to predict the TN value at the target pixel $P_{d+F}^{x,y}$. The output of the model for the target pixel is fed into Fig. 4.5. The shown kernel sizes, filter numbers, activation functions, etc. are optimized for this study, and they can be different in other next-frame prediction applications.

Fig. 4.2 illustrates the architecture of the proposed FE-DNN model. It takes N elements along with N stiffness matrices as input. These inputs are fed into their N corresponding 2D Convolutional layers (Conv2D). The Conv2D represents $f(\mathbf{E}, \mathbf{K})$ in Table 4.1, which merges the stiffness matrices with their relevant TN elements. The resulting merged matrices then form a 3D matrix and fed to a Conv3D layer. Finally, the outputs of the Conv3D layer are flattened to enter a Multilayer Perceptron (MLP) with four dense layers.

Except for the last dense layer that uses the Rectified Linear Unit (ReLU) activation function, the rest of the MLP dense layers use sigmoid. Our experiments show that using ReLU in all dense layers enforces the lower TN bound to be 0. This makes the model lazy in truly learning the complex transformations in GBR nitrogen distribution. As a result, the model cannot capture TN variations, which results in a lower R^2 . In other words, a model with ReLU activation functions cannot capture TN variations in the vast GBR as good as it could do with sigmoid.

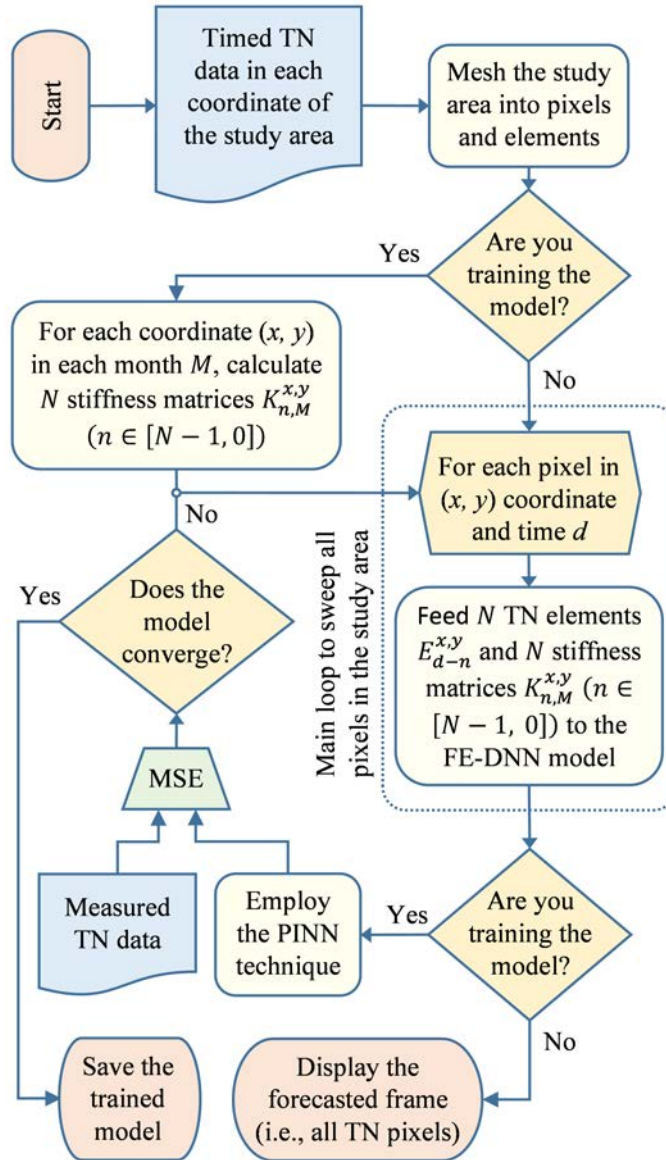


Figure 4.3: Flowchart of the FE-DNN workflow from input data to forecasted results.

The total number of trainable weight parameters in the proposed FE-DNN model is 154,136. To avoid overfitting, while training these weights, five dropout points with a 20% dropping ratio are placed in layer intervals of Fig. 4.2. All the layers are equipped with the Ridge regularization of $L2 = 0.01$. Meanwhile, the learning rate is set to 0.001 in an effort to both improve convergence of weight learning and avoid overfitting.

To better understand how the proposed FE-DNN model works, a flowchart is presented in Fig. 4.3. This flowchart covers both the training and inference phases. The workflow

starts by meshing the study area into pixels, and completes by saving the trained model or yielding the forecasted results. Data flow in this diagram has a main loop to sweep the input elements' locations to predict the TN values in every pixel of the output frame. This loop is marked by a dashed-line inside the figure.

4.3.2 Stiffness Matrices

In FEA, stiffness matrix calculation is a pre-processing step of numerical modelling. The stiffness matrix can be defined as an approximate solution to the underlying PDEs, which represent the elastic deformation of matters in accordance with both their own properties and the constant external perturbations [333].

Accordingly, calculating the stiffness matrix requires obtaining a solution to the complex underlying PDEs of the system under consideration. However, in our proposed FE-DNN method, instead of finding an approximate solution to these complex PDEs to achieve the required stiffness matrices, we use existing training data to extract the variation of the output pixel in response to the changes in the input frames. While the resulting matrix resembles the definition of the stiffness matrix in FEA, its calculation requires no knowledge of the underlying PDEs.

To elaborate, consider the linear spring problem in Table 4.1, with two external forces F_1 , F_2 , and one displacement value X_1 . The spring constant is $k = 2.0$, which results in the stiffness matrix $\mathbf{K} = [2.0, -2.0]$. We simulate this problem for F_1 , F_2 , and X_1 in Fig. 4.4a, where F_1 and F_2 are sine functions in the presence of random Gaussian noise, and X_1 is the target displacement. The simulation is conducted for 2 seconds with 20 sample points, which form our training dataset. By dividing this time into 4 segments with 5 samples per segment, one can numerically calculate the stiffness matrix of each segment r , \mathbf{K}_r , as

$$\mathbf{K}_r \mathbf{X} = \mathbf{F} \Rightarrow \mathbf{K}_r = \frac{\mathbf{F}}{\mathbf{X}} \xrightarrow[\text{matrix algebra}]{\text{In the form of}} \mathbf{K}_r = \mathbf{F} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}, \quad (4.1)$$

where \mathbf{F} is a 2×5 matrix of five F_1 and F_2 samples, and \mathbf{X} is a 1×5 matrix of five X_1 samples. The final stiffness matrix \mathbf{K} of the linear spring problem can then be calculated by averaging \mathbf{K}_r as follows

$$\mathbf{K} = \frac{1}{4} \sum_{r=1}^4 \mathbf{K}_r. \quad (4.2)$$

The true versus calculated values of \mathbf{K} are presented in Fig. 4.4a. As can be seen from the figure, the result has about 97% accuracy across the entire dynamic range.

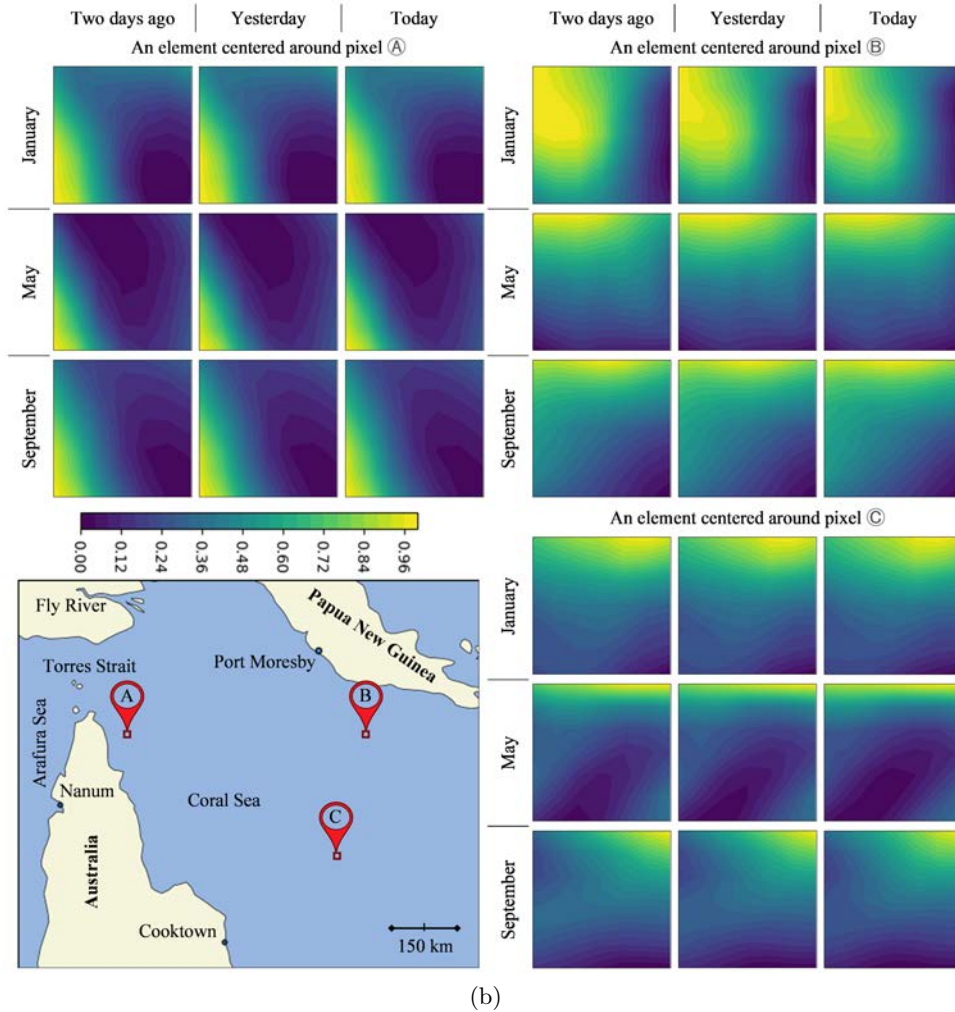
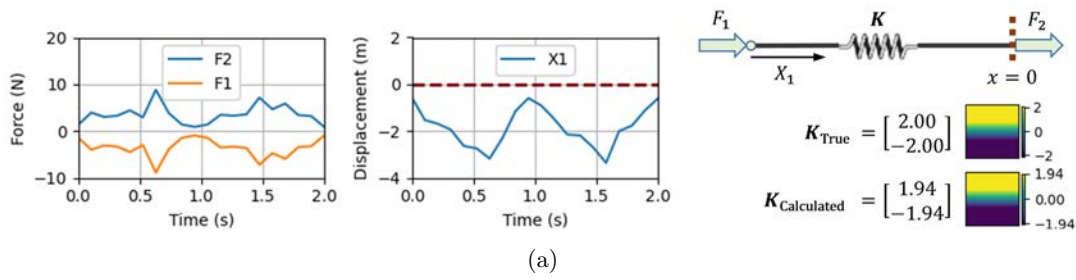


Figure 4.4: (a) Stiffness matrix calculation for the linear spring problem in Table 4.1 in the presence of white Gaussian noise. (b) The stiffness matrices in normalized logarithmic scale of three typical geolocations \textcircled{A} at $(11.18^\circ\text{S}, 143.25^\circ\text{E})$, \textcircled{B} at $(11.18^\circ\text{S}, 148.17^\circ\text{E})$, and \textcircled{C} at $(13.70^\circ\text{S}, 147.57^\circ\text{E})$ in three months of January, May, and September, where the element size is 7×7 .

Considering the TN stiffness matrix in the GBR, applying (4.1) and (4.2) to our TN prediction problem requires the following adjustments:

1. Each pixel in the targeted output frame is influenced by the past N days, so we will have N stiffness matrices for each pixel.
2. We can split the wide GBR TN values in time, by calculating the stiffness matrices for each month of the year. In this way, the high-dynamics of the GBR will be better captured.
3. As illustrated in Fig. 4.2, in our proposed method \mathbf{K} is calculated and provided to the model as a separate input. This is unlike FEA, where the stiffness matrix \mathbf{K} is mathematically multiplied by its relevant element.
4. By assigning a 1×1 kernel and 1 filter to all the Conv2D layers in Fig. 4.2, these layers implement $\mathbf{E} + \mathbf{K}$. However, given the negative values in \mathbf{K} , the output of the Conv2D layers are $\mathbf{E} - \mathbf{K}$. Therefore, in our proposed model, unlike the original FEA implementation where the stiffness matrix is calculated as $\mathbf{K} = \mathbf{F}/\mathbf{X}$, it is calculated by subtracting the known outputs of the model (\mathbf{P}) from element \mathbf{E} , i.e. $\mathbf{K} = \mathbf{E} - \mathbf{P}$.

Given the above adjustments, we reformulate (4.1) and (4.2) to better address the requirements of our GBR application. As also shown in Fig. 4.1, consider the situation, where N input elements \mathbf{E} around the target pixel at coordinate (x, y) are used to calculate its value P in F days after today $P_{d+F}^{x,y}$, in month M . The stiffness matrix \mathbf{K} will be

$$\mathbf{K}_{n,M}^{x,y} = \frac{1}{N_y} \frac{1}{N_d} \sum_y \sum_d (\mathbf{E}_{d-n}^{x,y} - P_{d+F}^{x,y} \mathbf{J}) \quad \forall n \in [0, N - 1], \quad (4.3)$$

where y sweeps the years of the training dataset, N_y is the number of training years, d sweeps the days of month M , N_d is the total number of days in month M , and \mathbf{J} is an all-one matrix of the same size as our elements. As can be seen from (4.3), we will have N stiffness matrices for a given coordinate (x, y) in a given month M of the year, with a given forecasting horizon F . In other words, $\mathbf{K}_{N-1,M}^{x,y}, \mathbf{K}_{N-2,M}^{x,y}, \dots, \mathbf{K}_{0,M}^{x,y}$ represent the average variations of a pixel in month M , in response to the element-wise TN variations in the last N days. It is worth mentioning that the stiffness matrix \mathbf{K} in (4.3) depends on the month but is independent of both the day and year.

To better illustrate these calculations, the normalized stiffness matrices for three random geolocations (x, y) in the GBR are plotted in Fig. 4.4b. These plots are made for the

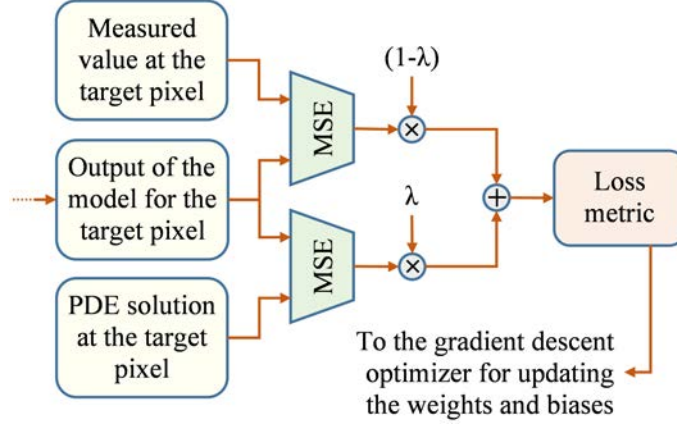


Figure 4.5: PINN loss function applied to our proposed FE-DNN model.

months of January, May, and September. The element size is set to be 7×7 , and the forecasting horizon (F) is one day. The number of input frames (N) is equal to 3, resulting in three stiffness matrices per month. These stiffness matrices are labeled as *two days ago*, *yesterday*, and *today*, denoted by $\mathbf{K}_{2,M}^{x,y}$, $\mathbf{K}_{1,M}^{x,y}$, and $\mathbf{K}_{0,M}^{x,y}$, respectively.

4.3.3 Physics-informed Neural Network

To train the proposed FE-DNN model, one would require a large quantity of observational TN data in the GBR. However, the existing sparse TN measurements in GBR are insufficient for our data-hungry DNN. To overcome this problem, a novel method termed Physics-Informed Neural Network (PINN) [334] is employed. The use of PINN enables us to merge scarce observational data with readily available *eReefs* simulation results, and use both types of data to train our neural network.

Despite its name, the PINN is not a new neural network on its own right, but a technique in defining a physics-informed loss function, which mixes PDE solutions with measured values (i.e., the ground-truth). Therefore, it can be applied to almost any neural network corresponding to a physical model that can be described by underlying PDEs.

The PINN-inspired loss function that we develop for our FE-DNN model is illustrated in Fig. 4.5, using the Mean Squared Error (MSE) metric. The *output of the model* in this figure is the output of our FE-DNN network in Fig. 4.2. We use this output to calculate two loss functions as follows

$$L_{\text{Measured}} = \frac{1}{N_{\text{data}}} \sum (\text{TN}_{\text{Output}} - \text{TN}_{\text{Measured}})^2, \quad (4.4)$$

$$L_{\text{PDE}} = \frac{1}{N_{\text{data}}} \sum (\text{TN}_{\text{Output}} - \text{TN}_{\text{PDE}})^2, \quad (4.5)$$

where N_{data} is the total number of data points, $\text{TN}_{\text{Measured}}$ is the observational TN values, and TN_{PDE} is the simulated TN values obtained from *eReefs*.

The loss functions in (4.4) and (4.5) are then combined together to create the following overall loss metric

$$L = (1 - \lambda) L_{\text{Measured}} + \lambda L_{\text{PDE}}, \quad (4.6)$$

where λ is an adjustable hyperparameter. We then use the loss function in (4.6) to train our model and to recursively optimize the unknown weights of the FE-DNN network. To summarize, we overcome the observational data sparsity problem by integrating the PINN technique with the process of DNN training.

4.4 Results and Discussions

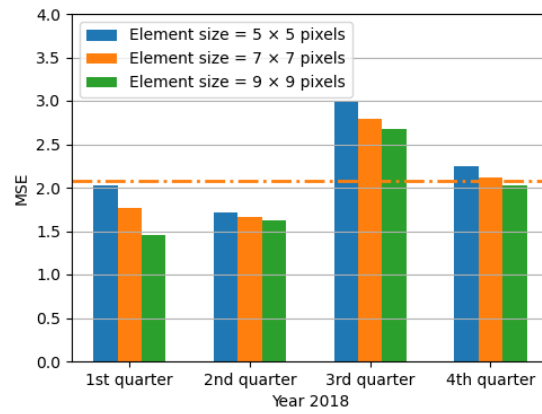
In this section, we will start by introducing the measured TN data, along with the PDE simulation results for TN in the GBR. We will then optimize the element size, before proceeding to the accuracy analysis, computational complexity, and ablation studies.

4.4.1 Data Sources

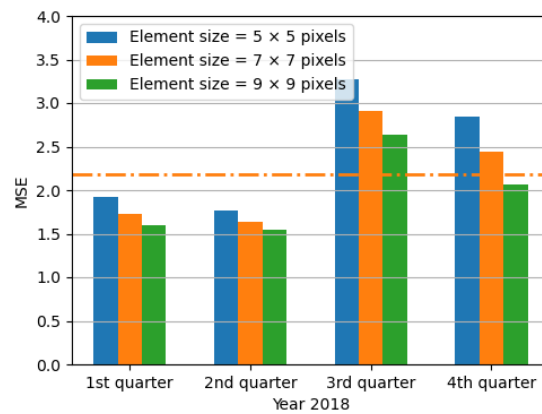
The proposed FE-DNN network is used to predict the TN distribution in the GBR. The observational TN values (i.e., $\text{TN}_{\text{Measured}}$ in (4.4)) are gathered from the GBR Marine Park Authority Marine Monitoring Program (MMP), which is led by Australian Institute of Marine Science (AIMS) [335]. These measurements are sparsely gathered and thus are insufficient for training the proposed FE-DNN model. Hence, by integrating the PINN technique described in Section 4.3.3, the simulated data (from solving PDEs) are used to compensate for the scarcity of the measured data.

As discussed earlier in Section 4.2, to obtain the PDE solutions for the TN distribution in the GBR (i.e., TN_{PDE} in (4.5)), the *eReefs* modelling suite is employed. *eReefs* has a regional model on a 4 km grid (GBR4), which extends into the Coral Sea and covers the entire GBR area [336]. However, this raw GBR4 biogeochemical model has another version, which is interpolated onto a regular grid. This version of the *eReefs* simulation data is downloadable from the AIMS website [337], and we have, therefore, used it in our study.

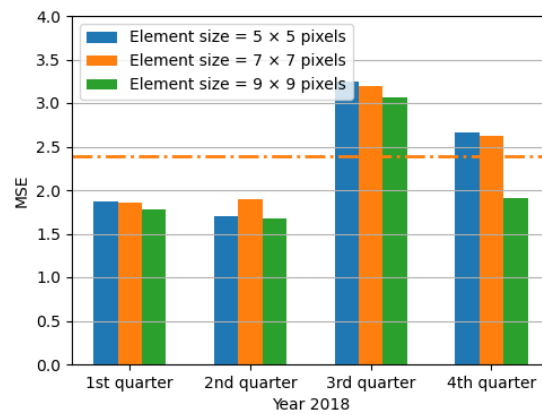
The *eReefs* simulation data from the AIMS website [337] is provided on a daily basis, from 2011 to 2018. We divide this time span into 2011 to 2017 for the training and validation dataset, and 2018 for the testing dataset. To increase the model training speed,



(a)



(b)



(c)

Figure 4.6: Effect of element size on the MSE metric in (a) 3-day, (b) 5-day, and (c) 7-day forecasting horizons, where number of past days used for training is $N = 3$. The horizontal dashed lines show the average MSE for the 7×7 element size.

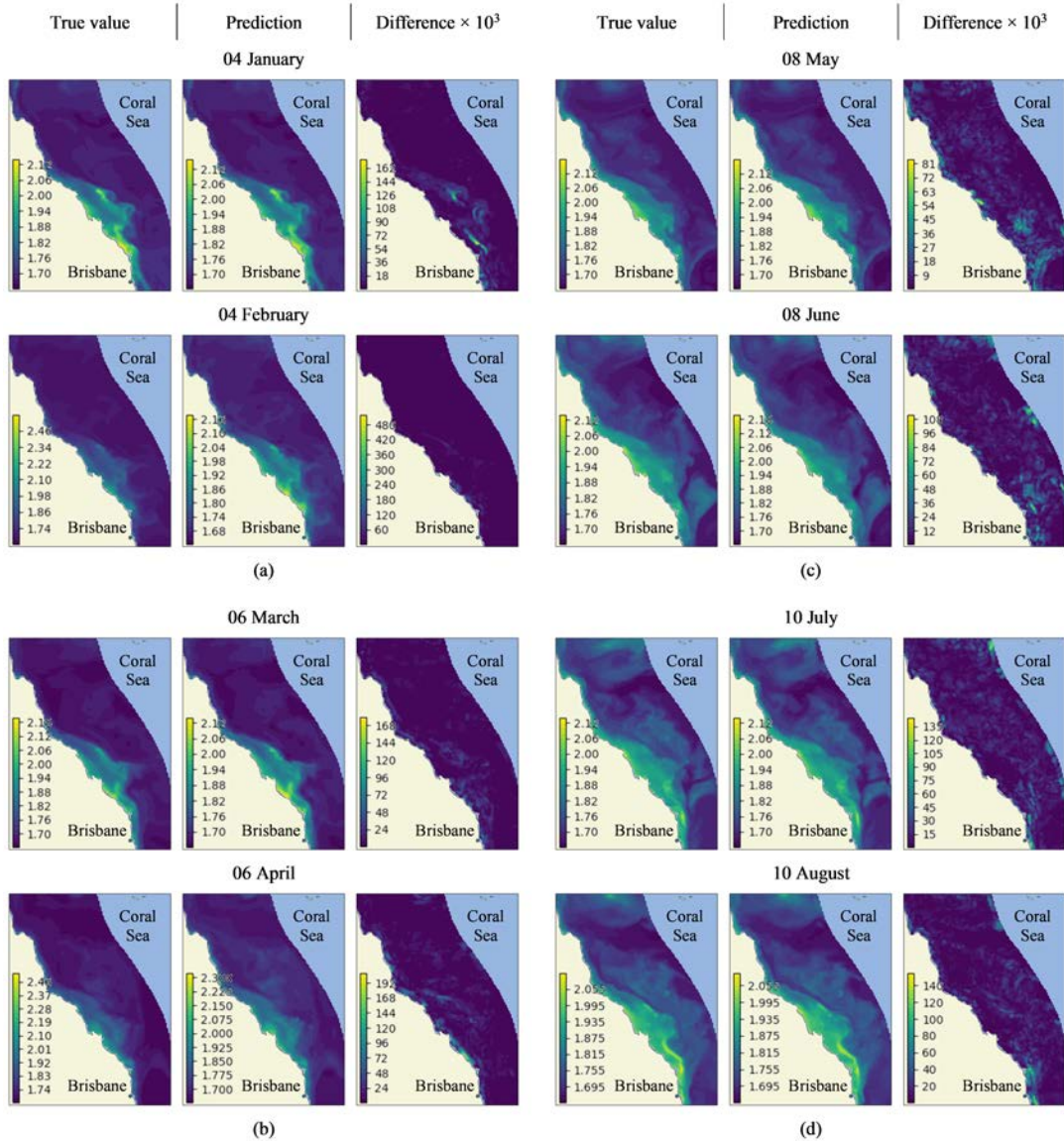


Figure 4.7: True values, predictions, and their absolute differences (multiplied by 10^3) for the 2018 test TN_{\log} dataset in the GBR, for (a) 1-day (b) 3-day, (c) 5-day, and (d) 7-day forecasting horizons.

we have spatially downsampled the dataset by a factor of 4, which has resulted in 16 km wide pixels, as mentioned in Section 4.3.

The TN concentrations within river deltas in the GBR are so high that using the full range of the data values for processing makes variations in other areas appear insignificant, even though there are significant and ecologically important variations in water quality (including sediment and nitrogen concentrations) throughout the nearshore regions and

Table 4.2: Statistics of the values of TN_{PDE} from *eReefs* marine models

Dataset	Year	Min	Max	Mean	STD
Train and Validation	2011	44.05	528.37	59.36	22.88
	2012	44.33	464.63	56.88	16.57
	2013	44.10	427.97	56.26	15.84
	2014	43.18	410.69	54.63	12.71
	2015	43.00	254.96	53.04	10.43
	2016	42.86	244.99	52.25	10.20
Test	2018	43.03	369.38	54.00	11.85

Table 4.3: MSE and R^2 metrics in TN prediction of the test dataset in 2018, using elements of size 7×7 pixels

Metric	Forecasting horizon	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
MSE	1 Day	1.21	0.83	1.75	1.11	0.98	1.62	2.25	3.50	2.04	1.33	1.55	1.55
	3 Days	1.76	1.22	2.30	1.91	1.15	1.92	2.84	3.03	2.51	1.81	2.24	2.31
	5 Days	1.74	1.33	2.11	1.55	1.31	2.07	2.66	3.33	2.73	2.15	2.46	2.74
	7 Days	2.04	2.00	1.54	1.97	1.49	2.23	2.47	4.27	2.84	2.40	2.29	3.18
R^2 (%)	1 Day	99.22	99.45	99.12	99.40	99.41	99.07	98.84	98.32	98.89	99.13	98.91	99.07
	3 Days	98.86	99.20	98.85	99.00	99.31	98.90	98.54	98.53	98.64	98.82	98.45	98.63
	5 Days	98.88	99.11	98.93	99.15	99.21	98.81	98.63	98.39	98.51	98.59	98.27	98.40
	7 Days	98.68	98.61	99.25	98.87	99.09	98.70	98.71	97.94	98.47	98.46	98.34	98.12

out to the Midshelf waters. To capture these variations, we use the logarithmic scale for model training as follows

$$TN_{\log} = \log_{10}(TN + 1). \quad (4.7)$$

The added 1 in (4.7) is to avoid $\log(0)$. This equation is used both for TN scaling and for the stiffness matrix calculation. It is worth mentioning that the logarithmic scaling is not required in other next-frame prediction applications if the data is linearly distributed between its boundaries. To better understand the nature of these *eReefs* simulation outputs, the statistics of TN values are presented in Table 4.2. All the data in this table are in linear scale.

