Upscaling Fog Computing in Oceans for Underwater Pervasive Data Science using Low-Cost Micro-Clouds

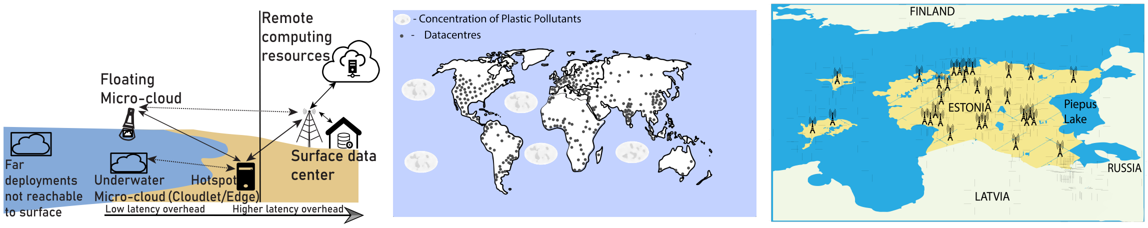
FAROOQ DAR, University of Tartu, Estonia
MOHAN LIYANAGE, University of Tartu, Estonia
MARKO RADETA, Wave Labs, MARE/ARNET, ARDITI, University of Madeira, University of Belgrade, Portugal
ZHIGANG YIN, University of Tartu, Estonia
AGUSTIN ZUNIGA, University of Helsinki, Finland
SOKOL KOSTA, Aalborg University, Denmark
SASU TARKOMA, University of Helsinki, Finland
PETTERI NURMI, University of Helsinki, Finland
HUBER FLORES, University of Tartu, Estonia

Underwater environments are emerging as a new frontier for data science thanks to an increase in deployments of underwater sensor technology. Challenges in operating computing underwater combined with a lack of high-speed communication technology covering most aquatic areas mean that there is a significant delay between the collection and analysis of data. This in turn limits the scale and complexity of the applications that can operate based on these data. In this paper, we develop underwater fog computing support using low-cost micro-clouds and demonstrate how they can be used to deliver cost-effective support for data-heavy underwater applications. We develop a proof-of-concept micro-cloud prototype and use it to perform extensive benchmarks that evaluate the suitability of underwater micro-clouds for diverse underwater data science scenarios. We conduct rigorous tests in both controlled and field deployments, using river and sea waters. We also address technical challenges in enabling underwater fogs, evaluating the performance of different communication interfaces and demonstrating how accelerometers can be used to detect the likelihood of communication failures and determine which communication interface to use. Our work offers a cost-effective way to increase the scale and complexity of underwater data science applications, and demonstrates how off-the-shelf devices can be adopted for this purpose.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Cloudlets, Edge computing, Cloud computing, Aquatic environments, Computation offloading

ACM Reference Format:
Farooq Dar, Mohan Liyanage, Marko Radeta, Zhigang Yin, Agustin Zuniga, Sokol Kosta, Sasu Tarkoma, Petteri Nurmi, and Huber Flores. 2022. Upscaling Fog Computing in Oceans for Underwater Pervasive Data Science using Low-Cost Micro-Clouds. In *ACM Transactions on Internet of Things, 2018, New York, NY*. ACM, New York, NY, USA, 29 pages. https://doi.org/10.0000/0000000.123456



(a) (b) (c)

Fig. 1. Complexity of moving underwater collected data to computing infrastructure. (a) Surface-based deployments can only support underwater infrastructure that is close to coastal areas, (b) Major areas of marine litter are far from data centers and also in areas that are far from coastal regions, (c) Inland regions, such as lakes, rivers and streams, also can be poorly covered by communication infrastructure.

1 INTRODUCTION

Underwater environments are slowly emerging as the new frontier for data science. Indeed, underwater sensors ranging from hydrophones [53] to video cameras [54] and sensors measuring salinity, pH or other water characteristics are being increasingly deployed. Data from these deployments can then used to support a variety of underwater applications, such as oil pipeline monitoring [2, 33], fishery management [57, 64], reef and fish school estimation [10, 31], and harbour safety monitoring [45]. Challenges in operating computing underwater combined with a lack of high-speed communication technology result in limited computing infrastructure being available to the devices that produce the data. This results in significant delays between the collection and analysis of data, which in turn limits the scope and scale of applications that can take advantage of these deployments and the data they produce [25, 56]. Overcoming this limitation requires providing access to computing resources close to the data sources. Indeed, this is essential for the adoption of applications that increase the awareness of the underwater contexts. Examples of these applications range from increasing coverage of underwater pollution [15], forecasting for litter navigation and area growth [50], and real-time analysis of the impact of pollution in marine species [63]. Having computing resources close to the data sources can also facilitate the design of communication infrastructure, e.g., by using direct device-to-device underwater connectivity instead of relying on underwater-to-surface and surface-to-cloud connectivity.

Currently, the main approach for augmenting the computational resources of underwater sensors is to rely on surface-based infrastructure, such as ships or buoys, which can offer computing infrastructure or act as gateways to land-based infrastructure. The key limitations of this approach are that it only supports limited depths and distances from land-based infrastructure and that it requires specialized communication interfaces, such as laser-based optical communication [13, 62, 66], to relay the data from the underwater sensors to the surface. Using surface-based infrastructure to relay data to remote infrastructure is only feasible in coastal areas that are near populated areas as high-speed and bandwidth communications require base stations to be sufficiently close to the gateways – around 10km or at most few tens of kilometers (Fig. 1a). This means that most areas of interest cannot benefit from this approach, e.g., areas with heavy marine litter concentrations are far from computing and networking support (Fig. 1b), and even inland areas, such as rivers, lakes and streams often lack access to suitable computing infrastructure (Fig. 1c). While there have been recent efforts to bring data-centers closer to marine areas [17], these are also likely to target coastal areas located close to densely populated urban areas rather than become a widely adopted solution for underwater data processing.

We contribute a *fog computing*[1] solution for underwater data science that relies on submerged commercial-off-the-shelf (COTS) devices to deliver cost-effective and decentralized solutions to access computing and storage resources. We develop a proof-of-concept offloading framework and two prototypes that use COTS micro controllers (such as Raspberry PI) as the devices that deliver fog computing support for underwater applications (i.e., as fog nodes). By using inexpensive, small and energy-efficient COTS components, the fog can be integrated into AUVs, buoys, ships and other underwater infrastructure. This can be used to deliver on-demand support for processing and help to scale-up underwater data science. We focus on a micro-cloud architecture where multiple devices collaborate to provide the fog computing support as this is best suited for resource constrained devices and can be implemented cost-effectively [23, 39]. Our approach can even take advantage of standard communication interfaces, e.g., we demonstrate that Wi-Fi interfaces can be used to enable interactions with the micro-cloud as long as the distance from the devices is sufficiently small. Indeed, our solution offers comparable networking performance to established techniques, such as underwater LoRa [42], while being able to perform computations underwater. The range and bandwidth of communications can be further enhanced using advanced communication interfaces, e.g., advanced optical communications can deliver 10 Mbps data transmission with a range of 40 meters [36].

We demonstrate the feasibility and practical benefits of our solution through extensive benchmarks that involve both surface-based and underwater computing tasks in differing water conditions, including tests carried out during a recreational scuba dive where the solution is deployed on the seabed. We first evaluate the suitability to support diverse applications using tasks that are representative of the needs of underwater data science, while at the same time being part of established fog computing benchmarks [46]. We follow these experiments with tests conducted in underwater settings and focusing on object detection from camera footage, a common task in underwater data science that is relevant, e.g., for pollution detection, biodiversity estimation, and pipe leakage detection [56]. The results of our experiments demonstrate that micro-clouds can indeed provide general purpose support for a wide range of underwater computing tasks and that they are capable of operating even in complex underwater environments. Standard communication interfaces, such as Wi-Fi, are sufficient for maintaining connectivity within the devices forming the micro-cloud as long as the devices are within the same container. Clients can communicate with the micro-cloud as long as they are sufficiently close to the container (i.e., few centimeters), though specialized underwater communications technologies can be used to extend the range at which clients can send requests to the micro-cloud. Despite offering only short-range, the practical implication of these results is significant, suggesting that even relatively simple COTS underwater drones that lack dedicated communication interfaces could be used to support underwater data science by attaching a separate fog container into them that uses COTS technologies rather than having to integrate complex communication interfaces. In terms of computational performance, we find COTS devices, such as Raspberry PIs, to have sufficient computational power for most underwater data science needs, but the performance slightly drops as the depth of the container is increased. The drop is highest at shallow depths (due to heat accumulation inside the container) and the performance plateaus at deeper depths. We also demonstrate that optical communications are a good candidate for extending the communication range, but the stability of connectivity depends on the calmness of the water. Finally, we demonstrate that accelerometer-based motion analysis can be used to optimize the performance of the micro-cloud, e.g., by regulating resource usage and identifying conditions where communications are most likely to succeed.

[1]We define fog computing as decentralized support for computing and data processing that is offered close to the source of data. This definition is adapted from [79] and aligns with the original definition of fog computing by Cisco.

Summary of Contributions

- **Novel Underwater Fog Solution.** We develop a novel fog computing approach for underwater data science that uses low-cost micro clouds.
- **Scalable and Cost Effective Processing Support for Underwater Deployments.** As we exploit COTS devices to create fog nodes, our proposed solution can be deployed and replicated at a large scale with ease.
- **Novel Insights.** We perform rigorous benchmarks to assess the performance of micro-clouds in underwater environments, offering novel insights into how depth, turbidity and distance affect performance and communication of collaborative micro-clouds. Besides controlled experiments, we also provide a detailed analysis of micro-cloud deployment in the open sea.