Table 4.4: Comparing the performance of the proposed FE-DNN model with two recently published works in the literature

DNN Model	Forecasting horizon (F)				Forecasting horizon (F)				Forecasting horizon (F)			
	1	3	5	7	1	3	5	7	1	3	5	7
	MSE				MAE				R^2 (%)			
Conv3D by Mathieu et al. [365]	25.58	31.16	38.83	41.02	3.07	3.40	3.79	3.80	88.3	85.2	81.5	80.5
PhyDNet by Guen and Thome [367]	8.59	11.62	18.72	21.30	1.60	2.03	2.48	2.73	95.4	92.6	88.7	86.5
The Proposed FE-DNN model	1.64	2.08	2.18	2.39	0.75	0.84	0.87	0.91	99.1	98.8	98.7	98.6
	Bias				Scatter Index (%)				Reliability (%)			
Conv3D by Mathieu et al. [365]	-0.56	-0.38	-0.22	0.16	6.4	7.9	9.1	9.5	97.0	96.5	95.3	94.7
PhyDNet by Guen and Thome [367]	-0.41	-0.68	-0.26	-0.38	3.8	5.6	6.96	7.6	98.7	98.8	97.6	97.2
The Proposed FE-DNN model	-0.04	-0.01	0.14	0.23	2.3	2.6	2.7	2.8	99.97	99.96	99.95	99.95
	U95				F-test (p-value)				DI			
Conv3D by Mathieu et al. [365]	0.28	0.29	0.29	0.29	0.34	0.36	0.40	0.47				
PhyDNet by Guen and Thome [367]	0.26	0.27	0.27	0.27	0.30	0.33	0.41	0.44				
The Proposed FE-DNN model	0.26	0.26	0.26	0.26	2e-34	3e-25	9e-29	3e-20				

4.4.2 Prediction Accuracy

As in the *eReefs* modelling suite, the physical unit of TN in this thesis is $[\text{mg N/m}^3]$, which is the same as $[\mu \text{ g N/L}]$. By contrast, TN measurements in the MMP are made in $[\mu \text{ mol/L}]$. So, we need to convert the MMP values by multiplying them by 14.01, given that the molar mass of nitrogen is 14.01 g.

As stated earlier in Section 4.3.1, the element size is a hyperparameter that needs to be optimized. The effect of the element size on prediction accuracy is investigated in Fig. 4.6. The MSE values in this figure are averaged per quarter of the training year of 2018. In all cases of the 3-day, 5-day, and 7-day forecasting horizons, increasing the element size generally improves the performance. By contrast, a greater element size leads to more unpredictable marginal pixels, as shown in Fig. 4.1b. The horizontal dash lines in this figure indicate the average MSE values for the 7×7 element size. In the reported results of our FE-DNN, we have used the element size of 7×7 .

Fig. 4.7 demonstrates our FE-DNN TN prediction results for a typical day of the first

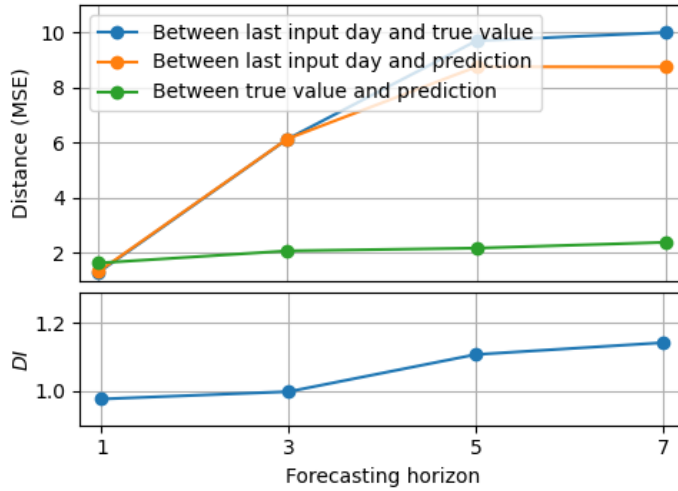


Figure 4.8: Pairwise mathematical distances between the input values of the last day, true values in F days later, and our prediction, using elements of size 7×7 pixels and $N = 3$, along with their corresponding DI.

8 months in 2018 (i.e., for the test dataset). The true values and the absolute differences between the true and predicted values are also plotted. All the predictions are made using 7×7 elements with $N = 3$, and all the absolute differences are multiplied by 1000. The results are in logarithmic scale, and they show 1-day, 3-day, 5-day, and 7-day forecasting horizons. This demonstrates the ability of our model to very closely predict TN values across the entire GBR area.

Due to the adoption of the element size of 7×7 and based on the illustration in Fig. 4.1, three marginal pixels are left unpredicted by the proposed FE-DNN model. These pixels are predicted in Fig. 4.7 by employing a simple linear regression model. The higher error values of the regression model are obvious in the surrounding margins of this figure, especially in the longer forecasting horizons.

For all $F = 1$ in Fig. 4.7a, $F = 3$ in Fig. 4.7b, $F = 5$ in Fig. 4.7c, and $F = 7$ in Fig. 4.7d, the 1st, 2nd, and 3rd days of each month are fed to the model's input. The 4th, 6th, 8th, or 10th days of the months are forecast in those forecasting horizons, respectively. The absolute differences (i.e., the prediction errors) spread geographically, when increasing the forecasting horizon F , resulting in a larger MSE.

Fig. 4.7 shows that not only the proposed FE-DNN can result in very accurate prediction, it also generates unblurred output frames (i.e., a high coefficient of determination R^2), which are not achievable by conventional next-frame prediction methods. Both of these advantages are demonstrated in more detail in Table 4.3, where the MSE is used to measure the prediction accuracy of our proposed model, while R^2 indicates the high-

resolution and unblurred prediction frames. Here, the MSE is as low as 3% of the test data on average, and it is almost constant for all studied F values. In addition, the R^2 values imply that, in all prediction cases, we have accurately captured around 98 – 99% of the predicted TN variations throughout the GBR.

We always use a direct forecasting approach in the reported scenarios of this chapter. This means that we separately train the model for each forecasting horizon. However, it is also possible to employ the direct-recursive forecasting approach. In other words, we can use a previously predicted frame as input to predict the next frame and so forth.

To investigate how well our model predicts future values compared with simple propagation of the historical input values, a study is conducted in Fig. 4.8, where the MSE is employed as a mathematical distance metric to measure the pairwise distances between our prediction, true future values, and historical input values. To better comprehend this figure, a new Distance Index (DI) metric is defined as

$$DI = \frac{\text{dist}(\text{true future values, historical input values})}{\text{dist}(\text{prediction, historical input values})}, \quad (4.8)$$

where $\text{dist}(\cdot)$ stands for the MSE distance. As expected, the distance between future values (in F days later) and historical values (in the current day) increases with F . Besides, by increasing the forecasting horizon, our prediction broadens its distance from the input values. Also, the prediction keeps its constant distance with the true values, even for the case of a week ahead prediction, i.e., ($F = 7$). All these desirable distancing behaviours, keep the DI value close to 1.0 for all forecasting horizons.

Finally in Table 4.4, the performance of the proposed FE-DNN model with 7×7 elements is compared with both the Conv3D model by [322] and the ConvLSTM PhyDNet model by [324]. At the time of writing, PhyDNet [324] is ranked as the best video predicting model in multiple categories ([papers with code](#)). All the comparisons in Table 4.4 are conducted for the 2018 test dataset. As can be seen in this table, neither of Conv3D nor PhyDNet can accurately predict the TN dynamics in the wide GBR. The MSE is greater than 10 for larger F values and the Mean Absolute Error (MAE) is always greater than one, which results in $R^2 < 90\%$ and $DI \neq 1$.

The FE-DNN error bias values in Table 4.4 are close to zero, indicating unbiased predictions. Another performance metric in Table 4.4 is the scatter index, which is calculated in percentage by dividing the root-MSE by the mean of the true values in each day and expressing the result as a percentage. The near-zero scatter index of FE-DNN indicates a low relative error with respect to the mean TN.

The reliability analysis in Table 4.4 calculates the percentage of relative absolute errors that are less than 0.2 (according to the Chinese Standards) [338]. High reliability values

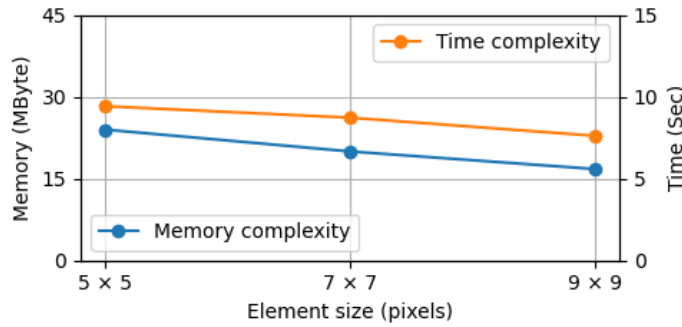


Figure 4.9: Time and memory complexities with respect to the element size for one-day forecasting horizon, with $N = 3$.

show that this approach is consistently accurate. U95 is another performance metric in Table 4.4, which is a type of uncertainty metric [338]. This metric considers the 95% confidence interval, and calculates the uncertainty range of models' predictions. The proposed FE-DNN model offers the lowest uncertainty values of the models evaluated, with 95% confidence.

We also conduct an F-test in Table 4.4 to analyze the variance of the forecasted TN. The F-test is a statistical test to find out whether the predictions and the true values have the same variance. The Null Hypothesis (H_0) is that the variances are equal. P-values greater than 0.05 reject the H_0 , indicating that variances are not equal. Based on this test, our model perfectly catches the variations in TN, while other models cannot adapt to the rapid TN changes in the wide GBR. It is worth reminding from Section 4.1 that both the Conv3D and PhyDNet next-frame predictors treat each frame as a whole. Consequently, they fall short against the proposed FE-DNN model that borrows the finite element concept from FEA, and uses the modified stiffness matrices to produce accurate predictions.

4.4.3 Computational Complexity

The proposed FE-DNN model was implemented using Keras APIs of TensorFlow in Python. The model was trained on a windows machine with Intel[®] Core i7-7700HQ CPU, NVIDIA[®] GeForce 1050 GPU, and 16 GB RAM.

The computational complexity is analyzed in terms of computational resource requirements, as well as the running times [298]. In this regard, we evaluate our model's demand when varying the input element size. This analysis for one-day TN prediction in the entire GBR is illustrated in Fig. 4.9. The results are obtained during inference, where the weights and biases are fixed. The memory in this figure refers to local RAM consumption, not the GPU memory in use. It also excludes the TN input and output data, and only

Table 4.5: Comparing time and memory complexities of the proposed FE-DNN with two recently published works in the literature

DNN Model	Time	Memory (MByte)
Conv3D by Mathieu et al. [365]	1.0 ms	17.06
PhyDNet by Guen and Thome [367]	13.0 ms	943.3
The Proposed FE-DNN model	8.7 s	19.97

includes the model variables.

The small reduction in resource demand when increasing the element size is due to an increase in the number of unpredictable marginal pixels. This results in a need for predicting fewer pixels by the model. Overall, the plots reveal almost a constant demand for both the simulation time and the memory, which suggest our model is efficient.

The time and memory demands of the proposed FE-DNN model with 7×7 elements are compared with those of the TensorFlow implementation of Conv3D [322] and the PyTorch implementation of PhyDNet [324] in Table 4.5. FE-DNN has a similar memory footprint to the contribution of [322], because both of them are based on the CONV3D neural networks. By contrast, the PhyDNet model demands larger RAM, as it is based on the ConvLSTM neural networks. One-day TN prediction in the entire GBR takes longer in the FE-DNN model. This is due to the fact that our model sweeps the study area pixel-by-pixel, while the other two models digest the whole input TN frame at once.

4.4.4 Ablation Study

In this subsection, an ablation study is conducted to better understand the impacts of different blocks of our model shown in Fig. 4.2. The ablation study calculates the overall accuracy of the model, when leaving a target block out of the structure [339]. To elaborate, we quantify the importance of any desired block, simply by omitting it from the ensemble of the proposed FE-DNN.

The results of the ablation study are shown in Table 4.6, using 7×7 elements over the test dataset in 2018. The MSE in the first row is calculated in presence of all the blocks, and it is averaged over the 12 months. As expected, the MSE increases in subsequent rows, by removing functional parts from the structural body. The difference between the MSE values in the first row and any other row is an indicator of the significance of the excluded block in the prediction performance of our proposed model.

Comparing all the MSE values in this table reveals the importance of the stiffness matrix in the next-frame analysis. Removing this FEA-inspired parameter reduces the

Table 4.6: Evaluating the elimination of selected blocks on accuracy of the one-day TN forecasting

Ablation study	MSE	MSE increment
Including all the blocks	1.64	0%
Excluding stiffness matrix	1.94	18%
Excluding the Conv3D layer	1.85	12%
Excluding the first dense layer (with 100 neurons)	1.76	7%
Excluding both the first and second dense layers	1.98	21%

model accuracy by 18%. Moreover, the overall effect of the four dense layers of MLP in Fig. 4.2 seems to be more significant than removing the Conv3D layer.

4.4.5 Limitations

Similar to other DNN-based models, the proposed FE-DNN is subject to some limitations. These limits are applicable when using the FE-DNN in other geolocations or employing it to forecast other environmental parameters. Some of these limitations are listed bellow.

- Our FE-DNN technique does not rely on PDEs or their solutions. As a DNN, FE-DNN learns the behavior of the underlying system only by looking at the training data. So, the main limitation of the developed model is the availability of training data. It is worth noting that, access to suitable training data is one of the main limitations of any DNN model.
- The current FE-DNN model is trained to forecast the TN distribution in the GBR. Applying the developed model to another study area requires re-training the model with local data, which must be available for each pixel on a daily time-step to achieve comparable results. Besides, the proposed model can be retrained regularly to take note of future challenges such as climate change and how they impact water quality.
- The proposed FE-DNN is only suitable for any spatio-temporal data that are gathered or interpolated in regular spatial nodes in regular time intervals. For example, the model can be applied to remotely sensed observation forecasting, only if its data are regularly interpolated in both the time and spatial domains to fill gaps due to sun-glint, clouds, or other observational quality issues. Though, this limitation will

be present for other similar forecasting models that require constant spatiotemporal training data.

- Computational time and memory resources are two important limitations of this model. To be able to train the model for the entire GBR in a decent time, we have downsampled the dataset into spatially 16-km wide pixels. In the absence of down-sampling, the lack of computation resources would impose a significant problem. However, this problem can be addressed by using a more powerful computing unit. This approach supports short-term forecasting of sediment concentrations in the GBR, complementing the capabilities of simulation models that are used to provide longer-term projections to support management and policy decisions to protect the GBR from land-derived pollutants and climate change impacts. While short-term forecasting does not support long-term policy decisions, it can be used to support short-term operational decisions, such as where and when to conduct in situ monitoring and process studies to improve the value of monitoring data.

4.4.6 Future Directions

Future research can involve actions either to address the limitations discussed in Section 4.4.5, or to enhance the capabilities of the proposed model. Some of these actions are discussed below.

FEA is a numerical technique to solve large-scale PDEs, arising in engineering and mathematical physics. FEA can also deal with arbitrarily shaped regions, as long as a discrete representation of the region (i.e., the meshing) exists [333]. Relying on the FEA concept, the proposed FE-DNN is applicable to almost any physical or environmental next-frame forecasting problem with ruling PDEs. While the next-frame prediction of TN is carried out in this contribution, and the next-frame prediction of sea surface temperature is conducted by [324], the proposed FE-DNN model can be applied, in future research, to many other environmental parameter such as heat transfer, water flow, small particle movements, etc.

Remote sensing data could be extremely useful in training our data-driven DNN-based model. While remote sensing TN data for the wide GBR is not available, ocean color algorithms have been developed to provide remote sensing observations of other water quality variables, including chlorophyll-a, total suspended sediments, Secchi depth, and benthic photosynthetically active radiation [340, 341]. There are some published works in the literature that have tried to estimate nitrogen distribution over wide areas using correlations with other remotely sensed environmental parameters. For example, [342] estimated nitrogen in southern Indian waters using remotely sensed sea surface tempera-

ture, and [343] estimated nitrogen in the coastal regions of East China Sea using remotely sensed sea surface salinity and sea surface reflectance. Having said that, an accurate algorithm for retrieving TN levels at oceanic scales in optically diverse waters from remote sensing observations has yet to be developed.

Finally, training of computationally expensive DNN solutions like the FE-DNN requires a variety of hardware resources. This demand for computational resources can be handled by existing parallel processing techniques, e.g., the shared-memory multiprocessors or the Distributed Computing Systems (DCS) [24]. GPUs, FPGAs, and multi-core CPUs are few examples of the shared-memory parallelization techniques. DCS, on the other hand, consist of a network of cooperating computers that offer high-performance data processing. Using an on-premise DCS or a cloud-based distributed computing service like Amazon AWS, Microsoft Azure, etc. can be the next step for FE-DNN implementation research.

4.5 Conclusion

Inspired by the well-known FEA, we proposed the FE-DNN model for next-frame prediction of physical parameters in wide spatial coordinates. Our model is applicable to any environmental modelling scenarios, which are governed by underlying PDEs. We applied our novel model to the problem of TN distribution prediction in the GBR. To the best of our knowledge, our study is the first to use a data-driven machine learning approach for nitrogen prediction in the GBR. One challenge in training our DNN-based model is the scarcity of observational TN data in the GBR. To address this problem, we employed the PINN technique to merge the large amounts of simulated data with the sparse measurement data. This enabled us to successfully train our proposed FE-DNN model for TN forecasting. The performed analyses revealed that our next-frame predictor model achieves a very high accuracy with a low prediction MSE, while yielding high-resolution prediction frames with very high R^2 values. The calculated R^2 metric was more than 98%, resulting in unblurred TN prediction frames in the entire GBR. We believe that our model and this study can be beneficial and support internally significant water quality programs like the Australia's Reef 2050 Plan. This can help improve ecosystem recovery and resilience by informed decision making based on accurate prediction modelling. Furthermore, it can be adopted by existing hindcasting simulators to provide accurate forecasting predictions.

Chapter 5

Finite Element Transformer for Sediment Distribution Prediction in the Great Barrier Reef

Suspended sediment is a significant threat to the Great Barrier Reef (GBR) ecosystem. Existing sediment forecasting methods suffer from the problem of low-resolution predictions, making them unsuitable for wide area coverage. In this chapter, a novel sediment distribution prediction model is proposed to augment existing water quality management programs for the GBR. For model training, the emerging physics-informed neural network is employed to incorporate both simulated and measured sediment data. This chapter is published in *Neural Networks* as

- [27] M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sediment prediction in the Great Barrier Reef using vision Transformer with finite element analysis,” *Neural Networks*, vol. 152, pp. 311–321, Aug. 2022.

5.1 Introduction

The Great Barrier Reef (GBR) in Australia is a world heritage site and the world’s largest coral ecosystem. Coral reefs provide jobs and foods for over half a billion people worldwide, protect coastlines from storms and erosion, offer opportunities for recreation, and source new medicines [344]. Unfortunately, these ecosystems are under great pressure both by natural events and human activities. These threats include storms, floods, man-made pollution, and warmer ocean temperatures that can stress corals, leading to coral bleaching or physical damages [345].

The GBR water quality has been declining over the last 150 years [346]. Sediments, nutrients, and pesticides are the main coral reef pollutants [300], which result in poor water quality that is believed to be one of the main contributors to the current adverse situation in the GBR coastal and marine areas [300].

More specifically, sediments have significant impacts on the GBR ecosystem. Suspended sediments eventually settle down on seafloor in a process called sedimentation. Then, nutrients and other pollutants attached to sediment release in the underwater environment, which in turn, limits coral recruitment, health, and productivity [302]. Suspended sediments can also reduce sunlight available for seagrasses, algae, and other plants' photosynthesis and reduce their growth.

GBR sedimentation is primarily attributed to the land-based sediments from 35 catchments into the GBR lagoon [346]. Among all these catchments, those with higher levels of land clearing show more contribution to the total GBR sedimentation. To elaborate, 70% of river-based sediments stem from 20% of GBR catchment. This suggests that much of the sedimentation problem can be mitigated by managing a relatively small area [347].

Management of sediment and nitrogen loads to the GBR and improving its water quality has been the focus of major investments for many years [310,311]. According to literature, the coastal waters of the GBR have noticeably lost their quality due to changes in their catchment land use since European settlement [306]. Besides, GBR sedimentation has been particularly elevated during flood events [348], which has been linked primarily to the land-use practices and reduced vegetation cover in QLD catchments [346].

These sediments include minerals, muds, dusts, and granite from soil erosion, as well as white calcium carbonate (CaCO_3) from coral erosion [325]. The sum of all these components is termed the Total Sediment (TS). Any TS value over the 2 mg/L threshold can adversely affect marine ecosystems and corals in open coastal and mid-shelf waters [349]. To better control adverse sediment erosion, a long-term water quality improvement plan was established by the Australian Government, dubbed *The Reef 2050 Plan* [301]. This plan aims to better manage the relevant catchments and to reduce run-off pollutants into the GBR.

However, implementing an accurate TS predictor is not an easy task, attributed to the high dynamics of sediment processes in the wide GBR region. These processes start from catchments in the vicinity of river discharge areas, where bulk masses of sediments settle on the sea bed of each river mouth, extending tens of kilometers along the coastline. These sediments are then dispersed by waves and relocated by ocean currents to remote GBR areas, and partly buried into deeper benthic sediments [346].

To understand the TS dynamics and to model its complex behaviour, a few nonlinear ocean models have been developed. For example, *eReefs* is a comprehensive suite of hydrodynamic, biogeochemical, and sediment transport models, which was especially designed for the GBR [299]. This suite of models provide access to historical environmental variables, and offer limited management options to mitigate risks associated with its pre-defined scenarios [350]. The sediment transport model in *eReefs* can analyse the fate of

suspended sediments in the GBR [299].

However, *eReefs* is a computationally expensive model, which has a very complex calibration process and may take weeks to simulate any desired ocean sedimentation. It also lacks the much-desired capability of forecasting. This means that *eReefs* cannot be used to forecast future values of TS distribution [350]. Therefore, *eReefs* and other similar modelling suits have limited capacities to underpin decision-support for the spatially vast GBR. This motivates demands for a data-driven deep learning model to predict future sediment distributions as fast and accurate as possible.

To meet the above demand, many studies have been carried out to design an accurate sediment prediction model [351–353]. For example, [351] designed a shallow neural network for sediment lead (Pb) prediction for two Australian bays. In other examples, [352] and [353] developed an Adaptive Neuro Fuzzy Inference System (ANFIS) and a shallow neural network model to predict the sediment distribution in river open-channel flows, respectively.

Thus far, all the aforementioned models have avoided the problem of next-frame sediment distribution prediction. These models simply average sediment in their study areas, making them the well-studied timeseries prediction models. However, many applications require an accurate distribution prediction of their parameters of interest across an area. This distribution forecasting is known in parlance as *next-frame* prediction, which has evolved from the former timeseries prediction models [318].

The existing next-frame forecasting models mainly utilize 3D Convolutional Neural Networks (Conv3D) [322] and Convolutional Long Short-Term Memory (ConvLSTM) [324]. These models result in blurred prediction frames with low coefficient of determination. To address this problem, we propose a novel Deep Neural Network (DNN) to predict the next-frame suspended sediment distribution in the GBR. This model is based on the state-of-the-art Transformer network [354].

Transformers are a new class of DNNs that were initially developed for tackling time-series and sequential problems [355]. Traditional timeseries processing models such as the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) suffer from the following drawbacks:

- *Long-term dependencies*: RNNs are unable to explore long-term correlations within data, due to the so-called vanishing gradient phenomenon. This problem is alleviated by the gating mechanism in LSTMs and GRUs to some extent. However, RNN-based networks are still incapable of addressing wide time-gaps in long-term dependencies [356].
- *Parallelization*: All RNN-based models are limited to directional data, where the

value in time t is strictly dependent on the value in time $t - \Delta t$. This directional property limits the model’s capability for parallel processing, distributed processing, etc. [357].

To address the above two significant limitations, an *attention* mechanism was initially proposed by [358]. This mechanism models long-term dependencies without considering the entity location in the sequence, which can better extract long-term dependencies than RNN-based models [359]. Additionally, this non-recurrent mechanism consists of multi-head attention and Multi-Layer Perceptron (MLP) layers, which make it a good candidate for parallelization. This initial self-attention architecture was used in the first Transformer model for natural language processing, by [354]. [360] later extended the Transformer network to process images by proposing the so-called Vision Transformer (ViT). The ViT is the model of choice in this chapter, which is used in conjunction with concepts of Finite Element Analysis (FEA) to accurately predict the next sediment frame in the entire GBR.

The proposed FE-Transformer takes as input the TS distribution frames (matrices) in the past N days, and forecasts the TS distribution frame in F days later. However, the observational TS data in the wide GBR are scarce and insufficient for training data-hungry DNNs. To address this problem, we employ a technique to merge sparse TS measurements with the readily available simulated TS data (e.g., *eReefs*). This approach proves effective in training our proposed FE-Transformer network, which demonstrates an excellent TS prediction performance across the GBR, when compared with state-of-the-art next-frame prediction approaches.

The remainder of this chapter is organized as follows. Section 5.2 proposes the FE-Transformer as a novel solution to the problem of sediment distribution prediction in the GBR. The physics-informed neural network technique for measured and simulated data fusion is described in the remaining of this section. We evaluate the accuracy of our FE-Transformer model for next sediment frame forecasting in Section 5.3, where a detailed investigation of the computational complexity and an ablation study are also carried out. The chapter is concluded in Section 5.4.

5.2 Proposed Model

As discussed in the preceding section, increased sedimentation since European settlement reduces water quality and imposes a major threat to the health of GBR ecosystems [308]. This makes TS improvement important through local management actions, which in turn improves the resilience of GBR ecosystems. To support such actions, it is important to be able to monitor and accurately predict sediment on the scale of the whole GBR.

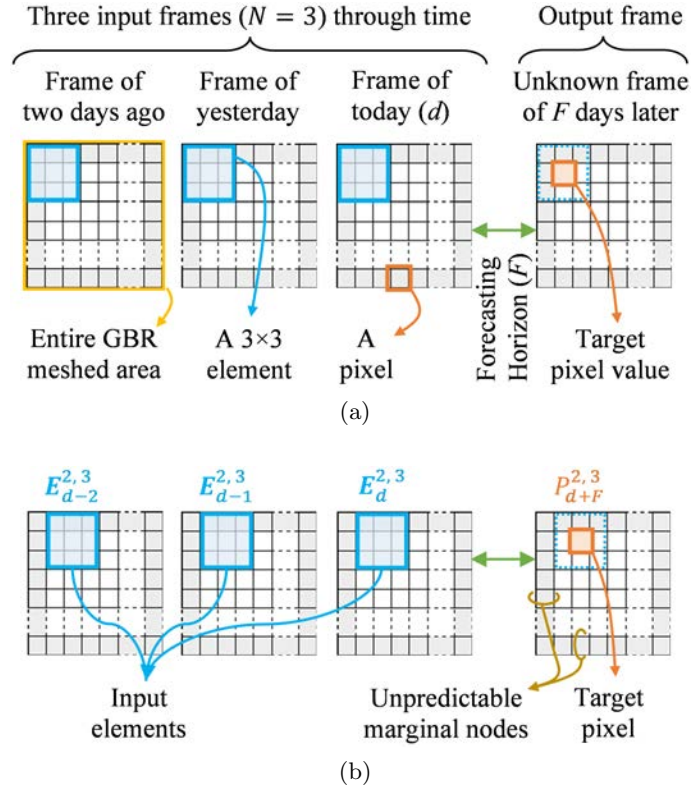


Figure 5.1: Three input frames ($N = 3$) are used to predict the next TS frame in F days after today (d). Towards this end, the pixels of historical TS values in the GBR are grouped into 3×3 elements $E_{d-2}^{x,y}$, $E_{d-1}^{x,y}$, and $E_d^{x,y}$. The central coordinate (x, y) of these historical elements will be adjusted to predict TS at (a) pixel $(2, 2)$ (i.e., $P_{d+F}^{2,2}$); and (b) pixel $(2, 3)$ (i.e., $P_{d+F}^{2,3}$), while leaving a 1-pixel margin of unpredictable TS values. The 3×3 element size is used here for illustration purposes only. This size can be optimized to better address the requirements of TS prediction in the GBR.