2 FEASIBILITY EXPERIMENT

Our work targets the need for general purpose solutions that can support the processing needs of a wide range of underwater applications. Micro-clouds consisting of commodity devices are a promising solution for delivering such support as they are inexpensive, readily available and can support common processing tasks [40]. The inexpensiveness of the components lowers the potential economic consequences of equipment failures and allows for denser and faster deployments. Note that, unlike conventional scenarios for fog computing, underwater scenarios benefit even from low-end devices as the main requirement is to have dedicated computing support available. Indeed, underwater platforms typically have very limited computational units due to navigation and maneuvering being the main functions that need support. At the same time, computing support cannot be easily integrated onto these devices due to the need of having waterproofing for the hardware and software components [56]. We envision the micro-clouds to be deployed either by submerging them as separate components or by attaching them as modules to the device operating in underwater environments. For example, the micro-cloud could be placed as a separate container on top of an AUV that is responsible for collecting and analyzing underwater measurements [26] or operate as part of buoys or other observation stations [71].

Feasibility Experiment: We first conduct controlled benchmark experiments to explore the feasibility of deploying functional micro-clouds and analyzing the influence submerging has on their computing power and other resources. We built a micro-cloud from a Raspberry Pi 4 (RPi4) micro-computer that is encased into a waterproof glass container (see Section 4 for further details of the experimental setup). We use a Raspberry Pi because it is one of the most common IoT devices used for rapid prototyping - yet it cannot be used for underwater IoT by default. A single RPi4 has limited computing resources and thus alone is insufficient for large-scale applications, but collaborative processing can be used to enable more powerful processing by aggregating and interconnecting multiple such devices [3]. In practice the fog should be able to provide sufficient computing power to analyze the data that is captured from its vicinity. The most common type of data are images or videos, with other types of data including environmental parameters (e.g., salinity, temperature or pH). RPi4 is sufficient for processing this kind of data and even for running deep learning based object detection on the images [56]. For this reason, we first focus on benchmarking the individual processing capabilities of a RPi4. We developed a lightweight fog service on RPi4 that can be requested by users and that follows a client-server architecture. After submerging the micro-cloud in water, we then analyze the influence of water when connecting to the micro-cloud to use its computational resources. For these controlled benchmarks we consider only shallow depths (few centimeters) to limit the risk of equipment loss. Later on in the paper (Section 7) we consider more realistic operating environments where the devices are submerged to a depth of several meters.

Experimental Task and Setup: We perform the feasibility benchmarks using a computing task that provides a constant and uniform response time and incorporates resource-intensive processing. By controlling these properties of the task, we ensured that the controlled experiment is not influenced by non-deterministic computing behavior. The selected task corresponds to a primality test and search, i.e., finding all prime numbers within a given list of numbers. In the experiment, a client sends a request to the micro-cloud based fog service containing a list of 20 integer numbers within the interval of $100000 - 105000$. The service takes the request and identifies the prime numbers in the list, and sends the result back to the client. To analyze multiple clients sending requests to the service, we use JMeter[2] to simulate different workloads of users. We use an increasing workload from 100 to 500 users to analyze the capacity of the micro-cloud to handle the workload. We emulate different connectivity conditions by varying the depth at which the micro-cloud is being submerged. In these controlled experiments we rely on the standard Wi-Fi interface for communication between the devices.

Results: Figure 2 shows the performance results of the micro-cloud when handling different workloads of concurrent users. To measure the performance, we estimate the RTT (round trip time) from users completing requests successfully. We measure the performance of handling the workload on the surface (baseline) and underwater using the same setup. Figure 2a and d show the results outside the water (baseline). We then proceed to analyze the influence of water for handling different computational workloads. Figures 2b,c,e and f depict the results of handling workload underwater. From these results, we can observe that the micro-cloud can handle workload when it is submerged with minimal overhead when compared to the baseline (figure 2b and e). We found that in distances between $1 - 5$ cm from the surface (referred to as Depth-1 in the figure), the micro-cloud can complete every workload successfully. Between six and twelve centimeters (Depth-2 in the figure 2c and f), the transmissions start to be unreliable and the micro-cloud drops some requests. The loss in connectivity also affects performance, which is mostly due to the higher overhead caused by the Wi-Fi interface when packets are lost. Wi-Fi is well known to suffer from very poor underwater propagation [16], which implies that the client and micro-cloud must indeed be located within a few centimeters of each other. Note that this only concerns the communication from the client to the micro-cloud and internally the devices forming the micro-cloud can use Wi-Fi or other standard communication interfaces whenever they are deployed in the same container. Naturally integrating all components into the same container increases heat accumulation inside the container, but this is only an issue at shallow depths (around one meter or less) as at deeper depths the confounding effect of sunlight is decreased, water temperature is cooler, and increased water pressure outside the container also facilitates cooling. Another option for extending the range is to use acoustic or optical communication technology, which can reach depths of several meters [13, 62]. We explore one such solution in Section 6.