The problem of accurate TS distribution prediction in GBR requires a sophisticated next-frame predicting model. The state-of-the-art next-frame predictors are surveyed by [318], where a collection of convolutional neural networks, long short-term memory, generative adversarial network, and gated recurrent unit DNNs along with their possible combinations are benchmarked.

A common weakness of all these next-frame predictors lies in their inability to learn input variations [318], which results in blurred predicted frames. In other words, these models try to reduce output frame dynamics with the objective of minimizing prediction errors. As a result, the forecasted output frame suffers from low variance and smoothed details. To address this problem, we propose to combine the FEA mechanism with the

state-of-the-art Vision Transformer network to create a highly-accurate TS distribution predictor.

5.2.1 Finite Element Analysis

The FEA is a numerical method for solving Partial Differential Equations (PDE) in engineering. This method is extensively used in simulating physical phenomena, by finding current transient responses to the current transient inputs. A FEA system is intrinsically unable to address forecasting scenarios in which one predicts future responses to the given historical inputs. In other words, FEA-based forecasting systems rely heavily on their coupled surrogate models to estimate future boundary conditions. However, our proposed FE-Transformer integrates the forecasting DNN techniques with the FEA to accurately predict unknown future TS values.

Similar to other hydrodynamic phenomena, sediments dynamics can be described and modeled by some governing PDEs when being suspended, transported, and deposited by ocean waves and flows [350]. As illustrated in Fig. 5.1a, the TS distribution for each day forms a *frame* that covers the entire GBR. By knowing the past N frames before today (d), the goal is to predict the future TS frame in F days later (forecasting horizon).

To accurately predict a future frame, the FEA concept can be employed by dividing the large GBR area into *pixels* and *elements*. Every pixel P retains the average TS in a 16 km^2 area in the GBR. For each given coordinate (x, y) , the pixel in that coordinate and several surrounding pixels are grouped into a small square termed an *element*. The historical elements across $\mathbf{E}_{d-N+1}^{x,y}, \mathbf{E}_{d-N+2}^{x,y}, \dots, \mathbf{E}_d^{x,y}$ constitute the known inputs to the model, which yields the unknown pixel $P_{d+F}^{x,y}$ as output. As shown in Figs. 5.1a and 5.1b, the input elements overlap and sweep to cover all pixels \mathbf{P} inside the TS distribution frame in the GBR, leaving the half element size unpredictable in margins, e.g., one pixel for 3×3 elements and three pixels for 7×7 elements.

After discretizing the study area into elements, the FEA formulation results in the following abstract system of algebraic equations [361]

$$\mathbf{P} = f(\mathbf{K}, \mathbf{E}), \quad (5.1)$$

where \mathbf{K} is the *stiffness matrix*. This matrix is a core parameter in FEA, which relies on scientific understanding of the underlying physics of the targeted system. In Section 5.2.2, we propose a new method to numerically calculate \mathbf{K} using the training dataset, where no prior knowledge of the TS dynamics in the GBR is required.

Furthermore, unlike the FEA that resorts to matrix algebra to solve (5.1), we feed both \mathbf{K} and \mathbf{E} to our proposed machine learning model. As a result, the governing equation in

(5.1) changes to

$$\mathbf{P} = \mathbf{W} f(\mathbf{K}, \mathbf{E}), \quad (5.2)$$

where \mathbf{W} represents the unknown weights of the supervised machine learning model. Using this innovative approach, the model will learn TS variations in each element, using the historical TS values for that element and its corresponding stiffness matrix. It is worth mentioning that the recursive technique for solving (5.1) is very similar to the gradient descent technique used by machine learning to solve (5.2).

5.2.2 Stiffness Matrices

In the FEA, a stiffness matrix denotes an approximate solution to underlying PDEs by relating nodal responses to external nodal forces [361]. Similarly, the stiffness matrix of an element in the GBR represents its physical properties and relates TS variations to its historical values. Having said that, solving the stiffness matrix for ocean sedimentation requires a deep understanding of GBR multiphysical behaviours, including fluid flow, mass transport, etc. [299]. To address this challenge, an innovative approach is proposed to numerically compute the stiffness matrix \mathbf{K} .

Consider a simple linear spring problem illustrated in Fig. 5.2a, where a linear spring with $k = 10$ acts against external forces F_1 and F_2 , and it compresses to the X_1 extent. We simulate this system for two seconds with 40 samples (20 samples per second) using sine waves and random Gaussian noise. By dividing this simulation time into four segments (r), we have 10 samples per segment. This way, the true stiffness matrix for each segment can be numerically estimated by

$$\mathbf{K}_r \mathbf{X} = \mathbf{F} \Rightarrow \mathbf{K}_r = \frac{\mathbf{F}}{\mathbf{X}} \xrightarrow{\text{In matrix algebra}} \mathbf{K}_r = \mathbf{F} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}, \quad (5.3)$$

where $\mathbf{F}_{2 \times 10}$ is the matrix of 10 F_1 and F_2 values, and $\mathbf{X}_{1 \times 10}$ is the matrix of 10 X_1 values in one segment. The final stiffness matrix \mathbf{K} of the linear spring problem can then be calculated by averaging \mathbf{K}_r as follows

$$\mathbf{K} = \frac{1}{4} \sum_{r=1}^4 \mathbf{K}_r. \quad (5.4)$$

The simulation results in Fig. 5.2a suggest 96% accuracy in numerical estimation of \mathbf{K} . However, adapting this technique to our TS prediction problem requires the following adjustments:

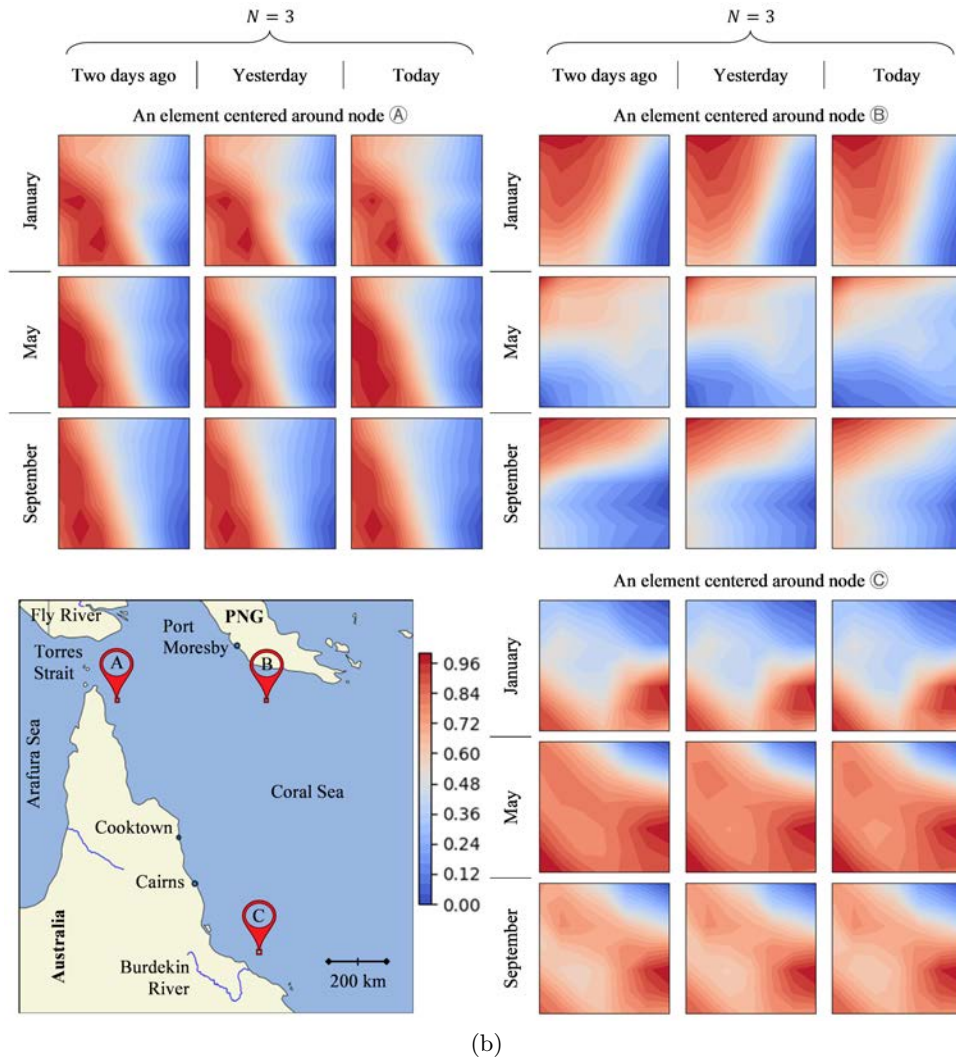
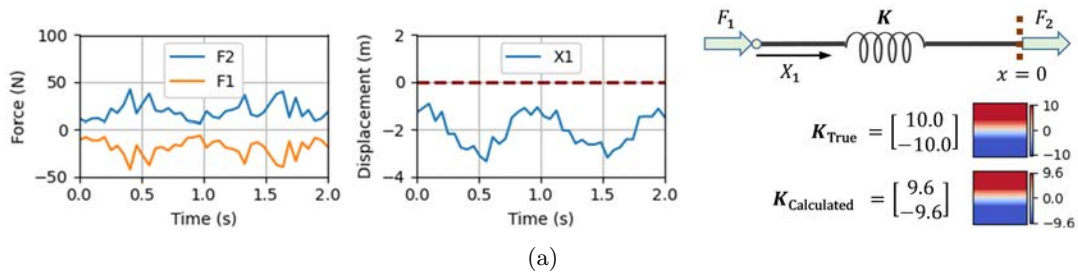


Figure 5.2: (a) Stiffness matrix of the classic linear spring problem in the presence of Gaussian noise. (b) Normalized logarithmic scale stiffness matrices of three typical geolocations: A at (11.18°S, 143.25°E), B at (11.18°S, 148.17°E), and C at (19.10°S, 147.93°E), where the element size is 7×7 .

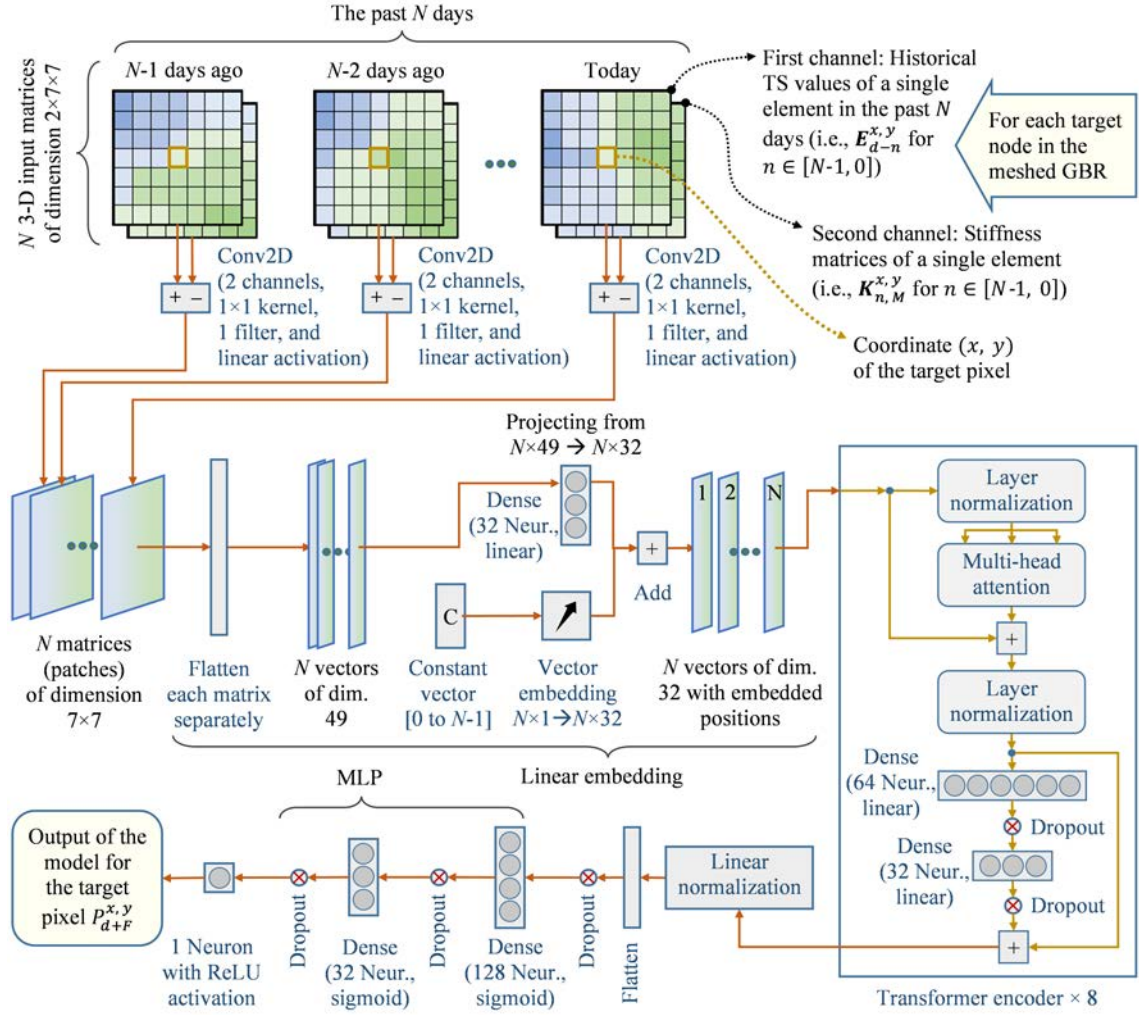


Figure 5.3: Architecture of the proposed FE-Transformer network to predict pixel $P_{d+F}^{x,y}$ in F days after today (d), where the N input elements are illustrated in Fig. 5.1, and the N stiffness matrices for month M are calculated by (5.5). The shown kernel sizes, filter numbers, activation functions, Transformer layers, etc. are optimized for TS prediction in the GBR.

1. As stated earlier, the number of input elements to our model is N , which is composed of elements from today (d) to N days ago. Consequently, we must have N stiffness matrices for each of these input elements;
2. To better capture the TS distribution, we mesh the GBR into small spatial elements. Furthermore, to properly capture temporal TS changes, we need to take a further step by dividing the study period into smaller time steps, e.g., months. This temporal division leads to a dedicated set of stiffness matrices for each month, which results

in a more accurate prediction of marine sedimentation. In this regard, smaller time steps will increase memory consumption, while larger time steps will reduce accuracy;

3. Different from (5.1) in conventional FEA that places the stiffness matrix \mathbf{K} in an equation with \mathbf{E} , we propose to feed \mathbf{K} as an additional input to our DNN. This way, the variations in each TS element will be learned in line with its corresponding stiffness matrix; and
4. The Conv2D layers in Fig. 5.3 implement $\mathbf{E} - \mathbf{K}$. This is in contrast to the original FEA implementation in (5.3), where $\mathbf{X} = \mathbf{F}/\mathbf{K}$. Therefore, in our proposed model, \mathbf{K} is calculated by subtracting the known output P from element \mathbf{E} , i.e., $\mathbf{E} - P$.

Considering the above adjustments, we can rewrite and merge (5.3) and (5.4) into

$$\mathbf{K}_{n,M}^{x,y} = \frac{1}{N_y} \frac{1}{N_d} \sum_y \sum_d (\mathbf{E}_{d-n}^{x,y} - P_{d+F}^{x,y} \mathbf{J}) \quad \forall n \in [0, N-1], \quad (5.5)$$

where M represents the corresponding month of the year, y sweeps all N_y years in our training dataset, d sweeps all N_d days of month M , and \mathbf{J} is an all-one matrix of the same size as our elements. After calculation of the stiffness matrices using the training dataset, the resulting \mathbf{K} matrices will become independent of both year and day, and will be a function of coordinates (x, y) , month M , and their time difference $n \in [0, N-1]$ from today.

Using the new equation in (5.5), the stiffness matrices of January, May, and September are plotted in Fig. 5.2b for three random locations in the GBR. For these matrices, the number of inputs elements considered is 3, i.e., $N = 3$, while the forecasting horizon is $F = 1$. These settings result in three matrices $\mathbf{K}_{2,M}^{x,y}$, $\mathbf{K}_{1,M}^{x,y}$, $\mathbf{K}_{0,M}^{x,y}$ per location per month, denoted by *two days ago*, *yesterday*, and *today*, respectively. In the normalized contour plots of this figure, 0 means the maximum negative difference from the central pixel today, 1 means the maximum positive difference, and 0.5 means no difference.

As a result, the stiffness matrices such as those shown in Fig. 5.2b reveal the overall behavior of TS in the GBR. For instance:

- TS in point ① is influenced by its vicinity to the water flows in the Torres Strait;
- TS in point ② is influenced by both the Fly River and Papua New Guinea (PNG) catchments; and
- TS in point ③ is under influence of the Burdekin River.

5.2.3 FE-Transformer Architecture

In the previous section we explained that, in our proposed FE-Transformer the GBR is meshed into distinct pixel-level geolocations (x, y) , and in each location there are N elements representing historical TS values around that location (i.e., $\mathbf{E}_{d-n}^{x,y}$ for $n \in [0, N - 1]$). We then calculate N relevant stiffness matrices $\mathbf{K}_{n,M}^{x,y}$ for each month M . Next, we use these two inputs in (5.2) to predict the TS distribution \mathbf{P} . This prediction is conducted in a pixel-by-pixel manner, by calculating the TS in each coordinate (x, y) (i.e., $P_{d+F}^{x,y}$).

The $f(\cdot)$ operator in (5.2) is implemented by using a novel DNN structure shown in Fig. 5.3. To elaborate, we employ the state-of-the-art ViT [360] to propose a new FEA-inspired DNN dubbed FE-Transformer for next-frame prediction of sediment in the GBR. As shown in Fig. 5.3, this process starts by merging N historical TS elements with their corresponding stiffness matrices using N two-dimensional convolutional layers (Conv2D). Feeding \mathbf{K} as a separate input to the DNN model is in contrast to the conventional FEA that multiplies an inversed stiffness matrix with the excitation TS element values. Besides, the attention mechanism in Transformer networks have limited capacity to exploit the sequential nature of the input. Using N Conv2D layers to merge historical elementwise TS values with their corresponding stiffness matrices will help the subsequent Transformer layers by capturing part of the temporal relations inside the elements.

The resulting N patches are then flattened before being fed to the next layer, where they are projected onto N new vectors with a reduced size of 32. These vectors are then embedded (added) with N position vectors. This allows the model to learn the day-by-day ordinal behavior, as well as the natural moving directions in the TS distribution.

Next, the resulting sequence of positional embedded vectors enter eight cascaded standard Transformer layers, as originally described in [354]. The output of the last Transformer layer is flattened and enters an MLP network to predict the TS value of the target pixel. To improve the convergence of weight learning and to avoid overfitting, we have

- Multiple dropout points with a 20% dropping rate are placed in the layer intervals in Fig. 5.3;
- All the dense layers are equipped with the Ridge regularization of $L2 = 0.01$; and
- The learning rate is set to 0.001.

5.2.4 Physics-Informed Neural Network

To train our FE-Transformer model, we need daily TS data, i.e., P_{Measured} , measured for each pixel in the GBR. However, the existing P_{Measured} datasets are intermittently

observed and sparsely distributed. This scarce dataset is not suitable to sufficiently train DNNs and our Transformer network.

[362] recently introduced a novel Physics-Informed Neural Network (PINN), which can be used to merge measurements with PDE simulations for DNN training. PINN works by summing two separately calculated measured and simulated loss functions. To elaborate, suppose that we use the scarcely measured data to calculate the Mean Squared Error (MSE) loss function as follows

$$\text{MSE}_{\text{Measured}} = \frac{1}{N_{\text{measured data}}} \sum (P - P_{\text{Measured}})^2, \quad (5.6)$$

where P is the output of our model in Fig. 5.3, and $N_{\text{measured data}}$ is the number of used data points. At the same time, PDEs can be solved to simulate the TS concentration distribution in the GBR (i.e., P_{PDE}) for the following loss function

$$\text{MSE}_{\text{PDE}} = \frac{1}{N_{\text{PDE data}}} \sum (P - P_{\text{PDE}})^2. \quad (5.7)$$

The MSE loss functions in (5.6) and (5.7) are then combined to create the following overall loss metric

$$\text{MSE} = (1 - \lambda) \text{MSE}_{\text{Measured}} + (\lambda) \text{MSE}_{\text{PDE}}, \quad (5.8)$$

where λ is an adjustable hyperparameter that is used to implement a weighted sum to balance between the measured and PDE simulated data.

The MSE loss function in (5.8) is used to train the proposed FE-Transformer. It is worth mentioning that whenever $\text{MSE}_{\text{Measured}}$ is not available, it is removed from (5.8).

5.3 Results and Discussions

This section starts by introducing the employed TS datasets in the GBR. We will then proceed with evaluating the accuracy of the proposed FE-Transformer for predicting the TS distribution. We will also study the computational complexity of our model and undertake an ablation study for a block-level evaluation of our model's structure.

5.3.1 Data Sources

According to (5.6) and (5.7), two sets of measured (P_{Measured}) and simulated (P_{PDE}) data are required to train the FE-Transformer in conjunction with PINN methodology. The measured TS values stem from the Australian Institute of Marine Science (AIMS) Marine Monitoring Program (MMP) [335]. This data is sparsely gathered, both temporally and

Table 5.1: Statistics of the TS Values from *eReefs* (P_{PDE})

Dataset	Year	Min	Max	Mean (μ)	STD (m)
Training and Validation	2011	0	0.34	104.05	1.19
	2012	0	0.26	80.44	1.25
	2013	0	0.28	122.26	1.82
	2014	0	0.32	91.21	1.48
	2015	0	0.30	84.12	1.53
	2016	0	0.20	67.13	1.10
Testing	2017	0	0.50	104.08	1.88
	2018	0	0.30	91.22	1.39

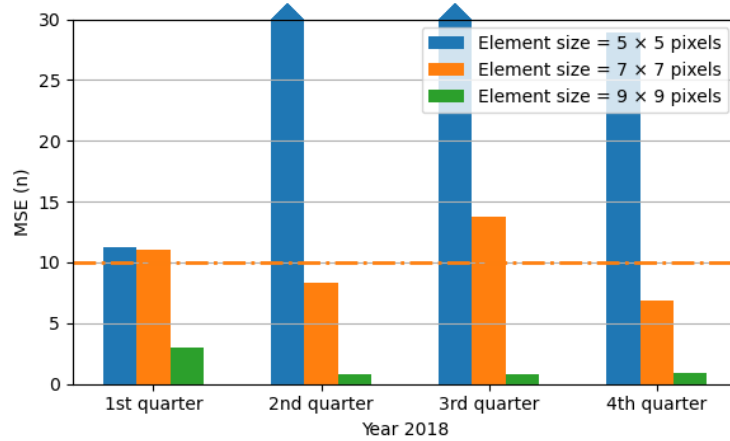
spatially, and thus is insufficient for DNN training purposes.

Hence, simulated data (from solving PDEs) is used to compensate for the scarcity of the measured data. To obtain the PDE solutions for the TS distribution in the GBR, the *eReefs* modeling suite is employed, which is a suite of models designed to deliver a near real-time coupled biogeochemical, hydrodynamic, and sediment transport simulation of the GBR [325]. This raw simulation data has been cleaned and interpolated onto a regular grid and is downloadable from the AIMS servers [337].

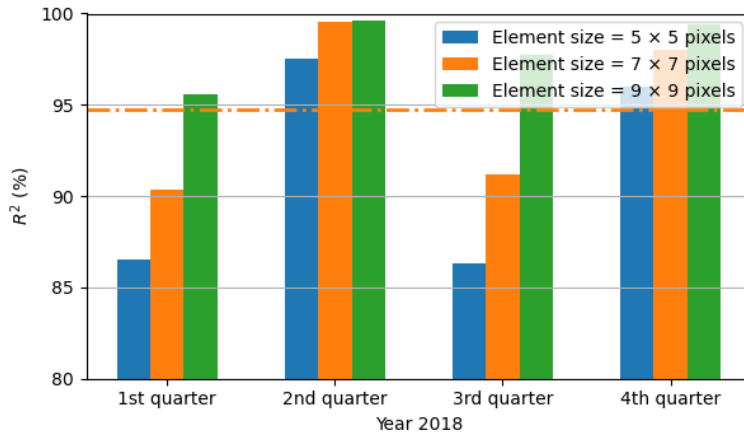
The sediment simulation in *eReefs* is conducted for multiple layers, from which the accumulated sedimentation in the top 3 m surface is used in this chapter. In addition, the AIMS version of *eReefs* is calculated on a daily basis during 2011 to 2018. We divide this period into three timespans. The first one from 2011 to 2016 is used as the training dataset, 2017 is used as the validation dataset, and 2018 is the testing dataset. The model weights are first calibrated using the training dataset and, without changing the calibrated weights, accuracy is evaluated by the validation datasets. Finally, the overall accuracy of the model is measured by the testing dataset, which is not *a priori* known to the model.

Overall, after meshing the whole GBR into 150×122 pixels, we would have 18,300, 17,760, and 16,704 overlapping 1×1 , 3×3 , and 7×7 elements, respectively, in each day. Accumulating these elements in 365 days of 7 years in our training dataset results in 47, 45, and 43 million training patch images (i.e., elements). According to [360], ViT-based models can benefit from this large dataset to gain higher accuracy, compared to traditional CNN architectures.

To better understand the nature of TS simulation in *eReefs*, the statistics of TS values are presented in Table 5.1. As can be seen from this table, the maximum TS values are far greater than their average values. This is because the TS concentrations within river deltas



(a)



(b)

Figure 5.4: Effect of element size on the metrics of (a) MSE and (b) R^2 with 3-day forecasting horizons and $N = 3$. The results are averaged per quarter of the testing year of 2018.

are usually very high, while in many other places these concentrations can be very low. As a result, the ecologically important TS variations throughout the near-shore regions and out to the Mid-shelf waters may look insignificant. To capture these small variations, we use the logarithmic scale for model training as follows

$$TS_{\log} = \log_{10} (TS + 10^{-7}), \quad (5.9)$$

where the constant 10^{-7} in (5.9) is added to avoid $\log(0)$.

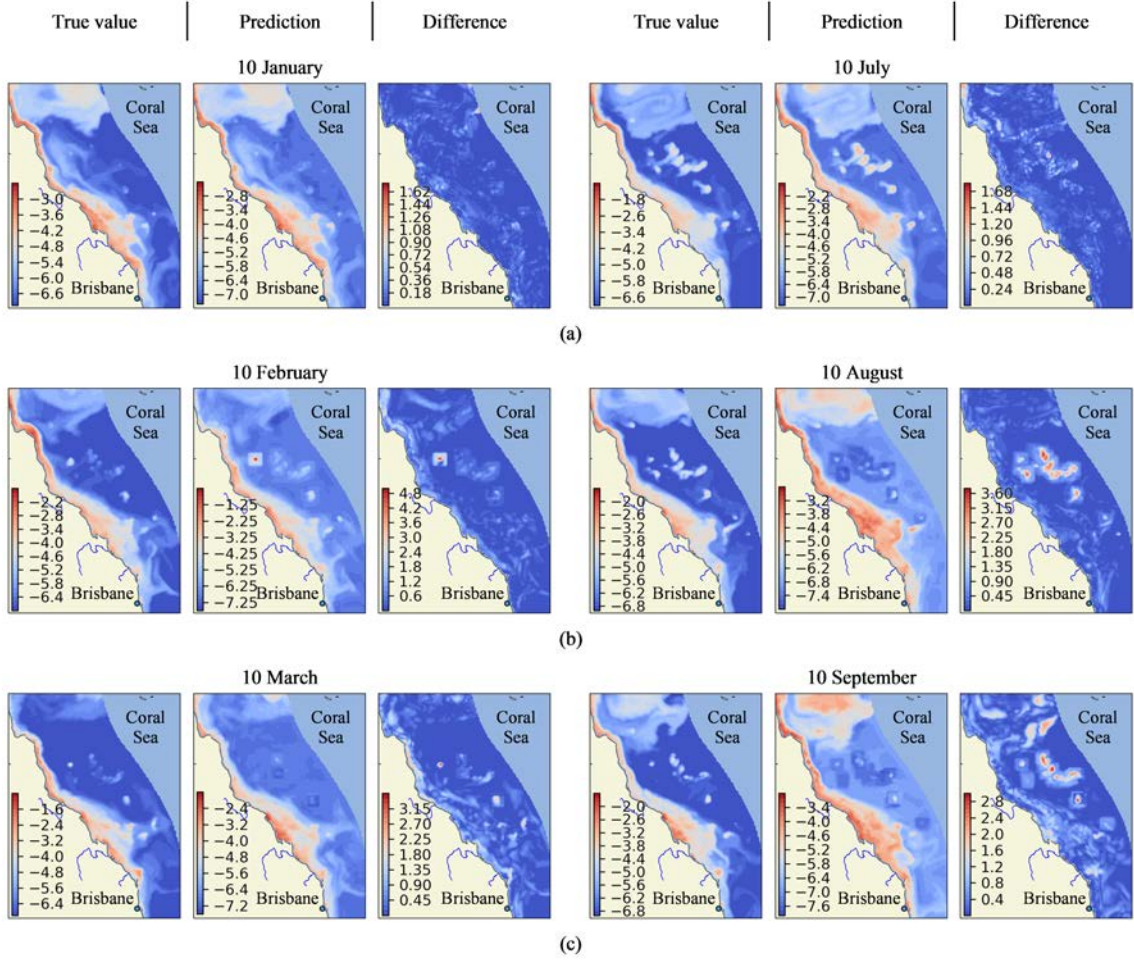


Figure 5.5: True values, predictions, and their absolute differences for TS_{\log} in the GBR at 2018 test dataset, for (a) 1-day, (b) 3-day, and (c) 7-day forecasting horizons, using 7×7 elements with $N = 3$.

5.3.2 Prediction Accuracy

The performance of the proposed FE-Transformer on TS distribution prediction in the GBR is studied in this section. We show that our model provides accurate predictions, as well as unblurred output frames, which are two desirable properties of any next-frame predictor.

To measure accuracy, we use the MSE metric in (5.8). Meanwhile, the coefficient of determination (R^2) is adopted as an indicator of our unblurred output, which is calculated by

$$R^2 = 1 - \frac{\sum (P - P_{\text{True}})^2}{\sum (P_{\text{True}} - \mu_{P_{\text{True}}})^2} \quad (5.10)$$

Table 5.2: MSE and R^2 of TS Prediction for the Testing Dataset in 2018, using Elements of Size 7×7 Pixels

Month	$F = 1$	$F = 3$	$F = 7$	$F = 1$	$F = 3$	$F = 7$
	MSE (n)			R^2 (%)		
Jan	2.95	7.58	14.30	98.05	88.44	97.12
Feb	0.24	0.71	7.11	98.67	94.31	76.78
Mar	3.82	27.68	23.59	99.15	87.85	97.55
Apr	0.05	2.33	5.16	99.96	98.99	97.79
May	9.34	21.92	15.60	99.95	99.81	99.85
Jun	0.35	0.70	11.19	99.96	99.91	99.74
Jul	0.42	1.22	4.73	99.39	96.34	79.13
Aug	0.35	2.39	3.27	99.06	87.03	96.04
Sep	17.34	37.66	22.23	99.26	90.11	98.42
Oct	0.03	0.15	1.75	99.68	98.86	76.20
Nov	0.14	0.77	1.37	99.52	97.04	87.24
Dec	59.37	19.63	27.55	98.87	98.22	94.65
Mean	7.87	10.23	11.49	99.29	94.74	91.71

where P_{True} stands for either P_{Measured} or P_{PDE} , and $\mu_{P_{\text{True}}}$ is the average of P_{True} . Here, R^2 shows how much of the true TS variation has been captured in the predicted output frame. This metric is also known as *goodness of fit*, where the larger is better, meaning even low levels of TS variation within a frame are captured.

As stated earlier in Section 5.2, the element size is a hyperparameter that requires adjustment. Towards this end, the effect of the element size on the MSE and R^2 metrics is investigated in Figs. 5.4a and 5.4b, respectively. As a rule of thumb, increasing the element size will lead to an improved model performance, but it also increases the number of unpredictable marginal pixels in Fig. 5.1. For consistency, 7×7 elements are used in our simulations. The horizontal dash lines in Fig. 5.4 indicate the average metric values for the 7×7 elements.

In addition to the element size, other hyperparameters of the proposed FE-Transformer are also optimized. These hyperparameters include but are not limited to the Conv2D kernel sizes and filter numbers, neural network activation functions, number of Transformer layers and their internal variables, etc. The hyperparameter optimization helps the model accurately forecast the TS in the GBR. This also tunes the number of model weights to

the optimum value, which in turn, assists the model to avoid overfitting and underfitting.

Some typical logarithmic-scale TS prediction results of the FE-Transformer model in 2018 (i.e., the test dataset) are plotted in Fig. 5.5. In the cases of the 1-day, 3-day, or 7-day forecasting horizons, the (7th, 8th, 9th), (5th, 6th, 7th), or (1st, 2nd, 3rd) days of each month are fed to the model, while the 10th day of the months are forecasted. This figure includes the predicted frames, ground-truths, and their absolute differences (i.e., the prediction errors), which are mostly close to zero. The prediction errors of larger F values have further spread geographically, resulting in a larger MSE. Besides, due to the adoption of the element size of 7×7 , three unpredictable marginal pixels are imposed along the shoreline. These pixels are predicted with a linear regression model.

It is worth mentioning that classic image processing techniques for compensating for the unpredictable margins (e.g., padding, flipping, etc.) do not suit the TS distribution prediction problem considered in this thesis. This mainly is due to the fact that current TS values in the GBR are dependent on their neighbors' historical TS values, which are asymmetric, chaotic, and dynamic [332]. Therefore, if these historical neighboring values are not available, we should not simply fill them with constant padding, symmetrical flipping, or other normal paddings. Besides, the unpredictable margin is rather small and has a minimum effect on the overall modeling performance.

Both the MSE and R^2 metrics of the proposed FE-Transformer model are investigated in Table 5.2. The reported metrics in this table do not include the unpredictable margins. Here, the average MSE increases from 7.87×10^{-9} for the 1-day forecasting horizon to 11.49×10^{-9} for the 7-day forecasting horizon. The R^2 values remain above 90% for all F values, which imply that our model is able to accurately capture the majority of the TS variations. Let us emphasize that this highly-accurate and unblurred prediction is achieved throughout the wide GBR with more than 35 river discharges [299], which makes the prediction even more challenging.

Since our model is a next-frame predictor, in Table 5.3, we compare its performance with two of the recent next-frame prediction methods, i.e., a Conv3D-based model [322] and a Convolutional LSTM model named PhyDNet [324]. The comparison is conducted on the 2018 test dataset with 7×7 elements. At the time of writing, PhyDNet [324] is ranked as the best next-frame video predicting model in multiple categories ([papers with code](#)). Having said that, the MSE and Mean Absolute Error (MAE) of the previous models are always greater than those of the proposed FE-Transformer, which results in their $R^2 < 60\%$ for all F values.

Another performance metric in Table 5.3 is the scatter index, which is calculated by dividing the root-MSE by the average TS values in each day. The low percentage of the scatter index of the FE-Transformer indicates a small relative error with respect to the

Table 5.3: Comparing the Performance of Our FE-Transformer Model with Two Recently Published Works in Literature

DNN Model	Forecasting horizon (F)		
	1	3	7
MSE			
Conv3D by Mathieu et al. [365]	58.2 μ	67.2 μ	64.7 μ
PhyDNet by Guen and Thome [367]	28.6 μ	27.8 μ	45.8 μ
The Proposed FE-Transformer	7.87 n	10.23 n	11.49 n
MAE			
Conv3D by Mathieu et al. [365]	120 μ	163 μ	177 μ
PhyDNet by Guen and Thome [367]	142 μ	174 μ	205 μ
The Proposed FE-Transformer	5.25 μ	8.54 μ	15.8 μ
Scatter Index			
Conv3D by Mathieu et al. [365]	8.6%	23.6%	45.6%
PhyDNet by Guen and Thome [367]	9.3%	22.7%	45.8%
The Proposed FE-Transformer	0.6%	1.3%	4.7%
R^2			
Conv3D by Mathieu et al. [365]	57%	29%	19%
PhyDNet by Guen and Thome [367]	51%	27%	19%
The Proposed FE-Transformer	99%	95%	92%
F-test (p-value)			
Conv3D by Mathieu et al. [365]	0.45	0.50	0.47
PhyDNet by Guen and Thome [367]	0.45	0.48	0.47
The Proposed FE-Transformer	0.01	0.01	0.13

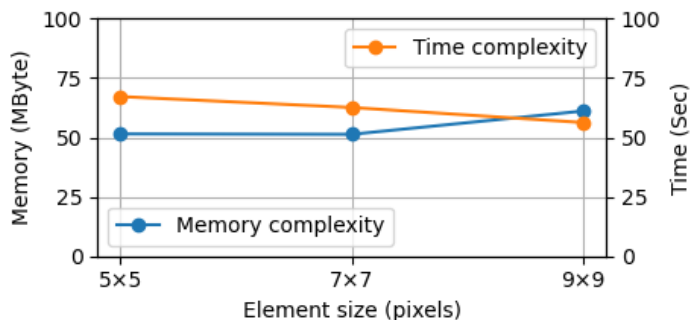


Figure 5.6: Time and memory complexities with respect to the element size for one-day forecasting horizon with $N = 3$.

average TS.

We also conduct an analysis of variance (F-test) in Table 5.3 to find out whether the TS predictions and their relevant true values have the same variance. Larger p-values reject the null hypothesis (H_0), indicating that the variances are not equal. Based on this test, our model perfectly captures the TS variances with probabilities greater than 99%, 99%, and 85% for $F = 1$, $F = 3$, and $F = 7$, respectively. These p-values are acceptable in the wide GBR with high-dynamic sediment distributions, especially when the other comparing models are barely close to the 55% probability at their best.

As can be seen from Table 5.3, neither Conv3D nor PhyDNet can accurately predict the next TS frame in the wide GBR. This is mainly because both of these next-frame predictors treat each frame as a whole, compared to FE-Transformer that borrows the finite element concept and uses the modified stiffness matrices to produce accurate predictions in meshed GBR frames. Additionally, according to [360], ViT-based models perform better than their convolutional counterparts on larger training datasets. As discussed in Section 5.3.1, we have a huge number of training image patches (i.e., elements), which justify the employment of the ViT in the proposed FE-Transformer model.

5.3.3 Computational Complexity

The proposed FE-Transformer model was implemented by TensorFlow 2.4.1 in Python, and the DNN layers were employed from the relevant Keras APIs. The model was trained on a machine with Intel[®] Core i7-7700HQ CPU, NVIDIA[®] GeForce 1050 GPU, and 16 GB RAM.

The computational complexity of the proposed FE-Transformer model is shown in Fig. 5.6, where both the memory consumption and running time are plotted versus the element size. The memory consumption refers to the allocated RAM for the model, which excludes the dataset memory and GPU memory. The running time on the other hand, is

the time required for inference. The slight reduction in running time versus the element size in this figure is due to the increased number of unpredictable marginal pixels. Overall, the plot reveals an $O(1)$ constant demand for both time and memory, which suggests our model is efficient.

5.3.4 Ablation Study

The ablation study in machine learning is the mechanism of removing a component of a model to study its importance in the overall system. This study is conducted in Table 5.4, where each row corresponds to the elimination of a given component from the proposed FE-Transformer in Fig. 5.3. The first row includes all the construction blocks, the second row excludes the stiffness matrix channel from input, the third row excludes linear embedding as well as the Transformer encoder, and the fourth row excludes two dense layers of the MLP. In each case, the MSE is averaged over the test dataset in year 2018. As expected, the MSE increases in subsequent rows, by removing functional components of the structural body.

Comparing all the MSE values in this table reveals a high contribution of the stiffness matrix to the TS next-frame prediction. Removing this FEA-originated parameter reduces the model accuracy by 33%. The stiffness matrix is a core component of the proposed FE-Transformer model. Not only does it approximate the underlying PDE solutions, but it also relates TS variations to its historical values. Therefore, removing the stiffness matrix has the most adverse effect on the model performance.

The transformer layer is another key component playing a significant role in our next-frame predicting model. In short, the transformer makes the model capable of addressing wide time-gaps in TS long-term dependencies. Eliminating this layer drops the model accuracy by 21%. Finally, the MLP module consists of two hidden layers and 160 neurons. This module is responsible for producing the TS prediction at the Transformer output for a given pixel. Similar to conventional DNN architectures, the MLP after the flattened values in Fig. 5.3 helps reduce the dimension of the transformer output to one final neuron for the target pixel. Without the MLP, the dimension reduction does not happen and an 18% drop in prediction accuracy would occur.

5.3.5 Limitations and Future Directions

Similar to other DNN-based models, the main limitation of the proposed FE-Transformer is the availability of training data. After dividing a study area into elements and pixels, training data must be collected for each pixel in regular time-steps. In other words, this model is only applicable to spatio-temporal problems with regularly measured or

Table 5.4: Evaluating the Elimination of Selected Blocks on Accuracy of the Model

Ablation study	MSE (n)	MSE increment
Including all the blocks	7.87	0%
Excluding stiffness matrix	10.47	33%
Excluding the transformer layer	9.52	21%
Excluding the MLP layer	9.29	18%

interpolated data in both the time and spatial domains. Additionally, computational resources are another limiting factor of the proposed model. To overcome this limitation, training data must be downsampled into widely separated pixels. Another workaround can be using a more powerful computing unit or even a distributed training scheme.

Accordingly, future research can both address the above limitations and enhance the capabilities of the proposed model. One such enhancement would involve combining the FE-Transformer with remote sensing data. While environmental data is hard to gather in remote areas, remotely sensed data in multiple frequency bands are readily available. These data can be merged with scarce measurements using the PINN method to train the model with more data, potentially improving its prediction accuracy.

5.4 Conclusion

Distribution of TS in the oceans is highly spatially and temporally dynamic, and is usually modeled using a complex set of PDEs. In this chapter, we proposed a novel FEA-inspired DNN to predict the sediment distribution in the wide GBR. In our proposed method, the GBR is meshed into small elements. For each element, a stiffness matrix is calculated using the historical training dataset. This stiffness matrix along with its relevant TS element are fed to the proposed FE-Transformer to predict the next TS frame in a forecasting horizon between 1 to 7 days. For training the proposed FE-Transformer model, we employed the PINN technique to merge measured TS data from AIMS MMP with PDE simulated data from *eReefs*. Through integrating the FEA concept with the state-of-the-art Transformer network, the trained FE-Transformer model was able to efficiently learn long-term TS dependencies. This resulted in highly accurate TS frame predictions with a small MSE. Besides, the model’s output frame had great R^2 , which is an indicator of the model’s unblurred prediction. Our proposed model will be beneficial in reliably predicting TS concentration and distribution across the GBR. This will aide in better managing and planning water quality targets set for the GBR.

Chapter 6

IoUT Distributed Training and Edge Computing

Data communications in underwater wireless sensor networks are dominantly conducted by acoustic waves, which are suffering from a narrow frequency bandwidth. To address this problem, an optimized deep learning design for low-energy and real-time image processing at the underwater edge is proposed. The proposed platform can remove the big data processing barrier in the Internet of Underwater Things (IoUT). It can be employed by a variety of underwater exploration, monitoring, navigation, tracking, disaster prevention, and scientific data collection projects. In addition, distributed learning to train deep learning models has been investigated. It is shown that distributed learning in the cloud can help more efficiently process big data and train more accurate deep learning models. This chapter is published in *IEEE Access* as

- [28] M. Jahanbakht, W. Xiang, N. J. Waltham, and M. R. Azghadi, “Distributed deep learning in the cloud and energy-efficient real-time image processing at the edge for fish segmentation in underwater videos,” *IEEE Access*, Aug. 2022.

6.1 Introduction

As Deep Neural Network (DNN) models grow to have billions of learning parameters, while the training data volumes expand to petabytes, model training on local computers becomes highly inefficient, if not impossible, demanding the use of Distributed Computer Systems (DCS). Besides, a well-designed DNN training on DCSs may achieve higher accuracy by disentangling the weight optimization into separated nodes [363]. On this basis, the next generation of cloud-based distributed computer networks may enable DNN processing on edge devices for both improved model training and efficient model inferencing [364], ultimately leading to enhanced decision making.

To that end, in this chapter we first explore cloud-based distributed DNN training

and analyze its benefits and shortcomings in training a large-scale DNN performing fish segmentation in real-world underwater videos. Next, we deploy our trained DNN on an embedded edge processor to show the benefits it provides for the Internet of Underwater Things (IoUT) and its wireless communication technology.

Wireless communication plays an important role in all branches of the Internet of Things. However, this type of communication faces many challenges, when it comes to the IoUT [24]. These challenges include high attenuation, multipath fading, frequency dispersion, and signal distortion of electromagnetic waves, which cannot penetrate and propagate deep in underwater environments.

These harsh underwater conditions led scientists toward the Underwater Acoustic Sensor Network (UASN), which by far is the dominant wireless technology in IoUT [24]. UASN is defined as sonic interconnection of marine objects that enables maritime exploration and monitoring activities. However, UASN has some limiting technical characteristics such as low transmission bandwidth, high signal attenuation, and high propagation delays [365].

To overcome these UASN drawbacks, the relatively new concept of edge processing seems to provide a promising solution [24]. In edge computing, endpoint devices perform parts of the required computations on their own data. The results of these computational processes have smaller volumes, compared to the raw input data. By sending the results instead of the initial unprocessed data, underwater network traffic will significantly reduce. Meanwhile, the lower data transmission rate will consequently result in lower communication latency and efficient energy consumption [366].

To get the most out of the edge computing, efficient computational processes must be employed. This is specially important when dealing with the marine high-resolution image and real-time video data streams. To address this problem, modern deep learning processes can be used. State-of-the-art use cases of DNNs in underwater image/video applications are ranging from image enhancement [367] to object detection and classification [368,369], and further to the vision-based undersea navigation and tracking [370].

However, DNNs usually consist of large architectures requiring large computational resources and power consumption, which are not readily available at the underwater edge. To overcome this challenge, here we propose two strategies. The first is optimizing and compressing the DNN models deployed at the edge. We show that this can result in significant power saving and reduced processing time when compared to remote processing on land. The second strategy we suggest is to use practical approaches to reduce power consumption or harvest environmental energy to power the edge device.

To summarize, in this chapter we use cloud-based distributed DNN training and edge DNN inferencing for a real-world underwater image processing task. This area has not been widely explored [371], and deserves further research. To fill this gap,

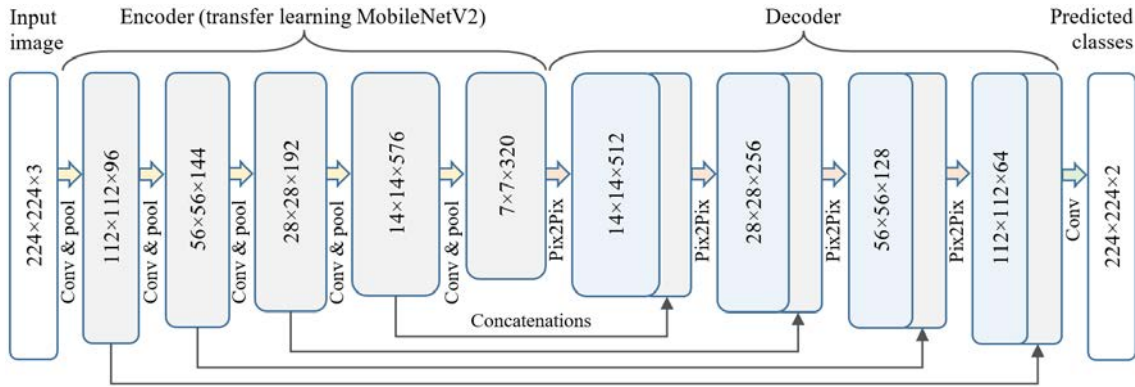


Figure 6.1: Modified architecture of the U-NET convolutional neural network for underwater image segmentation. The weights of the encoder part are transfer-learned from a pretrained MobileNetV2 on the ImageNet. The decoder part employs Pix2Pix for upsampling.

- We propose, to the best of our knowledge, the first DNN-based edge processor for real-time fish segmentation in underwater videos. This is implemented on an embedded Graphics Processing Unit (GPU) and is benchmarked against traditional HTTP inference on a GPU-powered computer on land.
- The fish segmentation model is trained on distributed cloud infrastructure to make it more suitable for big marine data training, while also slightly improving its learning accuracy. The distributed training is benchmarked against an on-premises standalone computer, in terms of speed and accuracy.
- To improve the delay and energy requirement of our system in the targeted underwater environment, we compress our DNN model to two quantized weight resolutions. We benchmark these against the full model and show the compressed networks can result in significant improvements. In addition, we propose an efficient energy management plan, consisting of renewable energy resources and motion detection technologies.

We discuss how the proposed underwater edge computing platform can remove the big data processing barrier in the internet of underwater things.

The rest of this chapter is organized as follows. In Section 6.2, a modified U-Net model will be designed for fish segmentation in real-life underwater images. This model will be accurately trained on a cloud-based DCS in Section 6.3. Later in Section 6.4, energy reduction techniques will be introduced, and the trained model with compressed weights will be deployed on a GPU-enabled edge device for fast and efficient inferencing. We will

also investigate how edge computing makes underwater big data processing possible. The chapter is concluded in Section 6.5.

6.2 Fish Segmentation

Image segmentation is a major topic in computer vision, which assigns a label to each pixel of the image. Pixels with the same label belong to the same semantic object (i.e., class). Image segmentation is applied in a number of domains including object localization, video surveillance, scene understanding, augmented reality, and many other image processing applications. To have an accurate image segmentation model, a wide variety of DNNs have been reported in the literature, and they are comprehensively surveyed by Minaee *et al.* [372].

However, the underwater environment is not image-friendly. The suspended particles in seawater absorb, backscatter, and forward scatter light rays towards the camera, which creates hazy images with low contrast and faded colors [367]. These image degradation processes make underwater image segmentation a challenging task.

6.2.1 Modified U-Net Architecture

To address the fish segmentation problem in underwater images, we employed a modified version of the U-Net model [373]. Here, U-Net is a Convolutional Neural Network (CNN)-based deep learning algorithm which has been applied in a number of applications ranging from medical imaging [374], to fish segmentation [375]. Similar to the original U-Net [374], modified U-Net in this chapter consists of two consecutive encoder and decoder paths. While the contracting encoder path captures contextual features, the symmetric and expanding decoder path enables segment localization. However, our modified U-Net makes use of a better upsampling block [373].

Both the encoder and decoder parts of the modified U-Net architecture are illustrated in Fig. 6.1. The encoder part follows the typical CNN architecture with five consecutive convolution layers, each followed by a ReLU activation function and a max-pooling operator. The $7 \times 7 \times 320$ output of the encoder part feeds to the following decoder part, which consists of four consecutive upsampling and concatenating blocks. Unlike the original U-Net, a Pix2Pix block [376] is used for upsampling in the modified U-Net. Pix2Pix was initially used in the generative adversarial networks for their capability of generating high-quality images across a variety of image translation tasks [376]. Since then, it is widely used as an upsampling block in a wide range of other applications. At the last stage of the modified U-Net model, a simple 2D convolution layer was used to map the extracted 64 features to the desirable two segmentation classes (i.e., fish or not fish).

To reduce the number of trainable parameters in our modified U-Net to improve its training speed, we used the transfer learning technique and froze the transferred weights. To elaborate, the weights and biases of a pretrained MobileNetV2 model on the ImageNet dataset were reused in the encoder part [377].

6.2.2 DeepFish Dataset

Supervised DNNs are data hungry models. They require hundreds of labeled data in each class, to succeed in their training phase. A successful training will result in accurate predictions at the subsequent inferencing phase. To train our modified U-Net, the open-access DeepFish dataset was used. DeepFish is a realistic dataset of fish images in their natural habitat [378]. The images have been accurately labeled for classification, localization, and segmentation algorithms evaluation.

DeepFish’s segmentation dataset contains 620 images with accurately labeled fish segments. These images are collected from 20 habitats in the tropical Australian marine environment with and without fish presence. To avoid overfitting, 10% of all images are separated for validation. The images with one or more fish have been augmented by 180° rotation, X and Y flipping, blue and green color degradation, and random noise addition. To have a balanced dataset, the images with no fish are not augmented. Furthermore, the DeepFish images are originally shaped in a 1920×1080 pixel rectangle. To feed these images into our modified U-Net model, they are resized into a square shape of 224×224 pixels. This resizing is done by 49 zero-padded pixels both on top and bottom of each image.

6.3 Distributed Training

Not so long ago, labeled datasets for supervised machine learning were scarce. However, the number of these expert-labeled and open-access datasets have recently been increased by the order of multiple hundreds. For example, none of the tens of labeled underwater datasets introduced in the GitHub repository [Awesome Underwater Datasets](#), existed a few years ago.

With labeled image proliferation in open-access datasets, processing them becomes more challenging and compute-demanding. More specifically, the pixel labeling algorithms (like segmentation DNNs) with petabytes of remotely gathered data, such as underwater fish videos, will require significant computation and memory resources, which are not available on a single computer. Here, Distributed Computer Systems (DCS) with several processing nodes that share the training, and consequently the memory and computing workload, can be considered [379].

Table 6.1: Hardware Specifications of the Utilized Local Computer, Distributed Computers, and the Edge Device

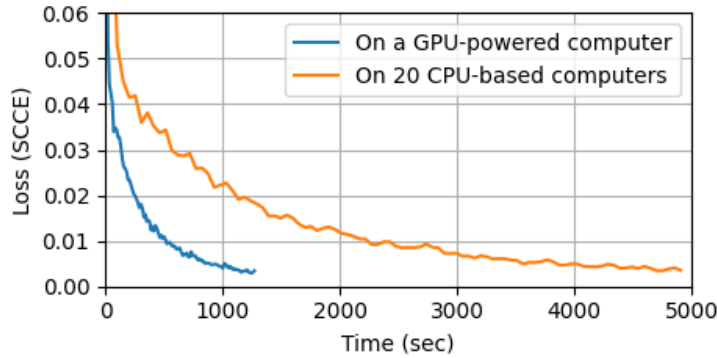
CPU	GPU	RAM	SSD	Price
Local computer: Dell Precision 3630				
Intel 8-core	Nvidia GeForce (1.6 GHz, 8 GB, and 3071 CUDA cores)	32 GB	1 TB	\$3500
Distributed computers: AWS M5.2xLarge Instances				
Intel 8-core	–	32 GB	61 GB	\$0.576 per hour
Edge processor: Nvidia Jetson Nano				
ARM 4-core	Nvidia Maxwell (1.1 GHz, 4 GB, and 2048 CUDA cores)	4 GB	64 GB	\$100

Since the volume of the DeepFish dataset used in our work was only 15 GB, and our ImageNet pretrained MobileNetV2 model size was 150 GB, the data volume was not a major concern in our chosen task. However, our DNN can still prove the DCS training concept and benefit from distributed learning by increasing its accuracy. DNNs are directed-acyclic-graphs that have unknown weights on the graph vertices. These vertices will disentangle in a distributed computing, and the weights of every vertex will be independently trained on a dedicated computer node [380]. This weight separation will reduce overfitting to the training dataset, and will consequently increase the overall DNN accuracy.