Lastly, we analyze the influence of sudden water motion when the micro-cloud is submerged. We inspect solely the first depth class as it corresponds to the case where the micro-cloud can be accessed without dropping requests. To analyze the influence of water motion, e.g., waves, currents and tides, for accessing computing resources of the micro-cloud, we place an accelerometer sensor floating in the water surface in a glass container while having the micro-cloud submerged. We then proceed to capture water motion on the surface while handling workload of users. Figure 3 depicts the results, indicating different types of water motion. In particular, Figure 3a and 3b depict situations where water surface has a low motion. During this type of motion, micro-clouds can complete computing workload of users smoothly. In contrast, Figure 3c and 3d indicate the behavior of a water surface induced suddenly by nearby water vehicles. When this occurs, the water surface depicts high motion and makes the connectivity with the micro-cloud

[2]https://jmeter.apache.org/

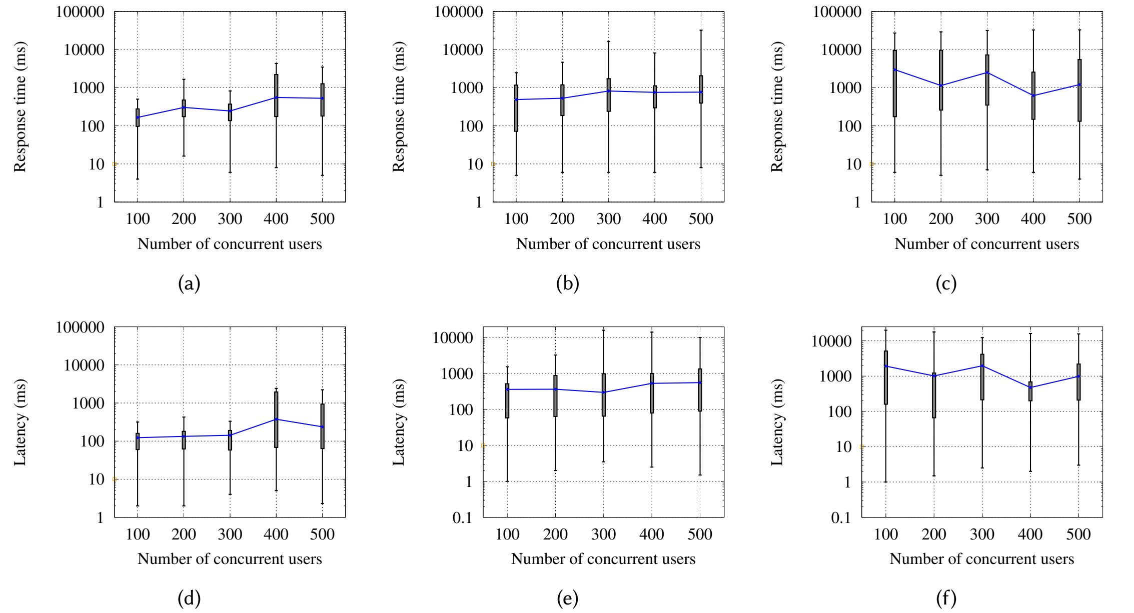


Fig. 2. Capacity results of submerged micro-clouds when handling multiple users, a) Response time (Baseline - no water), b) Depth-1, c) Depth-2, d) Latency (Baseline - no water), e) Depth-1, f) Depth-2.

unstable. Therefore, when high motion is experienced, the connectivity to the micro-cloud suffers and there is a need to identify optimal transmission conditions.

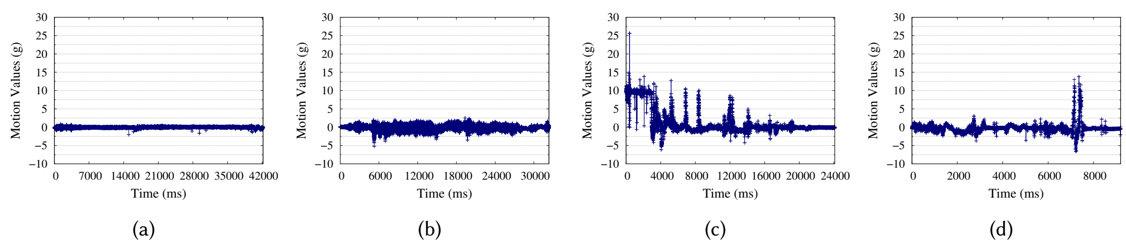


Fig. 3. Different types of water motion influence connectivity to the submerged micro-cloud, a-b) Low, c-d) High (induced by water vehicles operating nearby).