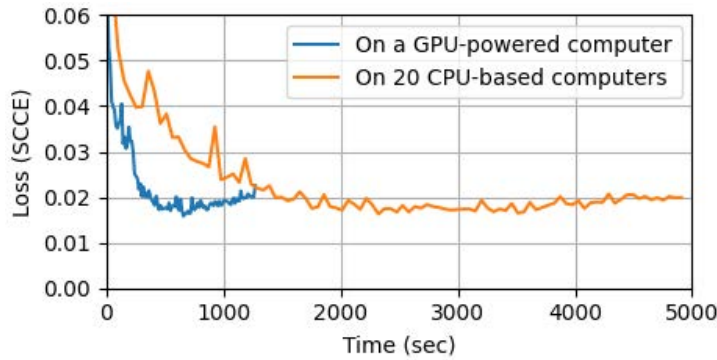
6.3.1 Cloud-based Distributed Computer Systems

A DCS is a private network of separated computers, each holding a set of software components that collaboratively work as a single system. Despite their so many benefits, DCSs need complex experimental and architectural design procedures for [24]:

- *Distributed algorithms:* Every DCS relies on a parallelization algorithm for fast and accurate operation. Selecting this algorithm is usually challenging and needs expert knowledge of the data and model.
- *Horizontal expansion:* There is always a trade-off between the running time of a project and the number of DCS nodes.



(a)



(b)

Figure 6.2: The SCCE loss metric convergence for both (a) the augmented train and (b) the augmented validation datasets, versus the training time. The model is initially transfer learned on ImageNet and is shown while fine tuning with the DeepFish dataset.

- *Setup and maintenance costs:* DCS establishment requires an initial investment of both time and money. This initial cost will continue to grow by the later demands for maintenance, backup, improvements, etc.

To address these difficulties, commercial cloud services like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform seem to offer a promising solution. For instance, AWS SageMaker is a cloud-based machine learning platform for running a customer's DNN training, as well as inferencing jobs on a dedicated DCS. The number of distributed AWS training computers can horizontally expand from one node to tens of nodes on-demand. This is subject to the AWS service quotas. The end-users will only pay for the number of running hours of their application.

To perform this study and to evaluate the performance of cloud-based DCS for our proposed application, 20 identical training computers (i.e., instances) were rented from

the AWS SageMaker. Hardware specifications of each instance are compared in Table 6.1 with a GPU-powered on-premises computer. The 61 GB capacity of the cloud-based Solid State Drives (SSD) were enough for the requirements of this research.

It is worth noting that AWS SageMaker offers **P3 instances** with high-end GPUs in the cloud. These instances are capable of distributed training of DNN models across hundreds of GPUs. However, these instances are significantly more expensive than ordinary CPU-based computers. For example, a P3.2xLarge SageMaker instance with similar specifications to the employed M5.2xLarge in this study, but with one Tesla GPU, will cost \$3.825 per hour. This is nearly 7 times the cost of our current setup. Therefore, to have an affordable DCS training system, we chose to use CPU-based computers.

6.3.2 Distributed Training Results

The modified U-Net model in Fig. 6.1 accepts a 224×224 RGB image at input, and returns two 224×224 integer values for the two possible pixel classes. In other words, the DNN assigns two integer values to every single pixel of the image. These integer values indicate whether each pixel belongs to the body of a fish or not. During inference, an $\operatorname{argmax}(\cdot)$ operator must be employed to produce a single 224×224 channel with 1 and 0 values for the fish and not-fish classes. However, this is not the case in the training phase, where the exact integer outputs are passed to the Adaptive Moment (Adam) optimizer of the DCS processor.

The Adam optimizer in DCS uses the Sparse Categorical Cross Entropy (SCCE) as its major loss metric for accurate underwater fish segmentation. In addition to SCCE, the Sparse Categorical Focal (SCF) loss values with $\gamma = 2$ is also measured. SCF is particularly useful in our study, where the foreground fish pixels in each image is densely located against the huge number of background pixels [381]. To elaborate, the labeled segmentation masks are mostly zero-valued for the no-fish class, with occasional occurrences of dense one-values for the fish class.

The convergence of the SCCE loss metric for both the distributed computers and the local computer are compared in Fig. 6.2. Here, the DCS spends a lot of time on HTTP data transaction between its nodes. This makes the overall DCS convergence slower than a GPU-powered local computer.

However, the trained model on DCS achieves a lower loss compared to a single GPU training. The SCCE loss metric, SCF loss, and the Sparse Categorical Crossentropy Accuracy (SCCA) are benchmarked in Table 6.2 for both the distributed- and the locally-trained DNNs. The values in this table are separately calculated for the real training and validation datasets before augmentation. This is in contrast to the plots in Fig. 6.2,

Table 6.2: SCCE and SCF Loss Metrics, as well as SCCA Accuracy Metric of the Modified U-NET Model, Which has been Trained on a Local Computer and a Distributed Computer System

Dataset name	SCCE	SCF	SCCA
Trained locally on a GPU-powered computer			
Train Dataset	0.077	5.48	98.7%
Validation Dataset	0.065	4.63	98.8%
Trained on 20 distributed CPU-based computers			\$19.1
Train Dataset	0.067	4.46	98.7%
Validation Dataset	0.053	3.48	98.8%

which are for augmented train and validation datasets to artificially increase the amount of training data, without actually collecting new data.

As explained earlier, it was expected that due to the DNN weight separation and independent training on dedicated computer nodes [380], the model is better trained on the DCS, achieving a lower loss. This is confirmed in the results shown in Table 6.2. While the accuracy is the same, the loss metric shows better performance for the distributed training scheme. It is worth mentioning that the total renting price for 20 AWS SageMaker training instances was AUD 19.1. To summarize,

- Table 6.2 shows that the DCS training loss (0.053) is 18% better than a standalone computer (0.065).
- Due to the HTTP transactions in DCS, a GPU-powered local computer is faster than 20 distributed computers without GPU. This was illustrated in Fig. 6.2.

6.4 Underwater Edge Computing

Edge computing is an information technology paradigm that brings computation closer to the data collection hardware. In the case of IoUT, edge computing is very beneficial, without which the processing should be performed on local computers or centralized clouds on land. Transferring the data to these remote land processors would require wideband data transactions, which is not readily available or sometime feasible in IoUT. In contrast, by processing raw data on edge, only the low-volume results should be communicated.

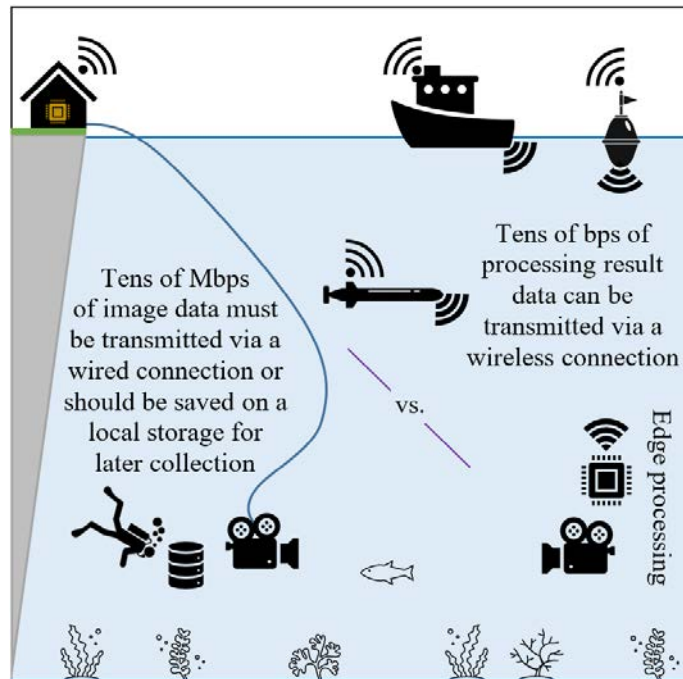


Figure 6.3: Conceptual comparison between wired underwater communication and wireless communication made possible by performing data processing at the data collection edge and only communicating the processing results.

Communicating only the results consumes narrower bandwidth (data-rate) and requires shorter transaction latency, making it suitable for IoUT [24].

Additionally, engaging the edge devices in data processing, can shift the load from a single centralized processing point to numerous distributed nodes. In this case, the system would not have a single point of failure. Besides, advanced applications such as prompt decision making will be feasible [382]. These advantages of the edge computing are better illustrated in Fig. 6.3, where the traditional multi-Mbps wired network is replaced by a multi-bps UASN using the edge processing technology. However, edge computing in IoUT must tackle the challenge of limited underwater energy resources, as its main drawback.

6.4.1 Energy Management at the Edge

Sustainable power cannot be readily delivered to the underwater edge devices. Therefore, wired energy transferring might be considered, which is a robust but limiting solution. Using wired energy will limit IoUT sensor deployment to the shoreline vicinity, and will make the system vulnerable to cable damages by gnawing animals and corrosive environments.

Due to these drawbacks, the use of wired energy in remote IoUT applications becomes infeasible. To address this challenge, here we propose two parallel techniques to effectively

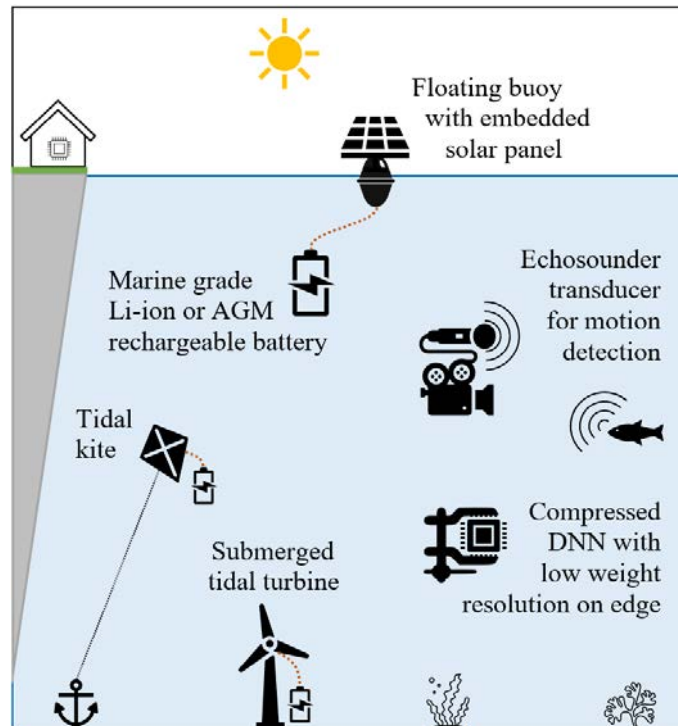


Figure 6.4: Underwater energy management plan with solar and tidal wave energy harvesting, along with the motion detection technology and DNN compression.

manage the energy at the edge, as shown in Fig. 6.4 and described below.

Reducing power consumption at the edge

Energy management from the edge's perspective means lower computation demand and/or more power-efficient consumption. To reduce the demand, ultrasonic motion detectors can be used to only demand imaging and processing when there is movement in the environment, putting the system into low-power sleep mode at all other times. This, of course, adds to the entire system power, the amount consumed by the submersible echosounder transducers, which depends on the quality of their coverage. A typical 115 kHz [ping transceiver](#) with 50 m directional range, 0.5% resolution, and 300 m depth rating that operates for 100 milliseconds per every second will take no more than 50 mW.

However, both sonar-based and vision-based motion detection technologies are sensitive to cavitation (bubbles), vegetation (leaves, sticks, etc.), water depth, and current speed. Consequently, using them in high-dynamic conditions require artificial passageways with careful control of the environmental parameters [383].

The fabrication and maintenance of passageways for underwater motion detection will

Table 6.3: Average Power Consumption of the Original Segmentation DNN with FP32 Parameters, Compared with FP16 and Int8 Compressed DNNs

Compression level	Power consumption [mW]		
	System	GPU	Total
Rest mode, with no ongoing processes			
Single-precision (FP32)	1275	42	1317
Half-precision (FP16)	1315	45	1360
8-bit integer (Int8)	1311	44	1355
Busy mode, with ongoing real-time video processing			
Single-precision (FP32)	5135	2373	7508
Half-precision (FP16)	2900	48	2948
8-bit integer (Int8)	2888	47	2935

add extra cost and time to the project, which is not desirable. Therefore, where possible, other power reduction techniques should be considered. A technique that we investigated in this chapter is utilizing a compressed DNN, which consumes less power, due to lighter computations performed at the edge. In this way, the trained weights of the initial DNN in Table 6.2 that have single-precision (32-bit) floating-point resolution (FP32), can be compressed. However, this compression may lead to a slight decrease in the DNN accuracy.

To experiment with compressed quantized DNNs in our performed segmentation task, we employed TensorFlow Lite ([TFLite](#)) to conduct the following model quantizations:

- *Half-precision (16-bit) floating-point (FP16)*: This post-training compression reduces the floating-point size to the IEEE standard float16.
- *8-bit integer (Int8)*: This dynamic range compression quantizes the weights from floating-point to integers with 8 bits precision.

To analyze the DNN weight quantization effect on its energy demand, the Nvidia Jetson nano embedded GPU was selected as our edge device. The average power consumption of our segmentation DNN models with FP16 and Int8 compressed network parameters are compared in Table 6.3, with the original FP32 model. The *System* in this table refers to the edge device internal components, except GPU (i.e., CPU, RAM, SSD, high-speed IO, etc.).

Table 6.4: SCCE, SCF, and SCCA Metrics of the Original Segmentation DNN with FP32 Parameters, Compared with FP16 and Int8 Compressed DNNs







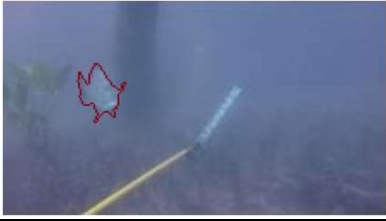





Dataset name	SCCE	SCF	SCCA
Single-precision floating-point resolution (FP32)			
Train Dataset	0.067	4.46	98.7%
Validation Dataset	0.053	3.48	98.8%
Half-precision floating-point resolution (FP16)			
Train Dataset	0.067	4.54	98.7%
Validation Dataset	0.053	3.59	98.8%
8-bit integer resolution (Int8)			
Train Dataset	0.068	4.58	98.7%
Validation Dataset	0.054	3.61	98.8%

In the rest mode with no GPU tasks, the TFLite backend of both the FP16 and Int8 demand slightly more energy (around 3%) than TensorFlow backend of FP32. This is because, TFLite does not optimize model size, compared to the FP32 case trained using TensorFlow. Therefore, the edge device requires larger storage both on its system and on its GPU. Besides, the mobile-friendly TFLite libraries do not support all TensorFlow operators. Consequently, some CUDA mounting commands might run inefficiently at the edge.

On the other hand, in the busy mode with heavy GPU inferencing tasks, the proposed FP16 and Int8 DNNs demand significantly lower power. The results in Table 6.3 show 61% total power reduction compared to uncompressed FP32 model. This improvement can eliminate the need for motion detection technologies, by enabling the edge device to continuously process underwater video frames. Alternatively, if possible, it can be combined with the motion detection methods for further decreased power consumption.

Despite the fact that weight compression reduces power demand at the edge, it can also decrease the DNN accuracy. This concern has been investigated in Table 6.4, where the loss and accuracy metrics of the compressed FP16 and Int8 models are compared with the original FP32 DNN. Based on this table, the reduction in model performance (specially for the SCF metric) is insignificant.

Despite the reduction techniques proposed above, the power consumption of the edge processor can be still significant and in the order of a few watts, requiring a sustainable

	Ground truth with zero, one, or more fish	Int8 DNN segmentation result on edge	Metrics
True positive			SCCE: 0.178 SCF: 11.936 SCCA: 96.7%
			SCCE: 0.243 SCF: 17.903 SCCA: 96.3%
			SCCE: 0.134 SCF: 10.336 SCCA: 97.8%
			SCCE: 0.059 SCF: 4.113 SCCA: 98.9%
True negative			SCCE: 0.001 SCF: 0.000 SCCA: 100%
			SCCE: 0.000 SCF: 0.000 SCCA: 100%

(To be Continued)

Figure 6.5: Example outputs of the modified U-NET segmentation model trained on 20 cloud-based distributed computers of the AWS SageMaker, and then deployed with Int8 parameter resolution on a Jetson Nano device.





	Ground truth with zero, one, or more fish	Int8 DNN segmentation result on edge	Metrics
False positive			SCCE: 0.004 SCF: 0.094 SCCA: 99.8%
			SCCE: 0.003 SCF: 0.041 SCCA: 99.8%

Figure 6.5: (*Continued.*) Example outputs of the modified U-NET segmentation model trained on 20 cloud-based distributed computers of the AWS SageMaker, and then deployed with Int8 parameter resolution on a Jetson Nano device.

method of power delivery to the edge device. Here, we propose combining the above-mentioned reduction techniques with several energy harvesting approaches described below, as shown in Fig. 6.4, to make IoUT edge processing feasible.

Energy harvesting

In undersea energy harvesting, solar panels and the marine grade Absorbent Glass Mat (AGM) or Li-ion batteries are well-developed for small-scale applications. For example, today's rechargeable Li-ion and AGM batteries can provide superior power to support the electrical demands of start-stop edge devices, while being affordable, impervious to seawater, resilient to underwater vibrations, and maintenance-free [384].

Tidal stream energy harvesting systems are another source of power in undersea environments. While multi-megawatt tidal stream turbines can feed power grids with clean energy, small-scale submersible generators [385] and tidal kites [386] can provide efficient power for UASNs in our proposed application. In addition, tidal power is almost perfectly predictable over long timescales, which is an appealing feature for power management systems. The output energy during neap tides is significantly less than that during spring tides, hence, in a realistic system with steady output power, rechargeable batteries are inevitable for covering the lower energy production periods [385].

When the above mentioned power harvesting methods are combined with the proposed power reduction at the edge, inferencing can happen efficiently at the edge.

6.4.2 Edge Inferencing Results

To further analyze the benefits of edge processing compared with transferring the underwater video frames to land for processing, we performed some further inference analyses. To perform these analyses, we needed to choose a GPU-enabled edge device to infer video frames on our proposed modified U-NET DNN architecture. Although, a variety of edge (embedded) GPUs exist in the market, the Nvidia Jetson Nano from the Jetson processor family, which is the world's leading platform for machine learning at the edge [371] was selected. Nano's specifications are shown in Table 6.1. The Jetson Nano module is an efficient minicomputer that runs on Linux for Tegra (L4T). L4T is a free distribution of the Ubuntu Linux by Nvidia for its Tegra processor series.

To utilize Jetson Nano for our application, we installed Python, TensorFlow, and many other software packages on L4T to enable fast and accurate DNN inferencing. Additionally, we developed a Flask-based REST-API in L4T to answer remote HTTP queries. Flask is a light-weight microframework that suits the low-power requirements of an edge computing ecosystem.

As described earlier in Sections 6.3 and 6.4.1, our proposed modified U-NET was initially trained on 20 cloud-based distributed computers. Next, the trained model was compressed to Int8 resolution for power reduction. To test the functionality of our developed framework, we deployed it on a Jetson Nano to perform fish segmentation on a number of video frames from the DeepFish dataset. Fig. 6.5 shows a number of sample segmentation outputs. The demonstrated samples in Fig. 6.5 include some desirable true-positive and true-negative predictions, along with two undesirable false-positive detection. We did not encounter any false-negative report on the images in the DeepFish dataset.

6.4.3 Edge Inferencing Speed

The inferencing speed on the Jetson Nano edge device was compared to an on-premises land server in Fig. 6.6. The first frame inferencing in both the land-side server and the FP32 edge device requires much longer time than the TFLite-based FP16 and Int8. This low initialization speed is due to the model storage method in TensorFlow, which requires more time to map a saved DNN model into a directed-acyclic-graphs in GPU.

In the case of on-premises GPU-powered server in Fig. 6.6, multi-MB video frames must be submitted via a wired LAN connection first to then be processed on the GPU. The transmission significantly increases the inference time as shown in the figure. In contrast, the edge device processes video frames in-place, producing few bytes of results that usually do not require instant wireless transmission. These results may include (x, y) coordinate of the detected fish, its size, and the frame timestamp that are submitted to the land

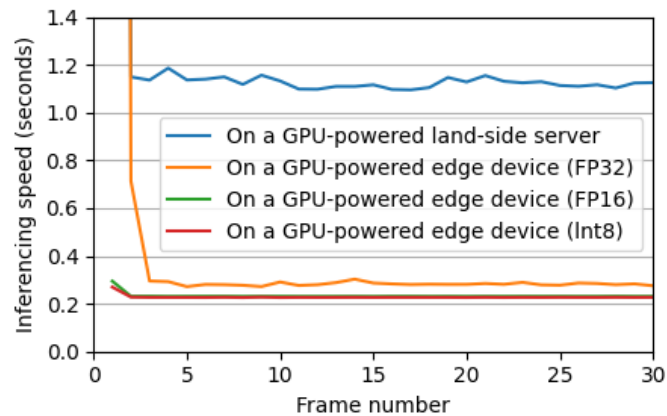


Figure 6.6: Image or video frame segmentation speed of an edge device with multiple compression levels is compared to a local computer that has been inquired by HTTP protocol.

station once in a while.

The GPU-powered landside server with better computing resources (see the first row of Table 6.1) is 4 times slower than the edge embedded Nano GPU. Meanwhile, the model compression to FP16 and Int8 reduces inference latency at the edge by 18%, compared to the FP32 model. This lower latency means that the edge processor can segment 3-4 times more frames (images). It is worth noting that, reducing latency also reduce the overall energy consumption of the system. Of course, if faster processing rates are required, more powerful but also expensive devices like Jetson Xavier can be employed.

Overall, the presented results show that the proposed methodologies for underwater edge inferencing can significantly reduce the latency and power requirements of any underwater DNN processing tasks, compared to transferring the large data to land for processing. This can significantly advance IoUT ecosystems in various applications ranging from marine ecological studies to disaster prevention.

6.5 Conclusion

In this chapter, we first proposed a modified U-NET architecture for fish segmentation in underwater videos. We then demonstrated the use of DCS for training our fish segmentation task and discussed the benefits DCS provides for training DNNs using big marine data. In the second part of our work, we utilized our DCS-trained DNN to show the benefits of edge processing in underwater environments. It was shown that edge processing can result in more than 4 times speedup, compared to the conventional method of remote land processing. Furthermore, we proposed an energy management plan at the

edge, which utilized well-known techniques such as motion-detection, energy harvesting, and compressed DNNs to make underwater edge computing feasible. We showed that, by compressing a DNN model for fish segmentation, 61% of power saving can be achieved compared to a FP32 DNN.

The techniques proposed in this chapter can be applied in other domains, most notably, in other underwater applications, where sending large data to land for processing is not feasible. These simple but practical techniques can significantly advance current and future underwater processing applications, leading to more informed decisions in remote underwater environments, for applications such as marine research, navigation, and tracking.

Chapter 7

Conclusion and Future Studies

7.1 Conclusion

A comprehensive survey of the IoUT and BMD, along with an analysis of multiple machine learning applications for marine data processing, were conducted in this thesis. We commenced by surveying the state-of-the-art in underwater communications. The families of advanced underwater acoustic, electromagnetic, and optical technologies were introduced and innovative solutions were proposed for increasing the overall link reliability. Furthermore, SDN and edge computing techniques were reviewed as a promising technique of improving underwater communications.

By connecting more objects to the IoUT, BMD generation is inevitable. This brings data storage, transportation, preparation, and analysis challenges. To address these challenges, machine/deep learning and its applications to marine data processing was reviewed, along with the critical appraisal of recent published works. ML solutions cover the areas of sensor, image, and video data leveraging, marine geographic data processing, localization and tracking, and distributed data processing.

Among the numerous BMD processing challenges with ML solutions, timeseries forecasting, next-frame prediction, and marine image processing on edge were selected as our contribution areas. For the timeseries forecasting, accurate prediction of SST was selected for its environmental importance. The SST significantly affects the global mean air temperature, its anomalies can directly affect the fish habitat, and it is a crucial factor for the health of the corals and reefs. We proposed a highly accurate DNN structure that leverages the correlation between SST datasets and air temperature at nearby isolatitude weather stations. We devised an ensemble of two stacked LSTM-MLP networks and trained them with the correlated SST and air temperature datasets. We demonstrated that our model significantly outperforms the state-of-the-art SST prediction algorithms.

For the next-frame prediction, two important problems of nitrogen and sediment forecasting in the wide GBR were addressed. Nutrients and sediments are considered to be the

primary land-based pollutants that significantly reduce ocean water quality. High rainfall, flash floods, numerous short river basins, and the close proximity of the reef to the Wet Tropics of Queensland mean nutrients and sediments are flushed to the reef lagoon quickly.

In this regard, two predictive models have been designed to forecast and manage the high risk areas in the coral reef ecosystems. Implementing those predictive models in the vast areas of the GBR was challenging, and they were addressed by incorporating FEA concepts into the DNN technology. To elaborate, inspired by the well-known FEA, we proposed

- The FE-DNN model for next-frame prediction of nitrogen in the GBR, and
- The FE-Transformer for next-frame prediction of sediment in the GBR.

It is worth noting that the convolutional neural networks in FE-DNN and the Transformer network in FE-Transformer are serving the same next-frame forecasting concept in two different applications. So, both models are accurate predictors, and they can be used interchangeably in future studies.

One challenge in training both these DNN-based models was the scarcity of observational data in the GBR. To address this problem, we employed the PINN technique to merge the large amounts of simulated data with the sparse measurement data. The performed analyses revealed that our next-frame models achieve very high accuracy with a low prediction MSE, while yielding high-resolution prediction frames with very high R^2 values.

Marine image processing was the last BMD processing challenge that we researched in this thesis. Data communications in underwater wireless sensor networks are dominantly conducted by acoustic waves, which are suffering from a narrow frequency bandwidth. More specifically, current undersea monitoring projects with big data outputs are spending their resources on wired networks, technically because wireless underwater networks are either impossible or expensive. To address this problem, a modern platform was proposed for real-time underwater video processing on edge. This platform consisted of a deep neural network, which was accurately trained on a distributed cloud system. The trained model was then compressed and deployed on an edge device for real-time fish detection in videos.

The proposed edge computing platform reduces the required data transmission volume, which makes it possible to be transmitted in the limited underwater communication networks. To elaborate, an underwater acoustic network can easily transmit a few bytes of data, summarising the results acquired on the edge, instead of sending all the image data to surface for processing. Besides, the utilized weight compression technique dramatically reduces the DNN power consumption, making it suitable for underwater applications.

7.2 Future Studies

As stated in the previous section, recent advances in IoT and IoUT technologies has led to the proliferation of Internet-connected objects both in over- and underwater applications. This technological evolution in the area of marine sensors, cameras, hydrophones, etc. raises new issues for undersea data acquisition, data communication, and BMD processing.

Having studied the state-of-the-art research, the followings are some of the most important remaining research questions and future study directions. These few items can provide an insight into the prospective applications, trends, and challenges.

- The underwater sensors and multi-sensor buoys are usually very costly compared to their terrestrial counterparts. Additionally, sensor deployment in the vast oceanic areas and their regular maintenance is not practical. The lack of data arising from this issue, was resolved in our proposed FE-DNN and FE-Transformer models by benefiting from *eReefs* simulation data merged with the scarce measured data. A future research direction is to utilize satellite data to train new forecasting DNN models for highly-accurate predictions. These remote sensing data with ocean color algorithms can be used to measure water quality variables like total nitrogen, total suspended sediments, chlorophyll-a, Secchi depth, benthic photosynthetically active radiation, etc. It is worth noting that the existing models for remote water quality sensing are based on the traditional neural networks [387]. This makes them suffer from unstable estimations in rivers, as well as in close proximities to the shorelines. Other potential research questions to be investigated are to retrain our proposed FE-DNN models with satellite data only, or with a combination of satellite, eReefs, and scarce in-situ environmental measurement data.
- The processing of BMD requires powerful hardware and software tools that can automatically extract knowledge from large databases. This thesis proposed some novel models toward developing software tools for accurate 1D and 2D temperature, nitrogen, and sediment prediction. However, many important environmental parameters in the sense of water quality are left untouched. For example,
 - The proposed FE-DNN and FE-Transformer models can be employed for other chemical, physical, and biological water parameter forecasting. This includes but is not limited to dissolved oxygen, pH, conductivity, biomass, etc. [325].
 - Every environmental parameter can be considered as an endogenous statistical variable that depend on many other exogenous factors. As shown in Fig. 7.1 for the sea surface temperature, the proposed ensemble model of stacked DNNs can

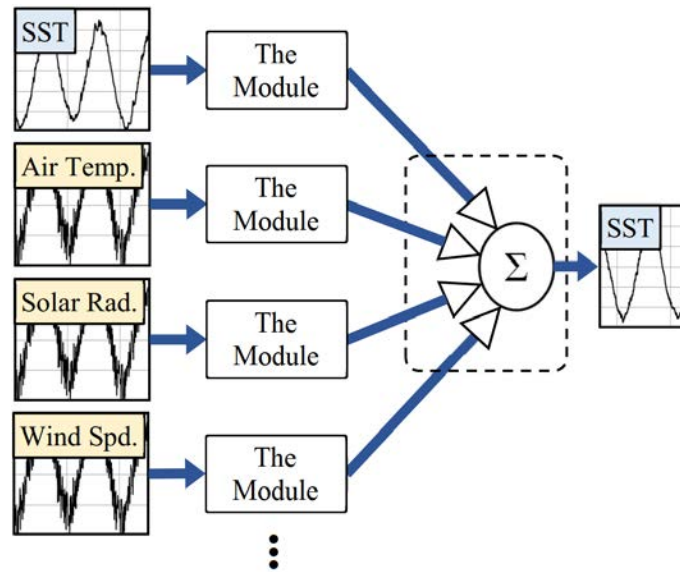


Figure 7.1: Employing as many exogenous factors as desired to predict targeted endogenous variable with the proposed ensemble model of stacked DNNs.

be used to add as many sensory data as required for highly-accurate forecasting of any targeted environmental variable.

- Most of the maritime water quality factors like dissolved oxygen, turbidity, biomass, etc. depend on water depth. For example, the dissolved oxygen levels in sea surface varies due to oxygen exchange with the atmosphere. At depth, the dissolved oxygen levels are more stable. In this regard, our 1D and 2D prediction models are designed for water quality studies in sea surface. Further research can be done to extend these studies into 3D models at ocean depth. Having these forecasting models in future can help improve ecosystem recovery by informed decision making. Furthermore, they can be adopted by existing hindcasting simulators like *eReefs*, to provide accurate forecasting predictions.

References

- [1] G. Xu, Y. Shi, X. Sun, and W. Shen, “Internet of things in marine environment monitoring: a review,” *Sensors*, vol. 19, no. 7, pp. 1711–1732, Apr. 2019.
- [2] J. F. Fuentes-Pérez, C. Meurer, J. A. Tuhtan, and M. Kruusmaa, “Differential pressure sensors for underwater speedometry in variable velocity and acceleration conditions,” *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 418–426, Apr. 2018.
- [3] R. Al-Zaidi, J. C. Woods, M. Al-Khalidi, and H. Hu, “Building novel VHF-based wireless sensor networks for the internet of marine things,” *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2131–2144, Mar. 2018.
- [4] C. A. Pérez, F. S. Valles, R. T. Sánchez, M. J. Buendía, F. López-Castejón, and J. G. Cervera, “Design and deployment of a wireless sensor network for the Mar Menor coastal observation system,” *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 966–976, Oct. 2017.
- [5] R. Khalifeh, M. S. Yasri, B. Lescop, F. Gallée, E. Diler, D. Thierry, and S. Rioual, “Development of wireless and passive corrosion sensors for material degradation monitoring in coastal zones and immersed environment,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 4, pp. 776–782, Oct. 2016.
- [6] M. Jouhari, K. Ibrahim, H. Tembine, and J. Ben-Othman, “Underwater wireless sensor networks: a survey on enabling technologies, localization protocols, and internet of underwater things,” *IEEE Access*, vol. 7, pp. 96 879–96 899, Jul. 2019.
- [7] Z. Zeng, S. Fu, H. Zhang, Y. Dong, and J. Cheng, “A survey of underwater optical wireless communications,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 204–238, Jan. 2017.
- [8] H. Kaushal and G. Kaddoum, “Underwater optical wireless communication,” *IEEE Access*, vol. 4, pp. 1518–1547, Apr. 2016.
- [9] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

- [10] Y. LeCun, Y. Bengio *et al.*, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 255–258, Apr. 1995.
- [11] K. Kim and H. Myung, “Autoencoder-combined generative adversarial networks for synthetic image data generation and detection of jellyfish swarm,” *IEEE Access*, vol. 6, pp. 54 207–54 214, Sep. 2018.
- [12] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings Empirical Methods in Natural Language Processing*. Seattle, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642.
- [13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [14] A. Canziani, A. Paszke, and E. Culurciello, “An analysis of deep neural network models for practical applications,” *Computing Research Repository*, Apr. 2017.
- [15] S. Zagoruyko, “ImageNet-validation.torch,” www.github.com, Jun. 2016.
- [16] J. Xie, J. Zhang, J. Yu, and L. Xu, “An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 740–744, Aug. 2020.
- [17] K. Zhang, X. Geng, and X. Yan, “Prediction of 3-D ocean temperature by multilayer convolutional LSTM,” *IEEE Geoscience and Remote Sensing Letters*, Jan. 2020.
- [18] S. Rani, S. H. Ahmed, J. Malhotra, and R. Talwar, “Energy efficient chain based routing protocol for underwater wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 92, pp. 42–50, Aug. 2017.
- [19] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, “Collaborate edge and cloud computing with distributed deep learning for smart city internet of things,” *IEEE Internet of Things Journal*, May 2020.
- [20] M. Kumar and S. Chand, “A secure and efficient cloud-centric internet of medical things-enabled smart healthcare system with public verifiability,” *IEEE Internet of Things Journal*, Jul. 2020.
- [21] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, “IoT, big data and artificial intelligence in agriculture and food industry,” *IEEE Internet of Things Journal*, May 2020.

-
- [22] C. Lammie, A. Olsen, T. Carrick, and M. R. Azghadi, “Low-power and high-speed deep FPGA inference engines for weed classification at the edge,” *IEEE Access*, vol. 7, pp. 51 171–51 184, 2019.
- [23] N. Morozs, P. D. Mitchell, and R. Diamant, “Scalable adaptive networking for the internet of underwater things,” *IEEE Internet of Things Journal*, Apr. 2020.
- [24] M. Jahanbakht, W. Xiang, L. Hanzo, and M. Rahimi Azghadi, “Internet of Underwater Things and big marine data analytics – a comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 904–956, Jan. 2021.
- [25] M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sea surface temperature forecasting with ensemble of stacked deep neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1 502 605–1 502 609, Aug. 2021.
- [26] M. Jahanbakht, W. Xiang, B. Robson, and M. R. Azghadi, “Nitrogen prediction in the Great Barrier Reef using finite element analysis with deep neural networks,” *Environmental Modelling & Software*, vol. 150, pp. 105 311–105 327, Apr. 2022.
- [27] M. Jahanbakht, W. Xiang, and M. R. Azghadi, “Sediment prediction in the Great Barrier Reef using vision Transformer with finite element analysis,” *Neural Networks*, vol. 152, pp. 311–321, Aug. 2022.
- [28] M. Jahanbakht, W. Xiang, N. J. Waltham, and M. R. Azghadi, “Distributed deep learning in the cloud and energy-efficient real-time image processing at the edge for fish segmentation in underwater videos,” *IEEE Access*, Aug. 2022.
- [29] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: a survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, Jun. 2015.
- [30] M. H. Eldefrawy, N. Pereira, and M. Gidlund, “Key distribution protocol for industrial internet of things without implicit certificates,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 906–917, Feb. 2019.
- [31] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang, “A comprehensive study on the internet of underwater things: applications, challenges, and channel models,” *Sensors*, vol. 17, no. 7, Jul. 2017.
- [32] M. C. Domingo, “An overview of the internet of underwater things,” *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, Nov. 2012.

- [33] Woods Hole Oceanographic Institution, “Know your ocean,” www.whoi.edu, Feb. 2019.
- [34] Y. Li, Y. Zhang, W. Li, and T. Jiang, “Marine wireless big data: efficient transmission, related applications, and challenges,” *IEEE Wireless Communications*, vol. 25, no. 1, pp. 19–25, Feb. 2018.
- [35] G. Xiao, B. Wang, Z. Deng, M. Fu, and Y. Ling, “An acoustic communication time delays compensation approach for master-slave AUV cooperative navigation,” *IEEE Sensors Journal*, vol. 17, no. 2, pp. 504–513, Jan. 2017.
- [36] F. Javed, M. K. Afzal, M. Sharif, and B. Kim, “Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: a comparative review,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2062–2100, Mar. 2018.
- [37] X. Cao, L. Liu, Y. Cheng, and X. S. Shen, “Towards energy-efficient wireless networking in the big data era: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 303–332, Jan. 2018.
- [38] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, “A survey on network methodologies for real-time analytics of massive IoT data and open research issues,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1457–1477, Apr. 2017.
- [39] L. F. Maimo, A. L. P. Gomez, F. J. G. Clemente, M. G. Perez, and G. M. Perez, “A self-adaptive deep learning-based system for anomaly detection in 5G networks,” *IEEE Access*, vol. 6, pp. 7700–7712, Feb. 2018.
- [40] L. Meng, T. Hirayama, and S. Oyanagi, “Underwater-drone with panoramic camera for automatic fish recognition based on deep learning,” *IEEE Access*, vol. 6, pp. 17 880–17 886, Mar. 2018.
- [41] X. Sun, J. Shi, L. Liu, J. Dong, C. Plant, X. Wang, and H. Zhou, “Transferring deep knowledge for object recognition in low-quality underwater videos,” *Neurocomputing*, vol. 275, pp. 897–908, Jan. 2018.
- [42] N. Ani Brown Mary and D. Dharma, “Coral reef image classification employing improved LDP for feature extraction,” *Journal of Visual Communication and Image Representation*, vol. 49, pp. 225–242, Nov. 2017.

-
- [43] M. Reggiannini and O. Salvetti, "Seafloor analysis and understanding for underwater archeology," *Journal of Cultural Heritage*, vol. 24, pp. 147–156, Apr. 2017.
- [44] M. Jian, Q. Qi, J. Dong, Y. Yin, and K.-M. Lam, "Integrating QDWD with pattern distinctness and local contrast for underwater saliency detection," *Journal of Visual Communication and Image Representation*, vol. 53, pp. 31–41, May 2018.
- [45] T. Schoening, T. Kuhn, D. O. B. Jones, E. Simon-Lledo, and T. W. Nattkemper, "Fully automated image segmentation for benthic resource assessment of polymetallic nodules," *Methods in Oceanography*, vol. 15-16, pp. 78–89, Jul. 2016.
- [46] F. Bonin-Font, A. Burguera, and J. L. Lisani, "Visual discrimination and large area mapping of *Posidonia Oceanica* using a lightweight AUV," *IEEE Access*, vol. 5, pp. 24 479–24 494, Oct. 2017.
- [47] M. C. Chuang, J. N. Hwang, J. H. Ye, S. C. Huang, and K. Williams, "Underwater fish tracking for moving cameras based on deformable multiple kernels," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 9, pp. 2467–2477, Sep. 2017.
- [48] P. Y. Lau, P. L. Correia, P. Fonseca, and A. Campos, "Estimating Norway lobster abundance from deep-water videos: an automatic approach," *IET Image Processing*, vol. 6, no. 1, pp. 22–30, Feb. 2012.
- [49] J. Osterloff, I. Nilssen, and T. W. Nattkemper, "A computer vision approach for monitoring the spatial and temporal shrimp distribution at the LoVe observatory," *Methods in Oceanography*, vol. 15-16, pp. 114–128, Jul. 2016.
- [50] M. S. Schmid, C. Aubry, J. Grigor, and L. Fortier, "The LOKI underwater imaging system and an automatic identification model for the detection of zooplankton taxa in the Arctic Ocean," *Methods in Oceanography*, vol. 15-16, pp. 129–160, Jul. 2016.
- [51] R. Faillettaz, M. Picheral, J. Y. Luo, C. Guigand, R. K. Cowen, and J.-O. Irisson, "Imperfect automatic image classification successfully describes plankton distribution patterns," *Methods in Oceanography*, vol. 15-16, pp. 60–77, Jul. 2016.
- [52] J. Pearlman, S. Jirka, J. d. Rio, E. Delory, L. Frommhold, S. Martinez, and T. O. Reilly, "Oceans of Tomorrow sensor interoperability for in-situ ocean monitoring," in *Proceedings OCEANS*. Monterey, USA: IEEE, Sep. 2016.
- [53] G. L. Foresti, "Visual inspection of sea bottom structures by an autonomous underwater vehicle," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 5, pp. 691–705, Oct. 2001.

- [54] L. Paull, S. Saeedi, M. Seto, and H. Li, “AUV navigation and localization: a review,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, Jan. 2014.
- [55] T. Miyoshi *et al.*, ““big data assimilation” toward post-petascale severe weather prediction: an overview and progress,” *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2155–2179, Nov. 2016.
- [56] C. Kalyvas, A. Kokkos, and T. Tzouramanis, “A survey of official online sources of high-quality free-of-charge geospatial data for maritime geographic information systems applications,” *Information Systems*, vol. 65, pp. 36–51, Apr. 2017.
- [57] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, “The role of big data analytics in internet of things,” *Computer Networks*, vol. 129, pp. 459–471, Dec. 2017.
- [58] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, Jun. 2018.
- [59] D. Huang, D. Zhao, L. Wei, Z. Wang, and Y. Du, “Modeling and analysis in marine big data: advances and challenges,” *Mathematical Problems in Engineering*, vol. 2015, Sep. 2015.
- [60] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, “Underwater internet of things in smart ocean: system architecture and open issues,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4297–4307, Jul. 2020.
- [61] D. Raj, J. Lee, E. Ko, S. Shin, J.-I. Namgung, S.-H. Yum, and S.-H. Park, “Underwater network management system in internet of underwater things: open challenges, benefits, and feasible solution,” *Electronics*, vol. 9, pp. 1142–1175, Jul. 2020.
- [62] R. Khalil, M. Babar, T. Jan, and N. Saeed, “Towards the internet of underwater things: recent developments and future challenges,” *IEEE Consumer Electronics Magazine*, May 2020.
- [63] Z. Zhou, B. Yao, R. Xing, L. Shu, and S. Bu, “E-CARP: an energy efficient routing protocol for UWSNs in the internet of underwater things,” *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4072–4082, Jun. 2016.
- [64] F. Adamo, F. Attivissimo, C. G. C. Carducci, and A. M. L. Lanzolla, “A smart sensor network for sea water quality monitoring,” *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2514–2522, May 2015.

-
- [65] T. P. Hughes *et al.*, “Global warming and recurrent mass bleaching of corals,” *Nature*, vol. 543, pp. 373–377, Mar. 2017.
- [66] C. A. Balanis, *Antenna theory: analysis and design*. John Wiley & Sons, 2016.
- [67] M. A. Ainslie and J. G. McColm, “A simplified formula for viscous and chemical absorption in sea water,” *Journal of the Acoustical Society of America*, vol. 103, no. 3, pp. 1671–1672, Jun. 1998.
- [68] C. Gabriel, M. A. Khalighi, S. Bourennane, P. Leon, and V. Rigaud, “Monte-Carlo-based channel characterization for underwater optical communication systems,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 1, Jan. 2013.
- [69] E. Illi, F. E. Bouanani, K. Park, F. Ayoub, and M. Alouini, “An improved accurate solver for the time-dependent RTE in underwater optical wireless communications,” *IEEE Access*, vol. 7, pp. 96 478–96 494, Jul. 2019.
- [70] J. S. Jaffe, “Underwater optical imaging: the past, the present, and the prospects,” *IEEE Journal of Oceanic Engineering*, vol. 40, no. 3, pp. 683–700, Jul. 2015.
- [71] J. Wang, H. Zhou, Y. Li, Q. Sun, Y. Wu, S. Jin, T. Q. S. Quek, and C. Xu, “Wireless channel models for maritime communications,” *IEEE Access*, vol. 6, pp. 68 070–68 088, Dec. 2018.
- [72] W. Sun, X. Yuan, J. Wang, Q. Li, L. Chen, and D. Mu, “End-to-end data delivery reliability model for estimating and optimizing the link quality of industrial WSNs,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1127–1137, Jul. 2018.
- [73] H. Zhang, X. Liu, C. Li, Y. Chen, X. Che, L. Y. Wang, F. Lin, and G. Yin, “Scheduling with predictable link reliability for wireless networked control,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6135–6150, Sep. 2017.
- [74] R. D. Gomes, C. Benavente-Peces, I. E. Fonseca, and M. S. Alencar, “Adaptive and beacon-based multi-channel protocol for industrial wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 132, pp. 22–39, Apr. 2019.
- [75] S. Murali and A. Jamalipour, “Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2593–2601, Apr. 2019.

- [76] J. C. Fink, K. Fermanich, and T. Ehlinger, *The effects of urbanization on Baird Creek, Green Bay, Wisconsin*. University of Wisconsin-Green Bay, 2005.
- [77] Z. Li and Y. Wu, “Smooth mobility and link reliability-based optimized link state routing scheme for MANETs,” *IEEE Communications Letters*, vol. 21, no. 7, pp. 1529–1532, Jul. 2017.
- [78] H. Tran-Dang and D. Kim, “Channel-aware cooperative routing in underwater acoustic sensor networks,” *Journal of Communications and Networks*, vol. 21, no. 1, pp. 33–44, Feb. 2019.
- [79] R. Zhao, H. Long, O. A. Dobre, X. Shen, T. M. N. Ngatched, and H. Mei, “Time reversal based MAC for multi-hop underwater acoustic networks,” *IEEE Systems Journal*, vol. 13, no. 3, pp. 2531–2542, Sep. 2019.
- [80] P. Qarabaqi and M. Stojanovic, “Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels,” *IEEE Journal of Oceanic Engineering*, vol. 38, no. 4, pp. 701–717, Oct. 2013.
- [81] J. Zhou, H. Jiang, P. Wu, and Q. Chen, “Study of propagation channel characteristics for underwater acoustic communication environments,” *IEEE Access*, vol. 7, pp. 79 438–79 445, Jun. 2019.
- [82] M. Naderi, A. G. Zajić, and M. Pätzold, “A nonisovelocity geometry-based underwater acoustic channel model,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 2864–2879, Apr. 2018.
- [83] M. Naderi, M. Pätzold, R. Hicheri, and N. Youssef, “A geometry-based underwater acoustic channel model allowing for sloped ocean bottom conditions,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2394–2408, Apr. 2017.
- [84] W. Li, Z. Huang, M. Gong, and Y. Ji, “Land-to-underwater optical communication system based on underwater visible light communication network units and fiber links,” in *Proceedings Asia Communications and Photonics Conference*. Wuhan, China: IEEE, Nov. 2016.
- [85] W. Li, S. Zhou, P. Willett, and Q. Zhang, “Preamble detection for underwater acoustic communications based on sparse channel identification,” *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 256–268, Jan. 2019.
- [86] M. Liang, J. Duan, and D. Zhao, “Optimal redundancy control strategy for fountain code-based underwater acoustic communication,” *IEEE Access*, vol. 6, pp. 69 321–69 334, Nov. 2018.

-
- [87] C. Wang, Z. Wang, W. Sun, and D. R. Fuhrmann, "Reinforcement learning-based adaptive transmission in time-varying underwater acoustic channels," *IEEE Access*, vol. 6, pp. 2541–2558, Dec. 2017.
- [88] S. Jiang, "State-of-the-art Medium Access Control (MAC) protocols for underwater acoustic networks: a survey based on a MAC reference model," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 96–131, Jan. 2018.
- [89] J. Liu, G. Shou, Y. Liu, Y. Hu, and Z. Guo, "Performance evaluation of integrated multi-access edge computing and fiber-wireless access networks," *IEEE Access*, vol. 6, pp. 30 269–30 279, May 2018.
- [90] T. Wang, D. Zhao, S. Cai, W. Jia, and A. Liu, "Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4791–4799, Jul. 2020.
- [91] S. Cai, Y. Zhu, T. Wang, G. Xu, A. Liu, and X. Liu, "Data collection in underwater sensor networks based on mobile edge computing," *IEEE Access*, vol. 7, pp. 65 357–65 367, May 2019.
- [92] Q. Wang, H. Dai, Q. Wang, M. K. Shukla, W. Zhang, and C. G. Soares, "On connectivity of UAV-assisted data acquisition for underwater internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5371–5385, Mar. 2020.
- [93] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, Jun. 2018.
- [94] X. Di, Y. Zhang, T. Liu, S. Kang, and Y. Zhao, "Mobile fog computing-assisted resource allocation for two-hop SWIPT OFDM networks," *Wireless Communications and Mobile Computing*, vol. 2018, Sep. 2018.
- [95] S. Safavat, N. N. Sapavath, and D. B. Rawat, "Recent advances in mobile edge computing and content caching," *Digital Communications and Networks*, Sep. 2019.
- [96] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
- [97] M. S. Brown, *Data mining for dummies*, 1st ed. John Wiley & Sons, Sep. 2014.
- [98] S. Chhabra and D. Singh, "Data fusion and data aggregation/summarization techniques in WSNs: a review," *International Journal of Computer Applications*, vol. 121, no. 19, Jul. 2015.

- [99] S. Misra and R. Kumar, "An analytical study of LEACH and PEGASIS protocol in wireless sensor network," in *Proceedings International Conference on Innovations in Information, Embedded and Communication Systems*. Coimbatore, India: IEEE, Mar. 2017.
- [100] S. Chand, S. Singh, and B. Kumar, "Heterogeneous HEED protocol for wireless sensor networks," *Wireless personal communications*, vol. 77, no. 3, pp. 2117–2139, Feb. 2014.
- [101] J. Ma, S. Wang, C. Meng, Y. Ge, and J. Du, "Hybrid energy-efficient APTEEN protocol based on ant colony algorithm in wireless sensor network," *Wireless Communications and Networking*, vol. 2018, no. 1, pp. 102–117, Dec. 2018.
- [102] J. Qiu, Z. Xing, C. Zhu, K. Lu, J. He, Y. Sun, and L. Yin, "Centralized fusion based on interacting multiple model and adaptive Kalman filter for target tracking in underwater acoustic sensor networks," *IEEE Access*, vol. 7, pp. 25 948–25 958, Feb. 2019.
- [103] F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal*, vol. 2013, Sep. 2013.
- [104] K. Nomura, D. Sugimura, and T. Hamamoto, "Underwater image color correction using exposure-bracketing imaging," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 893–897, Jun. 2018.
- [105] S. N. Zulkifli, H. A. Rahim, and W.-J. Lau, "Detection of contaminants in water supply: a review on state-of-the-art monitoring technologies and their applications," *Sensors and Actuators B: Chemical*, vol. 255, pp. 2657–2689, Feb. 2018.
- [106] P. Kassal, M. D. Steinberg, and I. M. Steinberg, "Wireless chemical sensors and biosensors: a review," *Sensors and Actuators B: Chemical*, vol. 266, pp. 228–245, Aug. 2018.
- [107] J. E. Hewitt and S. F. Thrush, "Monitoring for tipping points in the marine environment," *Journal of Environmental Management*, vol. 234, pp. 131–137, Mar. 2019.
- [108] N. G. Wright and H. K. Chan, "Low-cost internet of things ocean observation," in *Proceedings OCEANS*. Monterey, USA: IEEE, Sep. 2016.
- [109] M. Palaniswami, A. S. Rao, and S. Bainbridge, "Real-time monitoring of the Great Barrier Reef using internet of things with big data analytics," *ITU Journal: ICT Discoveries*, vol. 1, no. 13, Oct. 2017.

-
- [110] O. Godø, S. Johnsen, and T. Torkelsen, “The LoVe ocean observatory is in operation,” *Marine Technology Society Journal*, vol. 48, Mar. 2014.
- [111] A. S. M. Shihavuddin, N. Gracias, R. Garcia, C. A. Gleason, and B. Gintert, “Image-based coral reef classification and thematic mapping,” *Remote Sensing*, vol. 5, no. 4, Apr. 2013.
- [112] M. Han, Z. Lyu, T. Qiu, and M. Xu, “A review on intelligence dehazing and color restoration for underwater images,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, Jan. 2018.
- [113] S. Q. Duntley, “Light in the sea,” *Journal of the Optical Society of America*, vol. 53, no. 2, pp. 214–233, 1963.
- [114] J. R. V. Zaneveld and W. S. Pegau, “Robust underwater visibility parameter,” *Optics Express*, vol. 11, no. 23, pp. 2997–3009, Nov. 2003.
- [115] A. A. M. Salih, K. Hasikin, and N. A. M. Isa, “Adaptive fuzzy exposure local contrast enhancement,” *IEEE Access*, vol. 6, pp. 58 794–58 806, Sep. 2018.
- [116] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, “Automatic fish classification for underwater species behavior understanding,” in *Proceedings 1st International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*. Firenze, Italy: ACM, Oct. 2010, pp. 45–50.
- [117] M. C. Chuang, J. N. Hwang, K. Williams, and R. Towler, “Tracking live fish from low-contrast and low-frame-rate stereo videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 167–179, Jan. 2015.
- [118] S. Negahdaripour, “On 3-D motion estimation from feature tracks in 2-D FS sonar video,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 1016–1030, Aug. 2013.
- [119] C. Cheng and H. Yang, “Realization of 3D stereo imaging for underwater applications,” in *Proceedings International Conference of Intelligent Robotic and Control Engineering*. Lanzhou, China: IEEE, Aug. 2018, pp. 242–246.
- [120] L. Mazzei, L. Corgnati, S. Marini, E. Ottaviani, and B. Isoppo, “Low cost stereo system for imaging and 3D reconstruction of underwater organisms,” in *Proceedings OCEANS*. Genoa, Italy: IEEE, May 2015.
- [121] H. Assalih, Y. Petillot, and J. Bell, “Acoustic stereo imaging (ASI) system,” in *Proceedings OCEANS*. Bremen, Germany: IEEE, May 2009.

- [122] A. Anwer, S. S. A. Ali, A. Khan, and F. Mériaudeau, “Real-time underwater 3D scene reconstruction using commercial depth sensor,” in *Proceedings International Conference on Underwater System Technology: Theory and Applications*. Penang, Malaysia: IEEE, Dec. 2016, pp. 67–70.
- [123] C.-C. Wang, S.-W. Shyue, and S.-H. Cheng, “Underwater structure inspection with laser light stripes,” in *Proceedings International Symposium on Underwater Technology*. Tokyo, Japan: IEEE, May 2000, pp. 201–205.
- [124] S. Ishibashi, “The stereo vision system for an underwater vehicle,” in *Proceedings OCEANS*. Bremen, Germany: IEEE, May 2009.
- [125] S. Negahdaripour, H. Sekkati, and H. Pirsiavash, “Opti-acoustic stereo imaging: on system calibration and 3-D target reconstruction,” *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1203–1214, Jun. 2009.
- [126] A. Marouchos, M. Sherlock, A. Filisetti, and A. Williams, “Underwater imaging on self-contained tethered systems,” in *Proceedings OCEANS*. Anchorage, USA: IEEE, Sep. 2017.
- [127] F. Westling, C. Sun, and D. Wang, “A modular learning approach for fish counting and measurement using stereo baited remote underwater video,” in *Proceedings International Conference on Digital Image Computing: Techniques and Applications*. Wollongong, Australia: IEEE, Nov. 2014.
- [128] C. Mulsow and H. Maas, “A universal approach for geometric modelling in underwater stereo image processing,” in *Proceedings ICPR Workshop on Computer Vision for Analysis of Underwater Imagery*. Stockholm, Sweden: IEEE, Aug. 2014, pp. 49–56.
- [129] E. Trucco and K. Plakas, “Video tracking: a concise survey,” *IEEE Journal of Oceanic Engineering*, vol. 31, no. 2, pp. 520–529, Apr. 2006.
- [130] M. Johnson-Roberson, S. Kumar, O. Pizarro, and S. Willams, “Stereoscopic imaging for coral segmentation and classification,” in *Proceedings OCEANS*. Boston, USA: IEEE, Sep. 2006.
- [131] W. Xingang, “A research review of distributed computing system,” in *Recent Developments in Intelligent Computing, Communication and Devices*. Springer, Aug. 2018, pp. 357–368.