3 UNDERWATER MICRO-CLOUD DESIGN

We next describe the design of submersible micro-cloud (cloudlets) that are equipped with the capability to identify water stability and optimize transmission reliability.

3.1 Architecture

Figure 5a shows the overall architecture of a micro-cloud. The architecture consists of modules that encapsulate functionalities for computing, sensing, energy monitoring and other capabilities. We opted for a modular architecture

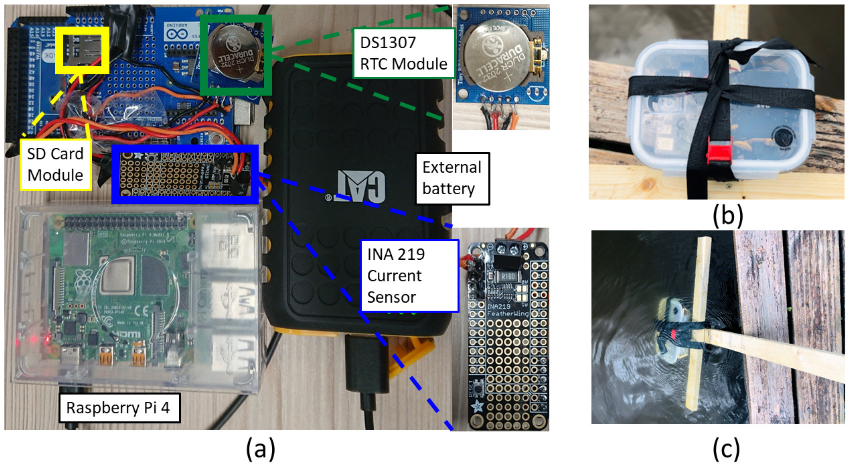


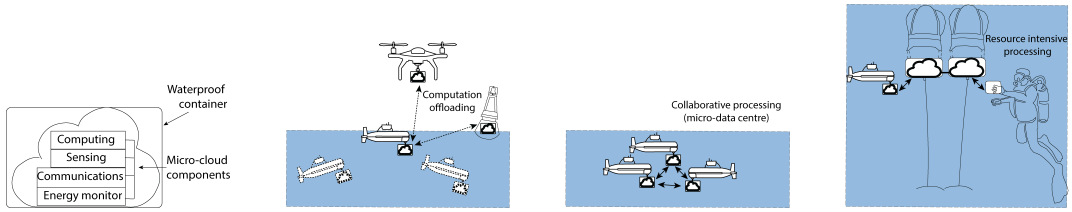
Fig. 4. Micro-cloud prototype: (a) Internal components, (b) Waterproof encasing, and (c) Deployment in the wild.

that can be easily extended. Micro-clouds rely on off-the-shelf devices that are fully portable, and have small size and weight. Moreover, off-the-shelf devices ensure flexibility to replace components and rapid prototyping. This is key to enabling the micro-cloud to be attached to other devices and infrastructure easily as shown in Figure 5. For instance, a micro-cloud can be easily attached to underwater drones or existing aquatic infrastructure, e.g., buoys, enabling the monitoring of pollutants (Figure 5b). Micro-clouds can also cooperate with each other to create high computing infrastructures via collaborative processing, and perform collaborative analysis (Figure 5c). In addition, micro-clouds can be deployed in specific locations aiding the devices in proximity (Figure 5d). By keeping the micro-cloud infrastructure in close range of the underwater IoT devices, it is possible to reduce the impact of computing operations on their constrained resources. This allows the underwater IoT devices to extend their exploration time, paving the way to apply more sophisticated fog analytic techniques.

3.2 Components

Computing: IoT devices have limited computing resources, which impose constraints to perform resource intensive analysis underwater, e.g., using machine/deep learning. To overcome such problem, submersible micro-clouds provide an additional computing component, allowing them to augment computing resources of devices. These micro-clouds use separate micro-controllers or smartphone devices as processing units. Thanks to their small size, weight and portability, such micro-cloud devices can be easily assembled, waterproofed and attached to an underwater device (e.g., ROV or AUV).

Communications: Micro-clouds provide interfaces that are accessible using common communication technologies. A micro-cloud can use these communication interfaces to respond to computing requests made from other devices, establish cooperation, collaborate to execute a task, and offload data and computation to external fog and cloud sources on the surface. While the absorption of the wireless signal underwater is an issue [80], it is still feasible to rely on wireless communication for underwater devices which are in the vicinity of each other, or close to the sea surface. In our case, we use Wi-Fi for communications between the devices in a single fog node (i.e., devices forming the micro-cloud)



(a) Micro-cloud architecture (b) Device augmentation (c) Micro-cloud cooperation (d) Dedicated micro-clouds

Fig. 5. Initial micro-cloud deployment, (a) Micro-cloud architecture, (b) augmenting individual devices, (c) collaborating to improve fog processing performance, (d) dedicated micro-clouds in a location that can be used by underwater IoT devices in a contact-based manner.

and between devices that are in close proximity to the fog (e.g., ROVs, AUVs, or divers that operate close to the fog node). A dedicated underwater communication interface is required to connect with buoys, ships, or other infrastructure or devices that are further away [62]. As an example, in section 6, we demonstrate how optical (light-based) communication can be used to increase the communication range and data transfer rate for devices that connect with the fog or with the devices that collaborate to form the fog [3, 27]. We stress that optical communications are not the only possibility for longer range connectivity as other forms of underwater communications, such as acoustic and electromagnetic communications, can be considered. These technologies currently require specialized instrumentation and cannot directly be used as a low-cost off-the-shelf solution for enabling fog computing.

Sensing: The micro-cloud prototype relies on sensors that are integrated in smart devices to estimate the water motion (turbulence) experienced by the micro-cloud. For instance, accelerometer and gyroscope sensors can be used to detect significant water motion. This information can then be used to detect optimal conditions for communication by detecting periods of low water motion to optimize the periods where data sampling is carried out. Indeed, turbulence can disrupt communications and induce resource overheads. Another use for motion sensors is to estimate the depth at which the micro-cloud has been submerged. For instance, high motion is experienced on the surface, and much less in the marine floor. We also envisioned additional sensors that are not inside the micro-cloud, but they are deployed outside the micro-cloud to collect information that can regulate its submerging process. Temperature and pressure sensors can provide information about depth, which can allow devices to trigger more intensive processing that can be cool down naturally through the environment. Oscillations of wireless signals could be also used to detect when devices are submerged, such that devices can adjust their duty cycling operations. For instance, automatic Wi-Fi discovery is reduced underwater. As another example, a sensor on the surface can inform the submerged micro-cloud to rise to the surface as there may be suitable solar power to recharge its battery resources.

Energy-monitoring: While micro-clouds are sufficient for augmenting the processing capabilities of underwater devices, isolated micro-clouds can suffer from high energy drain. Thus, outsourcing processing load by collaborating with other micro-clouds or offloading to dedicated fog and cloud infrastructure is required. As a result, the micro-cloud prototype is equipped with an energy monitor component which profiles the energy required by a particular task, allowing the later distribution of running processes.

3.3 Fog Provisioning Underwater

Current prototype of our micro-cloud for underwater edge deployments relies on wireless signals for the provisioning of services. As micro-clouds are formed by aggregated devices, we rely on the off-the-shelf service discovery mechanism integrated within smart devices of the micro-cloud. These mechanisms allow services to be discovered using the Peer-to-Peer functionalities, integrated within the default implementation of direct Wi-Fi (Wi-Fi P2P API). In this study, we demonstrate that light communication (See Section 6) releases the micro-cloud from the constrained signal coverage imposed by Wi-Fi (10 − 12 cm). However, since light technologies are not yet sufficiently mature to be adopted as an off-the-shelf technology, we expect the primary use-case for our fog to be with Wi-Fi. Therefore, we envision our micro-cloud to be used in two different manners. The former application can be used as and IoT device, anchored in a fixed location with a floating buoy for Passive Acoustic Monitoring (PAM), using LAN access points. In this case, micro-cloud is accessible on the water surface, and could be used, e.g., to support real-time analysis of underwater acoustic signals [53]. Latter application may be further enhanced by an AUV. In this setting, micro-clouds can get within vicinity to underwater device requiring computing power (e.g. cooperative multiple UAVs carrying cloudlets [43]). Other application examples include unobtrusive estimation of wave heights, localization of an AUV or divers underwater, and plastic or oil spill recognition.

4 COMPUTATIONAL BENCHMARKS

To demonstrate the potential of submersible fog to offer a *general purpose solution* for augmenting the computational capability of underwater applications, we next perform rigorous computational benchmarks using tasks that are representative of the processing needs of underwater data science applications, while also being representative of commonly used for fog benchmarks. In Section 7 we further demonstrate the feasibility of running the micro-cloud underwater by deploying it on the seabed.

In these experiments, we analyze the data processing and transfer requirements of several applications. We choose tasks from the DeFog benchmark suite [46] that have similar characteristics as tasks in underwater data science. We quantify the computation and communication latency, including the performance capacity of submerged micro-clouds to handle the workload of users accessing the processing resources. By using a wider set of applications with different computing requirements, it is possible to understand the stress of processing in the micro-clouds. We also measured the energy consumption of micro-clouds to process heavy computational tasks while submerged. Since submerged micro-clouds are not meant to be isolated, but merely supplement other infrastructure, we also conduct an experiment to measure the data transmission with external cloud and micro-cloud infrastructures, emulating a scenario where devices can offload processing when there is opportunistic connectivity to external sources. The overall prototype and the experimental testbed are depicted in Figure 4.