-
- [132] Apache Software Foundation, “Apache project directory,” projects.apache.org, Feb. 2019.
- [133] B. Power and J. Weinman, “Revenue growth is the primary benefit of the cloud,” *IEEE Cloud Computing*, vol. 5, no. 4, pp. 89–94, Jul. 2018.
- [134] Qubole Inc., “Big data trends and challenges,” go.qubole.com, Oct. 2018.
- [135] S. Zhelev and A. Rozeva, “Big data processing in the cloud - challenges and platforms,” in *Proceedings 43rd International Conference on Applications of Mathematics in Engineering and Economics*. Sozopol, Bulgaria: AIP, Dec. 2017.
- [136] J. N. Gomes-Pereira *et al.*, “Current and future trends in marine image annotation software,” *Progress in Oceanography*, vol. 149, pp. 106–120, Dec. 2016.
- [137] V. Barale, “A supporting marine information system for maritime spatial planning: the European atlas of the seas,” *Ocean & Coastal Management*, Dec. 2018.
- [138] P. Liu, B. Wang, Z. Deng, and M. Fu, “INS/DVL/PS tightly coupled underwater navigation method with limited DVL measurements,” *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2994–3002, Apr. 2018.
- [139] T. Zhang, L. Chen, and Y. Yan, “Underwater positioning algorithm based on SINS/LBL integrated system,” *IEEE Access*, vol. 6, pp. 7157–7163, Jan. 2018.
- [140] B. Wang, J. Zhu, Z. Deng, and M. Fu, “A characteristic parameter matching algorithm for gravity-aided navigation of underwater vehicles,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1203–1212, Feb. 2019.
- [141] Y. Chen, D. Zheng, P. A. Miller, and J. A. Farrell, “Underwater inertial navigation with long baseline transceivers: a near-real-time approach,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 240–251, Jan. 2016.
- [142] Z. Song and K. Mohseni, “Long-term inertial navigation aided by dynamics of flow field features,” *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 940–954, Oct. 2018.
- [143] P. Batista, “GES long baseline navigation with unknown sound velocity and discrete-time range measurements,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 219–230, Jan. 2015.

- [144] Y. Han, C. Shi, D. Sun, and J. Zhang, “Research on integrated navigation algorithm based on ranging information of single beacon,” *Applied Acoustics*, vol. 131, pp. 203–209, Feb. 2018.
- [145] C. Renner, “Packet-based ranging with a low-power, low-cost acoustic modem for micro AUVs,” in *Proceedings 11th ITG International Conference on Systems, Communications and Coding*. Hamburg, Germany: VDE, Feb. 2017.
- [146] T. A. Sarmiento and R. R. Murphy, “Insights on obstacle avoidance for small unmanned aerial systems from a study of flying animal behavior,” *Robotics and Autonomous Systems*, vol. 99, pp. 17–29, Jan. 2018.
- [147] F. Menna, P. Agrafiotis, and A. Georgopoulos, “State of the art and applications in archaeological underwater 3D recording and mapping,” *Journal of Cultural Heritage*, Oct. 2018.
- [148] A. Al-Kaff, D. Martín, F. García, A. d. l. Escalera, and J. María Armingol, “Survey of computer vision algorithms and applications for unmanned aerial vehicles,” *Expert Systems with Applications*, vol. 92, pp. 447–463, Feb. 2018.
- [149] J. Gao, A. A. Proctor, Y. Shi, and C. Bradley, “Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control,” *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2323–2334, Oct. 2016.
- [150] N. Palomeras, N. Hurtós, M. Carreras, and P. Ridao, “Autonomous mapping of underwater 3-D structures: from view planning to execution,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1965–1971, Jul. 2018.
- [151] P. Norgren and R. Skjetne, “A multibeam-based SLAM algorithm for iceberg mapping using AUVs,” *IEEE Access*, vol. 6, pp. 26 318–26 337, Apr. 2018.
- [152] M. Føre, K. Frank, T. Dempster, J. A. Alfredsen, and E. Høy, “Biomonitoring using tagged sentinel fish and acoustic telemetry in commercial salmon aquaculture: a feasibility study,” *Aquacultural Engineering*, vol. 78, pp. 163–172, Aug. 2017.
- [153] J. Schmidhuber, “Deep learning in neural networks: an overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.
- [154] Y. Noh, B. Zhang, and D. D. Lee, “Generative local metric learning for nearest neighbor classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 106–118, Jan. 2018.

-
- [155] E. Adeli, K. Thung, L. An, G. Wu, F. Shi, T. Wang, and D. Shen, “Semi-supervised discriminative classification robust to sample-outliers and feature-noises,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 515–522, Feb. 2019.
- [156] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G. Yang, “Deep learning for health informatics,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4–21, Jan. 2017.
- [157] L. Xiao, G. Sheng, X. Wan, W. Su, and P. Cheng, “Learning-based PHY-layer authentication for underwater sensor networks,” *IEEE Communications Letters*, vol. 23, no. 1, pp. 60–63, Jan. 2019.
- [158] C. C. Aggarwal, *Data mining: the textbook*. Springer, 2015.
- [159] S. Gupta and S. Roy, “Medav filter—filter for removal of image noise with the combination of median and average filters,” in *Proceedings Recent Trends in Signal and Image Processing*. Singapore: Springer, May 2018, pp. 11–19.
- [160] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, 7th ed. Springer, Jun. 2017.
- [161] M. Wang, X. Liu, Y. Gao, X. Ma, and N. Q. Soomro, “Superpixel segmentation: a benchmark,” *Signal Processing: Image Communication*, vol. 56, pp. 28–39, Aug. 2017.
- [162] C. Li, J. Guo, and C. Guo, “Emerging from water: underwater image color correction based on weakly supervised color transfer,” *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 323–327, Mar. 2018.
- [163] Y. Peng and P. C. Cosman, “Underwater image restoration based on image blurriness and light absorption,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1579–1594, Apr. 2017.
- [164] N. Wang, H. Zheng, and B. Zheng, “Underwater image restoration via maximum attenuation identification,” *IEEE Access*, vol. 5, pp. 18 941–18 952, Sep. 2017.
- [165] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, “Structure-revealing low-light image enhancement via robust retinex model,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2828–2841, Jun. 2018.

- [166] Y. Wang, N. Li, Z. Li, Z. Gu, H. Zheng, B. Zheng, and M. Sun, “An imaging-inspired no-reference underwater color image quality assessment metric,” *Computers & Electrical Engineering*, vol. 70, pp. 904–913, Aug. 2018.
- [167] K. Panetta, C. Gao, and S. Agaian, “Human-visual-system-inspired underwater image quality measures,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 541–551, Jul. 2016.
- [168] M. Yang and A. Sowmya, “An underwater color image quality evaluation metric,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 6062–6071, Dec. 2015.
- [169] C. O. Ancuti, C. Ancuti, C. D. Vleeschouwer, and P. Bekaert, “Color balance and fusion for underwater image enhancement,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 379–393, Jan. 2018.
- [170] J. Rife and S. M. Rock, “Segmentation methods for visual tracking of deep-ocean jellyfish using a conventional camera,” *IEEE Journal of Oceanic Engineering*, vol. 28, no. 4, pp. 595–608, Oct. 2003.
- [171] D. M. Kocak, N. d. V. Lobo, and E. A. Widder, “Computer vision techniques for quantifying, tracking, and identifying bioluminescent plankton,” *IEEE Journal of Oceanic Engineering*, vol. 24, no. 1, pp. 81–95, Jan. 1999.
- [172] T. Schoening, “Automated detection in benthic images for megafauna classification and marine resource exploration: supervised and unsupervised methods for classification and regression tasks in benthic images with efficient integration of expert knowledge,” Ph.D. dissertation, Der Technischen Fakultät der Universität Bielefeld, 2015.
- [173] M. C. Chuang, J. N. Hwang, K. Williams, and R. Towler, “Automatic fish segmentation via double local thresholding for trawl-based underwater camera systems,” in *Proceedings 18th International Conference on Image Processing*. Brussels, Belgium: IEEE, Sep. 2011, pp. 3145–3148.
- [174] T. Schoening, M. Bergmann, J. Ontrup, J. Taylor, J. Dannheim, J. Gutt, A. Purser, and T. W. Nattkemper, “Semi-automated image analysis for the assessment of megafaunal densities at the arctic deep-sea observatory HAUSGARTEN,” *PLoS ONE*, vol. 7, no. 6, pp. 38 179–38 193, Jun. 2012.
- [175] L. Tong, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins, “Recognizing plankton images from the shadow image particle profiling eval-

- uation recorder,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 4, pp. 1753–1762, Aug. 2004.
- [176] Q. Hu and C. Davis, “Accurate automatic quantification of taxa-specific plankton abundance using dual classification with correction,” *Marine Ecology Progress Series*, vol. 306, pp. 51–61, Jan. 2006.
- [177] A. Purser, M. Bergmann, T. Lundälv, and T. Nattkemper, “Use of machine-learning algorithms for the automated detection of cold-water coral habitats: a pilot study,” *Marine Ecology Progress Series*, vol. 397, pp. 241–249, Dec. 2009.
- [178] X. Li, J. Song, F. Zhang, X. Ouyang, and S. U. Khan, “MapReduce-based fast fuzzy c-means algorithm for large-scale underwater image segmentation,” *Future Generation Computer Systems*, vol. 65, pp. 90–101, Dec. 2016.
- [179] Z. Peizhe and L. Canping, “Region-based color image segmentation of fishes with complex background in water,” in *Proceedings International Conference on Computer Science and Automation Engineering*, vol. 1. Shanghai, China: IEEE, Jun. 2011, pp. 596–600.
- [180] S. L. Benfield, H. M. Guzman, J. M. Mair, and J. A. T. Young, “Mapping the distribution of coral reefs and associated sublittoral habitats in Pacific Panama: a comparison of optical satellite sensors and classification methodologies,” *International Journal of Remote Sensing*, vol. 28, no. 22, pp. 5047–5070, Oct. 2007.
- [181] M. C. Chuang, J. N. Hwang, and K. Williams, “A feature learning and object recognition framework for underwater fish images,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1862–1872, Apr. 2016.
- [182] P. X. Huang, B. J. Boom, and R. B. Fisher, “Hierarchical classification with reject option for live fish recognition,” *Machine Vision and Applications*, vol. 26, no. 1, pp. 89–102, Jan. 2015.
- [183] M. Chambah, D. Semani, A. Renouf, P. Coutellemont, and A. Rizzi, “Underwater color constancy: enhancement of automatic live fish recognition,” in *Proceedings 16th Annual symposium on electronic imaging*. San Jose, USA: International Society for Optics and Photonics, Dec. 2003, pp. 157–168.
- [184] D. Zhang, G. Kopanas, C. Desai, S. Chai, and M. Piacentino, “Unsupervised underwater fish detection fusing flow and objectiveness,” in *Proceedings IEEE Winter Applications of Computer Vision Workshops*. Lake Placid, USA: IEEE, Mar. 2016.

- [185] S. Pons, J. Piera, and J. Aguzzi, “Video-image processing applied to the analysis of the behaviour of deep-water lobsters (*Nephrops Norvegicus*),” in *Proceedings OCEANS*. Sydney, Australia: IEEE, May 2010.
- [186] G. W. Flake, R. E. Tarjan, and K. Tsioutsoulis, “Graph clustering and minimum cut trees,” *Internet Mathematics*, vol. 1, no. 4, pp. 385–408, 2004.
- [187] A. Lüdtke, K. Jerosch, O. Herzog, and M. Schlüter, “Development of a machine learning technique for automatic analysis of seafloor image data: case example, Pogonophora coverage at mud volcanoes,” *Computers & Geosciences*, vol. 39, pp. 120–128, Feb. 2012.
- [188] M. Pugh, B. Tiddeman, H. Dee, and P. Hughes, “Towards automated classification of seabed substrates in underwater video,” in *Proceedings ICPR Workshop on Computer Vision for Analysis of Underwater Imagery*. Stockholm, Sweden: IEEE, Aug. 2014, pp. 9–16.
- [189] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. Fisher, “Detecting, tracking and counting fish in low quality unconstrained underwater videos,” in *Proceedings 3rd International Conference on Computer Vision Theory and Applications*, vol. 2, Madeira, Portugal, 2008, pp. 514–519.
- [190] M. M. M. Fouad, H. M. Zawbaa, N. El-Bendary, and A. E. Hassanien, “Automatic Nile Tilapia fish classification approach using machine learning techniques,” in *Proceedings 13th International Conference on Hybrid Intelligent Systems*. Gammarth, Tunisia: IEEE, Dec. 2013, pp. 173–178.
- [191] F. Bonin-Font, M. M. Campos, and G. O. Codina, “Towards visual detection, mapping and quantification of *Posidonia oceanica* using a lightweight auv,” *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 500–505, 2016.
- [192] M. C. Chuang, J. N. Hwang, F. F. Kuo, M. K. Shan, and K. Williams, “Recognizing live fish species by hierarchical partial classification based on the exponential benefit,” in *Proceedings International Conference on Image Processing*. Paris, France: IEEE, Oct. 2014, pp. 5232–5236.
- [193] M. Elawady, “Sparse coral classification using deep convolutional neural networks,” *Computing Research Repository*, Nov. 2015.
- [194] P. R. Ambika and A. B. Malakreddy, “Review of existing research contribution toward dimensional reduction methods in high-dimensional data,” in *Proceedings*

-
- International Conference on Computer Networks and Communication Technologies*, Seoul, South Korea, Jun. 2018, pp. 21–23.
- [195] S. Gepshtein and Y. Keller, “Iterative spectral independent component analysis,” *Signal Processing*, vol. 155, pp. 368–376, Feb. 2019.
- [196] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, B. Brisco, S. Mahdavi, M. Amani, and J. E. Granger, “Fisher linear discriminant analysis of coherency matrix for wetland classification using PolSAR imagery,” *Remote Sensing of Environment*, vol. 206, pp. 300–317, Mar. 2018.
- [197] X. Tang, Y. Dai, and Y. Xiang, “Feature selection based on feature interactions with application to text categorization,” *Expert Systems with Applications*, vol. 120, pp. 207–216, Apr. 2019.
- [198] R. Zhao and K. Mao, “Fuzzy bag-of-words model for document representation,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, Apr. 2018.
- [199] L. H. Ceze, *Encyclopedia of parallel computing*. Springer, 2011, ch. Shared-Memory Multiprocessors, pp. 1810–1812.
- [200] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. Yoo, “UNPU: an energy-efficient deep neural network accelerator with fully variable weight bit precision,” *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 173–185, Jan. 2019.
- [201] S. I. Venieris and C. Bouganis, “fpgaConvNet: mapping regular and irregular convolutional neural networks on FPGAs,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 326–342, Feb. 2019.
- [202] T. Manderson and G. Dudek, “GPU-assisted learning on an autonomous marine robot for vision-based navigation and image understanding,” in *Proceedings OCEANS*. Charleston, USA: IEEE, Oct. 2018.
- [203] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed systems: concepts and design*, 5th ed. Pearson Education, 2011.
- [204] D. Xia, H. Li, B. Wang, Y. Li, and Z. Zhang, “A map reduce-based nearest neighbor approach for big-data-driven traffic flow prediction,” *IEEE Access*, vol. 4, pp. 2920–2934, Jun. 2016.
- [205] Z. Xu, B. Liu, S. Zhe, H. Bai, Z. Wang, and J. Neville, “Variational random function model for network modeling,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 318–324, Jan. 2019.

- [206] Y. Ming, E. Zhu, M. Wang, Y. Ye, X. Liu, and J. Yin, “DMP-ELMs: data and model parallel extreme learning machines for large-scale learning tasks,” *Neurocomputing*, vol. 320, pp. 85–97, Dec. 2018.
- [207] S. Garcia, D. Jeon, C. Louie, and M. B. Taylor, “The Kremlin oracle for sequential code parallelization,” *IEEE Micro*, vol. 32, no. 4, pp. 42–53, Jul. 2012.
- [208] E. Schwartz, R. Giryes, and A. M. Bronstein, “DeepISP: toward learning an end-to-end image processing pipeline,” *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912–923, Feb. 2019.
- [209] M. Modasshir, A. Q. Li, and I. Rekleitis, “Deep neural networks: a comparison on different computing platforms,” in *Proceedings 15th Conference on Computer and Robot Vision*. Toronto, Canada: IEEE, May 2018, pp. 383–389.
- [210] H. Kim, J. Koo, D. Kim, B. Park, Y. Jo, H. Myung, and D. Lee, “Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle,” *IEEE Access*, vol. 7, pp. 179 420–179 428, Dec. 2019.
- [211] V. Pallayil, M. Chitre, S. Kuselan, A. Raichur, M. Ignatius, and J. R. Potter, “Development of a second-generation underwater acoustic ambient noise imaging camera,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 1, pp. 175–189, Jan. 2016.
- [212] S. Karabchevsky, D. Kahana, O. Ben-Harush, and H. Guterman, “FPGA-based adaptive speckle suppression filter for underwater imaging sonar,” *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 646–657, Oct. 2011.
- [213] C. Szegedy, S. Loffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings 31st AAAI Conference on Artificial Intelligence*. San Francisco, USA: AAAI, Feb. 2017.
- [214] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, USA: IEEE, 2016, pp. 770–778.
- [215] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012, pp. 1097–1105.
- [216] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv*, Dec. 2013.
- [217] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, Apr. 2015.

-
- [218] A. Paszke and A. Jackson, “Torch-opCounter,” www.github.com, 2016.
- [219] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, Oct. 1986.
- [220] J. Adler and O. Öktem, “Solving ill-posed inverse problems using iterative deep neural networks,” *Inverse Problems*, vol. 33, no. 12, pp. 124 007–124 011, May 2017.
- [221] Y. Li, H. Lu, J. Li, X. Li, Y. Li, and S. Serikawa, “Underwater image de-scattering and classification by deep neural network,” *Computers & Electrical Engineering*, vol. 54, pp. 68–77, Aug. 2016.
- [222] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, “Blind image quality assessment using a deep bilinear convolutional neural network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36–47, Jan. 2020.
- [223] Y. Yan, H. Hao, B. Xu, J. Zhao, and F. Shen, “Image clustering via deep embedded dimensionality reduction and probability-based triplet loss,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5652–5661, Apr. 2020.
- [224] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, “DeepFish: accurate underwater live fish recognition with a deep architecture,” *Neurocomputing*, vol. 187, pp. 49–58, Apr. 2016.
- [225] N. Wahidin, V. P. Siregar, B. Nababan, I. Jaya, and S. Wouthuyzen, “Object-based image analysis for coral reef benthic habitat mapping with several classification algorithms,” *Procedia Environmental Sciences*, vol. 24, pp. 222–227, Dec. 2015.
- [226] G. Huo, Z. Wu, and J. Li, “Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data,” *IEEE Access*, vol. 8, pp. 47 407–47 418, Mar. 2020.
- [227] L. Jin, H. Liang, and C. Yang, “Accurate underwater ATR in forward-looking sonar imagery using deep convolutional neural networks,” *IEEE Access*, vol. 7, pp. 125 522–125 531, Sep. 2019.
- [228] S. W. Perry and L. Guan, “A recurrent neural network for detecting objects in sequences of sector-scan sonar images,” *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 857–871, Jul. 2004.
- [229] H. Singh, J. Howland, and O. Pizarro, “Advances in large-area photomosaicking underwater,” *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 872–886, Jul. 2004.

- [230] R. Nian, B. He, B. Zheng, M. van Heeswijk, Q. Yu, Y. Miche, and A. Lendasse, “Extreme learning machine towards dynamic model hypothesis in fish ethology research,” *Neurocomputing*, vol. 128, pp. 273–284, Mar. 2014.
- [231] S. Luo, X. Li, D. Wang, J. Li, and C. Sun, “Automatic fish recognition and counting in video footage of fishery operations,” in *Proceedings International Conference on Computational Intelligence and Communication Networks*. Jabalpur, India: IEEE, Dec. 2015, pp. 296–299.
- [232] J. Wang, G. Wu, L. Wan, Y. Sun, and D. Jiang, “Recurrent neural network applied to fault diagnosis of underwater robots,” in *Proceedings International Conference on Intelligent Computing and Intelligent Systems*, vol. 1. Shanghai, China: IEEE, Nov. 2009, pp. 593–598.
- [233] Y. Shan, Z. Yan, and J. Wang, “Model predictive control of underwater gliders based on a one-layer recurrent neural network,” in *Proceedings 6th International Conference on Advanced Computational Intelligence*. Hangzhou, China: IEEE, Oct. 2013, pp. 328–333.
- [234] Y. Zhang, P. J. Thorburn, W. Xiang, and P. Fitch, “SSIM—a deep learning approach for recovering missing time series sensor data,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6618–6628, Aug. 2019.
- [235] Y. Gou, T. Zhang, J. Liu, L. Wei, and J. Cui, “DeepOcean: a general deep learning framework for spatio-temporal ocean sensing data prediction,” *IEEE Access*, vol. 8, pp. 79 192–79 202, Apr. 2020.
- [236] Y. Yang, J. Dong, X. Sun, E. Lima, Q. Mu, and X. Wang, “A CFCC-LSTM model for sea surface temperature prediction,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 207–211, Dec. 2018.
- [237] H. Kim and J. Y. Kim, “Environmental sound event detection in wireless acoustic sensor networks for home telemonitoring,” *China Communications*, vol. 14, no. 9, Sep. 2017.
- [238] A. Zyner, S. Worrall, and E. Nebot, “A recurrent neural network solution for predicting driver intention at unsignalized intersections,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1759–1764, Jul. 2018.
- [239] F. S. Marzano, G. Rivolta, E. Coppola, B. Tomassetti, and M. Verdecchia, “Rainfall nowcasting from multisatellite passive-sensor images using a recurrent neural net-

- work,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 11, pp. 3800–3812, Nov. 2007.
- [240] A. Pras, J. Schonwalder, M. Burgess, O. Festor, G. M. Perez, R. Stadler, and B. Stiller, “Key research challenges in network management,” *IEEE Communications Magazine*, vol. 45, no. 10, pp. 104–110, Oct. 2007.
- [241] H. Urunov, S. Y. Shin, S. H. Park, and K. Yi, “U-SNMP for the internet of underwater things,” *International Journal of Control and Automation*, vol. 10, pp. 199–216, Jan. 2017.
- [242] A. K. Y. Wong, An Chi Chen, N. Paramesh, and P. Rav, “Ontology mapping for network management systems,” in *Proceedings Network Operations and Management Symposium*, vol. 1. Seoul, South Korea: IEEE/IFIP, Apr. 2004, pp. 885–886.
- [243] H. Mukhtar, K. Kang-Myo, S. A. Chaudhry, A. H. Akbar, K. Ki-Hyung, and S.-W. Yoo, “LNMP- management architecture for IPv6 based low-power wireless personal area networks (6LoWPAN),” in *Proceedings Network Operations and Management Symposium*. Salvador, Brazil: IEEE, Aug. 2008, pp. 417–424.
- [244] S. Sharma, D. Staessens, D. Colle, D. Palma, J. Gonçalves, R. Figueiredo, D. Morris, M. Pickavet, and P. Demeester, “Implementing quality of service for the software defined networking enabled future internet,” in *Proceedings 3rd European Workshop on Software Defined Networks*. London, UK: IEEE, Sep. 2014, pp. 49–54.
- [245] J. Bitto, R. Bahr, J. G. Hester, S. A. Nauroze, A. Georgiadis, and M. M. Tentzeris, “A novel solar and electromagnetic energy harvesting system with a 3-D printed package for energy efficient internet-of-things wireless sensors,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 5, pp. 1831–1842, May 2017.
- [246] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, “A robust, adaptive, solar-powered WSN framework for aquatic environmental monitoring,” *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45–55, Jan. 2011.
- [247] K. B. Joshi, J. H. Costello, and S. Priya, “Estimation of solar energy harvested for autonomous jellyfish vehicles (AJVs),” *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 539–551, Oct. 2011.
- [248] P. P. Jenkins, S. Messenger, K. M. Trautz, S. I. Maximenko, D. Goldstein, D. Scheiman, R. Hoheisel, and R. J. Walters, “High-bandgap solar cells for underwater photovoltaic applications,” *IEEE Journal of Photovoltaics*, vol. 4, no. 1, pp. 202–207, Jan. 2014.

- [249] D. M. Toma, J. del Rio, M. Carbonell-Ventura, and J. M. Masalles, “Underwater energy harvesting system based on plucked-driven piezoelectrics,” in *Proceedings OCEANS*. Genova, Italy: IEEE, May 2015.
- [250] F. U. Qureshi, A. Muhtaroglu, and K. Tuncay, “Near-optimal design of scalable energy harvester for underwater pipeline monitoring applications with consideration of impact to pipeline performance,” *IEEE Sensors Journal*, vol. 17, no. 7, pp. 1981–1991, Apr. 2017.
- [251] A. Bereketli and S. Bilgen, “Remotely powered underwater acoustic sensor networks,” *IEEE Sensors Journal*, vol. 12, no. 12, pp. 3467–3472, Dec. 2012.
- [252] T. Kan, Y. Zhang, Z. Yan, P. P. Mercier, and C. C. Mi, “A rotation-resilient wireless charging system for lightweight autonomous underwater vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 6935–6942, Aug. 2018.
- [253] L. Jing, C. He, J. Huang, and Z. Ding, “Energy management and power allocation for underwater acoustic sensor network,” *IEEE Sensors Journal*, vol. 17, no. 19, pp. 6451–6462, Oct. 2017.
- [254] Z. Wu, J. Yu, J. Yuan, M. Tan, and S. Qi, “Gliding motion regulation of a robotic dolphin based on a controllable fluke,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 2945–2953, Apr. 2020.
- [255] J. D. Hem, *Water analysis. inorganic species*. Academic Press, 2012, vol. 1, ch. Conductance: a collective measure of dissolved ions, pp. 137–161.
- [256] M. T. Islam, M. N. Rahman, M. S. J. Singh, and M. Samsuzzaman, “Detection of salt and sugar contents in water on the basis of dielectric properties using microstrip antenna-based sensor,” *IEEE Access*, vol. 6, pp. 4118–4126, Jan. 2018.
- [257] E. Vorathin, Z. M. Hafizi, A. M. Aizzuddin, M. K. A. Zaini, and K. S. Lim, “A novel temperature-insensitive hydrostatic liquid-level sensor using chirped FBG,” *IEEE Sensors Journal*, vol. 19, no. 1, pp. 157–162, Jan. 2019.
- [258] Y. Wang, S. M. S. M. Rajib, C. Collins, and B. Grieve, “Low-cost turbidity sensor for low-power wireless monitoring of fresh-water courses,” *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4689–4696, Jun. 2018.
- [259] W. D. Kirkey, J. S. Bonner, and C. B. Fuller, “Low-cost submersible turbidity sensors using low-frequency source light modulation,” *IEEE Sensors Journal*, vol. 18, no. 22, pp. 9151–9162, Nov. 2018.