Experiments and Metrics: We measured multiple performance aspects of the micro-cloud. First, we measured both computation latency (RTT) and communication latency (CL). Computation latency is quantified using three factors, which include the time taken to access the resource, execution time of the task and time taken to send the result back. Communication latency measures the data transfer time from back and forth interactions without accessing the device. We also analyzed the capacity of the computing resources of the micro-cloud to handle a workload of concurrent users submitting tasks (multi-tenancy). We also measured the energy consumption (EC) of the micro-cloud when processing a computational workload.

Apparatus: We used a Raspberry Pi 4B (RPi4) and a LG G4 mobile phone (LGP) as processing units of the micro-cloud computer board. Each one is used separately, one at a time. RPi4 includes up to 4GB RAM and a Quad Core Cortex-A72 (ARM v8) 64-bit SoC @1.5GHz. LGP uses Android version 6.0 and has a removable Li-Ion 2540 mAh battery. To measure energy underwater for both processing units, we developed an energy monitor using Arduino board and Adafruit INA260 current sensor[3]. We allocated the current sensor in between the USB cable (positive wire) connecting RPi4/LGP and the external battery pack. Then the sensor connects to the I2C pins (SCL - I2C clock pin, SDA - I2C data pin) of the Arduino MEGA ADK development board to measure the current flow through the sensor. An application running the Arduino sketch then takes the current sensor information every 100ms and stores it in the SD card that is mounted on the Arduino board. By doing this, it is then possible to obtain real-time energy consumption of the micro-cloud while submerged. In addition, for improving the accuracy of energy measurements, we also attached a DS1307 Real-Time Clock (RTC) module[4] that provides the real-time timestamps for the current sensor readings. The accuracy of our energy measurements is comparable with the ones obtained by the off-the-shelf multi-meter, such as Peaktech 3430[5]. DeFog was executed on RPi4, where as LGP was used to execute the offloading applications.

Setup: Two sets of experiments were conducted, baseline and underwater experiments. Baseline experiments were conducted to benchmark the micro-cloud deployed outside the water. We then conducted the same experiment to analyze the influence of water when the micro-cloud is submerged. To do this, we encased our micro-cloud prototype into a waterproof (glass) container to protect the processing, energy, and sensor resources from water damage. We then submerged the micro-cloud underwater, and collected the previously described metrics. As the focus of these experiments was to benchmark computational performance rather than demonstrate practical feasibility, for these experiments we restricted ourselves to Depth-1 level (i.e, 1 - 5 cm depth) only and ensured a low water motion on the surface.

Tasks: To obtain performance metrics of the micro-cloud underwater, we relied on four different applications of DeFog [46]. Each application takes as diverse input types of assets to trigger the execution of the task. We also developed two offloading modes for migrating the computation to external sources. The first mode offloads a long data stream, whereas the second offload the computing tasks at a code level. We briefly describe the four DeFog applications below and give examples of underwater data science applications in different fields where the computational tasks are similar to those in the benchmark applications.

- **YOLO:** is a deep learning based object classification application based on YOLOv3 dataset. For this experimental procedure, it uses assets images with an average size of 223kb. Deep learning based object classification is commonly used in several underwater data science applications, such as marine plastic monitoring [28, 75] and reef ecosystem monitoring [72].
- **PocketSphinx:** is a speech to text conversion engine that uses audio files as assets. In this experiment, audio files have an average size of 207kb. A relevant example of an underwater data science application that uses acoustic signals is the discrimination of marine mammals from vocal calls [47].
- **Aeneas:** included a text-audio synchronization application that enforces alignments of text-audio entries. For this particular setting, audio files are used as assets, with an average size of 400kb. Forced alignment is used to assess

[3]https://learn.adafruit.com/
[4]https://www.adafruit.com/product/3296
[5]https://www.peaktech.de/

the levels of man-made noise pollution in marine environments, mapping the silence and noise levels of vessel propellers [78].

- **iPokeman:** is a latency critical GPS application for VR Online Mobile Games. It uses assets files with an average of 131kb. Georeferencing of marine mammal trajectories is used by citizen science applications to allow users to report marine mammal sightings while onboard sea vessels [76].

To generate the workload of users executing these applications, we used JMeter, which is a load-testing tool to generate dynamic workloads of users in a concurrent manner. With such a tool, we analyzed the influence of increasing workloads in the underwater micro-cloud. We used workload of users ranging from 1, 2, 5, 10, 25, 50, 100 and 250. To analyze underwater offloading, we developed two offloading applications. We describe these applications below.