-
- [260] C. Liu, Y. Li, and M. Xu, “An integrated detection and location model for leakages in liquid pipelines,” *Journal of Petroleum Science and Engineering*, vol. 175, pp. 852–867, Apr. 2019.
- [261] P. Wang, X. Tian, T. Peng, and Y. Luo, “A review of the state-of-the-art developments in the field monitoring of offshore structures,” *Ocean Engineering*, vol. 147, pp. 148–164, Jan. 2018.
- [262] X. Zhou, Q. Yu, and W. Peng, “Fiber-optic Fabry–Perot pressure sensor for down-hole application,” *Optics and Lasers in Engineering*, vol. 121, pp. 289–299, Oct. 2019.
- [263] F. Berlinger, J. Dusek, M. Gauci, and R. Nagpal, “Robust maneuverability of a miniature, low-cost underwater robot using multiple fin actuation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 140–147, Jan. 2018.
- [264] E. A. Makled, A. Yadav, O. A. Dobre, and R. D. Haynes, “Hierarchical full-duplex underwater acoustic network: a NOMA approach,” in *Proceedings OCEANS*. Charleston, USA: IEEE, Oct. 2018.
- [265] A. Celik, N. Saeed, B. Shihada, T. Y. Al-Naffouri, and M. Alouini, “A software-defined opto-acoustic network architecture for internet of underwater things,” *IEEE Communications Magazine*, vol. 58, no. 4, pp. 88–94, Apr. 2020.
- [266] Z. M. Fadlullah, F. Tang, B. Mao, J. Liu, and N. Kato, “On intelligent traffic control for large-scale heterogeneous networks: a value matrix-based deep learning approach,” *IEEE Communications Letters*, vol. 22, no. 12, pp. 2479–2482, Dec. 2018.
- [267] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, “A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, Jan. 2019.
- [268] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks,” *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, Aug. 2018.
- [269] Y. Su, L. Guo, Z. Jin, and X. Fu, “A mobile-beacon based iterative localization mechanism in large-scale underwater acoustic sensor networks,” *IEEE Internet of Things Journal*, Sep. 2020.

- [270] Y. Li, S. Wang, C. Jin, Y. Zhang, and T. Jiang, "A survey of underwater magnetic induction communications: fundamental issues, recent advances, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2466–2487, Feb. 2019.
- [271] A. Atyabi, S. MahmoudZadeh, and S. Nefti-Meziani, "Current advancements on autonomous mission planning and management systems: an AUV and UAV perspective," *Annual Reviews in Control*, vol. 46, pp. 196–215, Aug. 2018.
- [272] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, Apr. 2019.
- [273] H. Wu, J. Chen, X. Liu, Y. Xiao, M. Wang, Y. Zheng, and Y. Rao, "One-dimensional CNN-based intelligent recognition of vibrations in pipeline monitoring with DAS," *Journal of Lightwave Technology*, vol. 37, no. 17, pp. 4359–4366, Sep. 2019.
- [274] K. Ma, H. Leung, E. Jalilian, and D. Huang, "Fiber-optic acoustic-based disturbance prediction in pipelines using deep learning," *IEEE Sensors Letters*, vol. 1, no. 6, Dec. 2017.
- [275] J. Pan, Y. Yin, J. Xiong, W. Luo, G. Gui, and H. Sari, "Deep learning-based unmanned surveillance systems for observing water levels," *IEEE Access*, vol. 6, pp. 73 561–73 571, Nov. 2018.
- [276] M. R. Hedayati, A. A. Amidian, S. A. Sadr, and A. Razazan, "Intelligent ship hull inspection and NDT using ROV based flux leakage expert system," in *Proceedings 2nd International Conference on Computational Intelligence, Modelling and Simulation*. Tuban, Indonesia: IEEE, Sep. 2010, pp. 412–415.
- [277] C. Li, X. Peng, J. Liu, C. Wang, S. Fan, and S. Cao, "D-shaped fiber bragg grating ultrasonic hydrophone with enhanced sensitivity and bandwidth," *Journal of Lightwave Technology*, vol. 37, no. 9, pp. 2100–2108, May 2019.
- [278] X. E. Gros, P. Strachan, and D. Lowden, "Fusion of multiprobe NDT data for ROV inspection," in *Proceedings OCEANS Challenges of Our Changing Global Environment*, vol. 3. San Diego, USA: IEEE, Oct. 1995, pp. 2046–2050.
- [279] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1587–1597, May 2018.

-
- [280] N. Li, Z. Zheng, S. Zhang, Z. Yu, H. Zheng, and B. Zheng, “The synthesis of unpaired underwater images using a multistyle generative adversarial network,” *IEEE Access*, vol. 6, pp. 54 241–54 257, Sep. 2018.
- [281] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, “WaterGAN: unsupervised generative network to enable real-time color correction of monocular underwater images,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 387–394, Jan. 2018.
- [282] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, “SensorNet: a scalable and low-power deep convolutional neural network for multimodal data classification,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274–287, Jan. 2019.
- [283] J. Chang and J. Sha, “Prune deep neural networks with the modified $L_{1/2}$ penalty,” *IEEE Access*, vol. 7, pp. 2273–2280, Dec. 2018.
- [284] Y. Tian, B. Liu, X. Su, L. Wang, and K. Li, “Underwater imaging based on LF and polarization,” *IEEE Photonics Journal*, vol. 11, no. 1, Feb. 2019.
- [285] D. G. Zawada, “Image processing of underwater multispectral imagery,” *IEEE Journal of Oceanic Engineering*, vol. 28, no. 4, pp. 583–594, Oct. 2003.
- [286] S. Suresh, E. Westman, and M. Kaess, “Through-water stereo SLAM with refraction correction for AUV localization,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 692–699, Apr. 2019.
- [287] G. Aparna, S. D’Souza, and B. N. Arjun, “Prediction of daily sea surface temperature using artificial neural networks,” *International Journal of Remote Sensing*, vol. 39, pp. 4214–4231, Mar. 2018.
- [288] N. Bounceur, I. Hoteit, and O. Knio, “A Bayesian structural time series approach for predicting Red Sea temperatures,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1996–2009, Apr. 2020.
- [289] X. Meng and J. Cheng, “Estimating land and sea surface temperature from cross-calibrated Chinese Gaofen-5 thermal infrared data using split-window algorithm,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 3, pp. 509–513, Mar. 2020.
- [290] K. Patil and M. Deo, “Prediction of daily sea surface temperature using efficient neural networks,” *Ocean Dynamics*, vol. 67, Feb. 2017.

- [291] K. Iqbal, M. Zhang, S. Piao, and G. He, “Gradual but persistent quest for the ocean observation by employing multifarious sensing gadgets: a preview,” in *Proceedings OCEANS*. Marseille, France: IEEE, Jun. 2019.
- [292] Y. Xu *et al.*, “Contribution of SST change to multidecadal global and continental surface air temperature trends between 1910 and 2013,” *Climate Dynamics*, vol. 54, no. 3, pp. 1295–1313, Feb. 2020.
- [293] S. Wolff, F. O’Donncha, and B. Chen, “Statistical and machine learning ensemble modelling to forecast sea surface temperature,” *Journal of Marine Systems*, vol. 208, pp. 103 347–103 359, Aug. 2020.
- [294] K. Bandara, C. Bergmeir, and H. Hewamalage, “LSTM-MSNet: leveraging forecasts on sets of related time series with multiple seasonal patterns,” *IEEE Transactions on Neural Networks and Learning Systems*, Apr. 2020.
- [295] Y. Lan, Y. Hao, K. Xia, B. Qian, and C. Li, “Stacked residual recurrent neural networks with cross-layer attention for text classification,” *IEEE Access*, vol. 8, pp. 70 401–70 410, Apr. 2020.
- [296] A. Yazdizadeh, Z. Patterson, and B. Farooq, “Ensemble convolutional neural networks for mode inference in smartphone travel survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2232–2239, Jun. 2020.
- [297] N. R. Nalli, A. Gambacorta, Q. Liu, C. D. Barnet, C. Tan, F. Iturbide-Sanchez, T. Reale, B. Sun, M. Wilson, L. Borg, and V. R. Morris, “Validation of atmospheric profile retrievals from the SNPP NOAA-unique combined atmospheric processing system. part 1: temperature and moisture,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 180–190, Jan. 2018.
- [298] R. Vaz, V. Shah, A. Sawhney, and R. Deolekar, “Automated big-O analysis of algorithms,” in *Proceedings International Conference on Nascent Technologies in Engineering (ICNTE)*. Mumbai, India: IEEE, Jan. 2017.
- [299] A. D. L. Steven, M. E. Baird, R. Brinkman, N. J. Car, S. J. Cox, M. Herzfeld, J. Hodge, E. Jones, E. King, and N. Margvelashvili, “eReefs: an operational information system for managing the Great Barrier Reef,” *Journal of Operational Oceanography*, vol. 12, no. S2, pp. 12–28, Aug. 2019.
- [300] J. Waterhouse, J. Brodie, D. Tracey, R. Smith, M. VanderGragt, C. Collier, C. Petus, M. Baird, F. Kroon, and R. Mann, “Land use impacts on the Great Barrier Reef

- water quality and ecosystem condition,” *The Scientific Consensus Statement, State of Queensland*, Nov. 2020.
- [301] Reef-2050, “Australian and Queensland government: The Reef 2050 plan,” www.environment.gov.au, Mar. 2021.
- [302] R. W. Howarth and R. Marino, “Nitrogen as the limiting nutrient for eutrophication in coastal marine ecosystems: evolving views over three decades,” *Limnology and oceanography*, vol. 51, no. 1, pp. 364–376, Jan. 2006.
- [303] V. H. Smith, G. D. Tilman, and J. C. Nekola, “Eutrophication: impacts of excess nutrient inputs on freshwater, marine, and terrestrial ecosystems,” *Environmental pollution*, vol. 100, no. 1-3, pp. 179–196, 1999.
- [304] F. J. Kroon, P. M. Kuhnert, B. L. Henderson, S. N. Wilkinson, A. Kinsey-Henderson, B. Abbott, J. E. Brodie, and R. D. R. Turner, “River loads of suspended solids, nitrogen, phosphorus and herbicides delivered to the Great Barrier Reef lagoon,” *Marine pollution bulletin*, vol. 65, no. 4-9, pp. 167–181, 2012.
- [305] P. R. F. Bell, I. Elmetri, and B. E. Lapointe, “Evidence of large-scale chronic eutrophication in the Great Barrier Reef: quantification of chlorophyll a thresholds for sustaining coral reef communities,” *Ambio*, vol. 43, no. 3, pp. 361–376, 2014.
- [306] G. L. McCloskey, R. Baheerathan, C. Dougall, R. Ellis, F. R. Bennett, D. Waters, S. Darr, B. Fentie, L. R. Hateley, and M. Askildsen, “Modelled estimates of fine sediment and particulate nutrients delivered from the Great Barrier Reef catchments,” *Marine pollution bulletin*, vol. 165, pp. 112 163–112 201, Apr. 2021.
- [307] G. De’ath and K. Fabricius, “Water quality as a regional driver of coral biodiversity and macroalgae on the Great Barrier Reef,” *Ecological Applications*, vol. 20, no. 3, pp. 840–850, Apr. 2010.
- [308] M. A. MacNeil, C. Mellin, S. Matthews, N. H. Wolff, T. R. McClanahan, M. Devlin, C. Drovandi, K. Mengersen, and N. A. J. Graham, “Water quality mediates resilience on the Great Barrier Reef,” *Nature Ecology & Evolution*, vol. 3, no. 4, pp. 620–627, Mar. 2019.
- [309] A. D. McKinnon, S. Duggan, M. Logan, and C. Lønborg, “Plankton respiration, production, and trophic state in tropical coastal and shelf waters adjacent to northern Australia,” *Frontiers in Marine Science*, vol. 4, pp. 346–359, Oct. 2017.

- [310] F. J. Kroon, P. Thorburn, B. Schaffelke, and S. Whitten, “Towards protecting the Great Barrier Reef from land-based pollution,” *Global change biology*, vol. 22, no. 6, pp. 1985–2002, Feb. 2016.
- [311] A. Coggan, P. Thorburn, S. Fielke, R. Hay, and J. C. R. Smart, “Motivators and barriers to adoption of improved land management practices: a focus on practice change for water quality improvement in Great Barrier Reef catchments,” *Marine Pollution Bulletin*, vol. 170, pp. 112 628–112 639, Sep. 2021.
- [312] N. J. Waltham, C. Wegscheidl, A. Volders, J. C. R. Smart, S. Hasan, E. Lédée, and J. Waterhouse, “Land use conversion to improve water quality in high DIN risk, low-lying sugarcane areas of the Great Barrier Reef catchments,” *Marine Pollution Bulletin*, vol. 167, pp. 112 373–112 386, Jun. 2021.
- [313] M. Najafzadeh, A. Ghaemi, and S. Emamgholizadeh, “Prediction of water quality parameters using evolutionary computing-based formulations,” *International Journal of Environmental Science and Technology*, vol. 16, no. 10, pp. 6377–6396, Jan. 2019.
- [314] M. Najafzadeh and S. Niazmardi, “A novel multiple-kernel support vector regression algorithm for estimation of water quality parameters,” *Natural Resources Research*, Jun. 2021.
- [315] M. Najafzadeh, F. Homaei, and H. Farhadi, “Reliability assessment of water quality index based on guidelines of national sanitation foundation in natural streams: integration of remote sensing and data-driven models,” *Artificial Intelligence Review*, Apr. 2021.
- [316] A. Sharifi, “Using Sentinel-2 data to predict nitrogen uptake in maize crop,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2656–2662, Jun. 2020.
- [317] J. Wen, J. Yang, B. Jiang, H. Song, and H. Wang, “Big data driven marine environment information forecasting: a time series prediction network,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 4–18, Jan. 2021.
- [318] Y. Zhou, H. Dong, and A. E. Saddik, “Deep learning in next-frame prediction: a benchmark review,” *IEEE Access*, vol. 8, pp. 69 273–69 283, Apr. 2020.
- [319] W. Huang, W. Ma, X. Liu, W. Peng, and J. Zhang, “Numerical study of hydrodynamics and water quality in Qinhuangdao coastal waters, China: implication for

-
- pollutant loadings management,” *Environmental Modeling & Assessment*, vol. 26, no. 1, pp. 63–76, Jan. 2021.
- [320] U. Khan, F. J. Cook, R. Laugesen, M. M. Hasan, K. Plastow, G. E. Amirthanathan, M. A. Bari, and N. K. Tuteja, “Development of catchment water quality models within a realtime status and forecast system for the Great Barrier Reef,” *Environmental Modelling & Software*, vol. 132, pp. 104 790–104 809, Oct. 2020.
- [321] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, “Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics,” in *Proc. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Jun. 2019, pp. 9154–9162.
- [322] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” in *Proc. 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.
- [323] S. Hong, S. Kim, M. Joh, and S.-K. Song, “Psique: next sequence prediction of satellite images using a convolutional sequence-to-sequence network,” *Computing Research Repository*, vol. 1711.10644, Aug. 2018.
- [324] V. L. Guen and N. Thome, “Disentangling physical dynamics from unknown factors for unsupervised video prediction,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Virtual, Jun. 2020, pp. 11 474–11 484.
- [325] M. E. Baird, K. A. Wild-Allen, J. Parslow, M. Mongin, B. Robson, J. Skerratt, F. Rizwi, M. Soja-Woźniak, E. Jones, and M. Herzfeld, “CSIRO Environmental Modelling Suite (EMS): scientific description of the optical and biogeochemical models (vB3p0),” *Geoscientific Model Development*, vol. 13, pp. 4503–4553, Sep. 2020.
- [326] L. Ren and J. Yang, “Nitrogen nutrients cycling in marine environment and its modeling research,” *Advances in Earth Science*, vol. 15, no. 1, pp. 58–64, Jan. 2000.
- [327] B. J. Robson, “State of the art in modelling of phosphorus in aquatic systems: review, criticisms and commentary,” *Environmental Modelling & Software*, vol. 61, pp. 339–359, Nov. 2014.
- [328] C. Lindemann, D. L. Aksnes, K. J. Flynn, and S. Menden-Deuer, “Modeling the plankton—enhancing the integration of biological knowledge and mechanistic understanding,” *Frontiers in Marine Science*, vol. 4, pp. 358–366, Nov. 2017.

- [329] A. N. Ahmed, F. B. Othman, H. A. Afan, R. K. Ibrahim, C. M. Fai, M. S. Hossain, M. Ehteram, and A. Elshafie, “Machine learning methods for better water quality prediction,” *Journal of Hydrology*, vol. 578, pp. 124 084–124 102, Nov. 2019.
- [330] A. H. Haghiabi, A. H. Nasrolahi, and A. Parsaie, “Water quality prediction using machine learning methods,” *Water Quality Research Journal*, vol. 53, no. 1, pp. 3–13, Jan. 2018.
- [331] N. Margvelashvili, J. Andrewartha, M. Herzfeld, B. J. Robson, and V. E. Brando, “Satellite data assimilation and estimation of a 3D coastal sediment transport model using error-subspace emulators,” *Environmental Modelling & Software*, vol. 40, pp. 191–201, 2013.
- [332] J. H. Skerratt, M. Mongin, M. E. Baird, K. A. Wild-Allen, B. J. Robson, B. Schaf-felke, C. H. Davies, A. J. Richardson, N. Margvelashvili, M. Soja-Wozniak, and A. D. L. Steven, “Simulated nutrient and plankton dynamics in the Great Barrier Reef (2011–2016),” *Journal of Marine Systems*, vol. 192, pp. 51–74, Apr. 2019.
- [333] R. Huang, M. Lu, A. Peyton, and W. Yin, “A novel perturbed matrix inversion based method for the acceleration of finite element analysis in crack-scanning eddy current NDT,” *IEEE Access*, vol. 8, pp. 12 438–12 444, Jan. 2020.
- [334] Q. Zhu, Z. Liu, and J. Yan, “Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks,” *Computational Mechanics*, vol. 67, pp. 619–635, Jan. 2021.
- [335] AIMS-MMP, “Australian institute of marine science: water quality particulate and dissolved nutrient data,” www.aims.gov.au, Mar. 2021.
- [336] NCI, “eReefs GBR4 BGC and sediments model data by Australian national compu-tational infrastructure,” www.nci.org.au, Mar. 2021.
- [337] AIMS-eReefs, “A mirror eReefs model data, with custom aggregations performed by the eAtlas team,” www.aims.gov.au, Mar. 2021.
- [338] F. Saberi-Movahed, M. Najafzadeh, and A. Mehrpooya, “Receiving more accurate predictions for longitudinal dispersion coefficients in water pipelines: training group method of data handling using extreme learning machine conceptions,” *Water Re-sources Management*, vol. 34, no. 2, pp. 529–561, Jan. 2020.
- [339] L. Du, “How much deep learning does neural style transfer really need? an abla-tion study,” in *Proceedings Winter Conference on Applications of Computer Vision (WACV)*. Snowmass, USA: IEEE, Mar. 2020, pp. 3139–3148.

-
- [340] M. M. Magno-Canto, L. I. W. McKinna, B. J. Robson, and K. E. Fabricius, “Model for deriving benthic irradiance in the Great Barrier Reef from MODIS satellite imagery,” *Optics Express*, vol. 27, no. 20, pp. 1350–1371, Sep. 2019.
- [341] C. Petus, J. Waterhouse, S. Lewis, M. Vacher, D. Tracey, and M. Devlin, “A flood of information: using Sentinel-3 water colour products to assure continuity in the monitoring of water quality trends in the Great Barrier Reef (Australia),” *Journal of environmental management*, vol. 248, pp. 109 255–109 275, Jul. 2019.
- [342] R. K. Sarangi, “Remote-sensing-based estimation of surface nitrate and its variability in the southern peninsular Indian waters,” *International Journal of Oceanography*, vol. 2011, Jan. 2012.
- [343] D. Wang, Q. Cui, F. Gong, L. Wang, X. He, and Y. Bai, “Satellite retrieval of surface water nutrients in the coastal regions of the east China Sea,” *Remote Sensing*, vol. 10, no. 12, Nov. 2018.
- [344] NOAA, “United States National Oceanic and Atmospheric Administration: Coral reef ecosystems,” www.noaa.gov, May 2021.
- [345] M. E. Baird, M. Mongin, J. Skerratt, N. Margvelashvili, S. Tickell, A. D. L. Steven, C. Robillot, R. Ellis, D. Waters, P. Kaniewska, and J. Brodie, “Impact of catchment-derived nutrients and sediments on marine water quality on the Great Barrier Reef: an application of the eReefs marine modelling system,” *Marine Pollution Bulletin*, vol. 167, pp. 112 297–112 313, Jun. 2021.
- [346] N. Margvelashvili, J. Andrewartha, M. Baird, M. Herzfeld, E. Jones, M. Mongin, F. Rizwi, B. J. Robson, J. Skerratt, K. Wild-Allen, and A. Steven, “Simulated fate of catchment-derived sediment on the Great Barrier Reef shelf,” *Marine Pollution Bulletin*, vol. 135, pp. 954–962, Oct. 2018.
- [347] L. A. McKergow, I. P. Prosser, A. O. Hughes, and J. Brodie, “Sources of sediment to the Great Barrier Reef world heritage area,” *Marine Pollution Bulletin*, vol. 51, no. 1, pp. 200–211, Jan. 2005.
- [348] B. Bhattacharya and D. P. Solomatine, “Machine learning in sedimentation modelling,” *Neural Networks*, vol. 19, no. 2, pp. 208–214, Mar. 2006.
- [349] R. Reichelt *et al.*, *Water quality guidelines for the Great Barrier Reef marine park*, 2nd ed. Great Barrier Reef Marine Park Authority, 2010.

- [350] N. Y. Margvelashvili, M. Herzfeld, F. Rizwi, M. Mongin, M. E. Baird, E. Jones, B. Schaffelke, E. King, and T. Schroeder, “Emulator-assisted data assimilation in complex models,” *Ocean Dynamics*, vol. 66, pp. 1109–1124, Aug. 2016.
- [351] S. K. Bhagat, T. Tiyasha, S. M. Awadh, T. M. Tung, A. H. Jawad, and Z. M. Yaseen, “Prediction of sediment heavy metal at the Australian Bays using newly developed hybrid artificial intelligence models,” *Environmental Pollution*, vol. 268, pp. 115 663–115 676, Jan. 2021.
- [352] Y. Mehri, M. Nasrabadi, and M. H. Omid, “Prediction of suspended sediment distributions using data mining algorithms,” *Ain Shams Engineering Journal*, vol. 1, May 2021.
- [353] M. Y. Ali-Khan, F. Tian, F. Hasan, and G. J. Chakrapani, “Artificial neural network simulation for prediction of suspended sediment concentration in the River Ramganga, Ganges Basin, India,” *International Journal of Sediment Research*, vol. 34, no. 2, pp. 95–107, Apr. 2019.
- [354] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv*, 2017.
- [355] G. C. Sergio and M. Lee, “Stacked DeBERT: All attention in incomplete data for text classification,” *Neural Networks*, vol. 136, pp. 87–96, Apr. 2021.
- [356] A. Cossu, A. Carta, V. Lomonaco, and D. Bacciu, “Continual learning for recurrent neural networks: An empirical evaluation,” *Neural Networks*, vol. 143, pp. 607–627, Nov. 2021.
- [357] X. Su, G. Wang, and Q. Li, “Prediction method for transformer state based on GRU network,” in *Proc. Industrial and Commercial Power System Asia*, Weihai, China, Sep. 2020, pp. 1751–1755.
- [358] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv*, 2014.
- [359] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, “A comparison of transformer and LSTM encoder decoder models for ASR,” in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, Singapore, Feb. 2020, pp. 8–15.
- [360] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An

- image is worth 16x16 words: transformers for image recognition at scale,” *arXiv*, Oct. 2020.
- [361] K. S. Surana and J. N. Reddy, *The finite element method for boundary value problems: mathematics and computations*, 1st ed. CRC press, 2017.
- [362] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.
- [363] S. Gupta, W. Zhang, and F. Wang, “Model accuracy and runtime tradeoff in distributed deep learning: a systematic study,” in *Proc. 16th International Conference on Data Mining (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 171–180.
- [364] M. Chen, D. Gunduz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, “Distributed learning in wireless networks: recent progress and future challenges,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.
- [365] X. Zhong, F. Ji, F. Chen, Q. Guan, and H. Yu, “A new acoustic channel interference model for 3-D underwater acoustic sensor networks and throughput analysis,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9930–9942, Apr. 2020.
- [366] B. Jedari, G. Premsankar, G. Illahi, M. D. Francesco, A. Mehrabi, and A. Ylä-Jääski, “Video caching, analytics, and delivery at the wireless edge: a survey and future directions,” *IEEE Communications Surveys and Tutorials*, vol. 23, no. 1, pp. 431–471, Jan. 2021.
- [367] L. Chen, Z. Jiang, L. Tong, Z. Liu, A. Zhao, Q. Zhang, J. Dong, and H. Zhou, “Perceptual underwater image enhancement with deep learning and physical priors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3078–3092, Aug. 2021.
- [368] X. Qin, X. Luo, Z. Wu, and J. Shang, “Optimizing the sediment classification of small side-scan sonar images based on deep learning,” *IEEE Access*, vol. 9, pp. 29 416–29 428, Jan. 2021.
- [369] I. H. Laradji, A. Saleh, P. Rodriguez, D. Nowrouzezahrai, M. R. Azghadi, and D. Vazquez, “Weakly supervised underwater fish segmentation using affinity lcfcn,” *Scientific Reports*, vol. 11, no. 1, 2021.

- [370] J. E. Almanza-Medina, B. Henson, and Y. V. Zakharov, "Deep learning architectures for navigation using forward looking sonar images," *IEEE Access*, vol. 9, pp. 33 880–33 896, Feb. 2021.
- [371] L. Wang, X. Ye, H. Xing, Z. Wang, and P. Li, "YOLO Nano Underwater: a fast and compact object detector for embedded device," in *Proc. Global Oceans*, Biloxi, MS, USA, Oct. 2020.
- [372] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Feb. 2021.
- [373] TensorFlow Tutorials, "Image segmentation," www.tensorflow.org, Aug. 2021.
- [374] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Berlin, Germany, 2015, pp. 234–241.
- [375] A. Saleh, M. Sheaves, and M. Rahimi Azghadi, "Computer vision and deep learning for fish classification in underwater habitats: A survey," *arXiv:2203.06951*, 2022.
- [376] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1125–1134.
- [377] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.
- [378] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Scientific Reports*, vol. 14671, no. 10, Sep. 2020.
- [379] S. Lee, H. Kim, J. Park, J. Jang, C.-S. Jeong, and S. Yoon, "TensorLightning: a traffic-efficient distributed deep learning on commodity Spark clusters," *IEEE Access*, vol. 6, pp. 27 671–27 680, May 2018.
- [380] D. Lunga, J. Gerrand, L. Yang, C. Layton, and R. Stewart, "Apache Spark accelerated deep learning inference for large scale satellite image analytics," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 271–283, Jan. 2020.

-
- [381] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.
- [382] H. Ping, H. Aiping, and T. Linwei, “Deep learning-based decision-making model for the submarine evade movement,” in *Proc. OCEANS*, San Diego, CA, USA, Sep. 2021.
- [383] M. Farzadkhoo, R. T. Kingsford, I. M. Suthers, P. Geelan-Small, J. H. Harris, W. Peirson, and S. Felder, “Attracting juvenile fish into tube fishways – roles of transfer chamber diameter and flow velocity,” *Ecological Engineering*, vol. 176, pp. 106 544–106 559, Mar. 2022.
- [384] V. B. Roman, G. A. E. Banos, C. A. Q. Solis, M. I. F. Banuelos, M. Rivero, and M. A. E. Soberanis, “Comparative study on the cost of hybrid energy and energy storage systems in remote rural communities near Yucatan, Mexico,” *Applied Energy*, vol. 308, pp. 118 334–118 345, Feb. 2022.
- [385] A. Roberts, B. Thomas, P. Sewell, Z. Khan, S. Balmain, and J. Gillman, “Current tidal power technologies and their suitability for applications in coastal and marine areas,” *Ocean Engineering and Marine Energy*, vol. 2, pp. 227–245, Jan. 2016.
- [386] E. Ackerman, “Underwater manta kites for tidal power harvesting,” spectrum.ieee.org, Apr. 2021.
- [387] X. Sun, Y. Zhang, K. Shi, Y. Zhang, N. Li, W. Wang, X. Huang, and B. Qin, “Monitoring water quality using proximal remote sensing technology,” *Science of The Total Environment*, vol. 803, pp. 149 805–149 817, Jan. 2022.