- **Stream app:** The first application (Stream app) depicts long data streaming to an external source. It consists of an image processing application that implements a Box Blur filter. The application gets an image from the local device storage, and applies the filter to hinder features from the image via a blurring effect. We use different sizes of images, including 0.5Mb, 1Mb, 3Mb, and 5Mb.
- **Code app:** The second application (Code app) depicts computation offloading at the code level. It consists of a chess game application[6] which is based on the MinMax algorithm optimization. This application sends the current locations of the chessboard to the MinMax algorithm that calculates the best location for the next move. Unlike the Stream app, this application transmits small amounts of data with an average size of 170kb. This data is used to trigger a resource intensive processing.

For both offloading applications, we analyze the offloading process to execute to cloud and fog, respectively. In the next section, we portray the evaluation of our micro-cloud through rigorous experiments described in Section 4.

5 RESULTS

We perform extensive benchmarks to assess the performance of submersible micro-clouds, analyzing the overall data processing performance of underwater micro-clouds in terms of computation latency, communication latency, energy consumption and multi-tenancy capacity. We also compare submerged computing with a traditional above the surface deployment baseline.

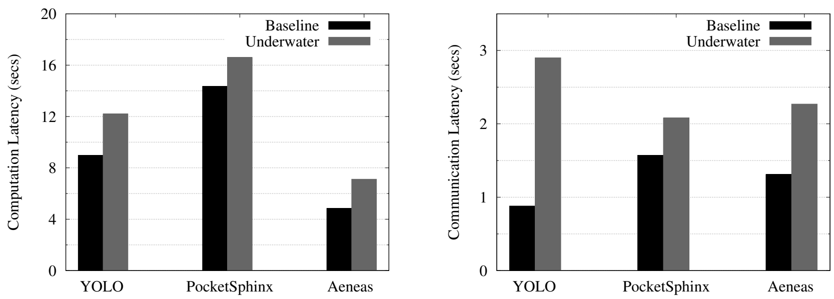
5.1 Underwater Computing Performance

We first quantify the impact in computation and communication latency when the micro-cloud is submerged and exposed to low water motion. Figure 6 shows the results for three different applications, including the YOLO, PocketSphinx and Aeneas applications. Baseline (outside water) results are also included for comparison. As expected, water induces an overhead on communication. For instance, for YOLO, it takes three times longer to transmit the assets. Likewise, PocketSphinx and Aeneas experienced an extra delay of one second to transmit the assets. More importantly, we can observe that the overhead also is present in computing latency, with task performance times slowing down as the micro-cloud is submerged. Above the surface, the average computing latency is eight seconds for YOLO. When the application is executed underwater, this increases up to 12 seconds. Considering that communication latency only adds up two seconds to transmit the assets, there thus is an additional two second delay used by the device for handling the data. This pattern can be also seen in PocketSphinx and Aeneas. We later demonstrate that this phenomenon is mostly

[6]https://github.com/huberflores/CodeOffloadingChess

present at shallow depths as heat transfer and cooling become more effective once the container is fully submerged and the depth increases above one meter.

All in all, our results indicate that applications that are in the proximal range can benefit from external resources of the micro-cloud, but that the performance gains may not be as high as above the surface. Nevertheless, considering that data transmission to the surface – let alone a remote cloud – is no longer needed, which implies there are clear benefits even if the performance of the submerged micro-cloud would not match that of a surface-based deployment. The results also show that these benefits can be achieved even when transmissions take place through common wireless communication instead of relying on specialized communication interfaces and thus there is significant potential to scale up computing support through the use of modular and easy-to-deploy solutions. For advanced technologies, our results suggest that placing the micro-cloud close to the application can help to minimize transmission power and thus help to prolong the operational time of the underwater applications.

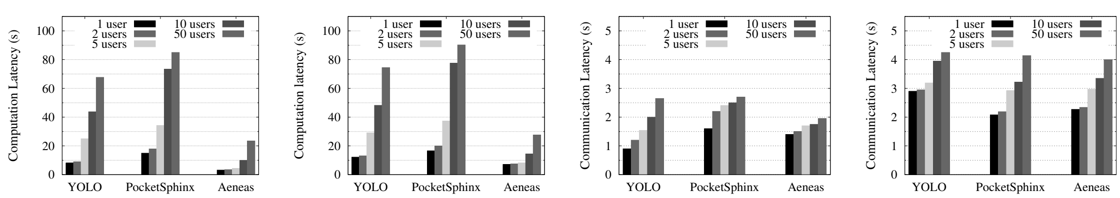


(a) (b)

Fig. 6. Latencies of different applications, a) Computation Latency, b) Communication Latency

5.2 Underwater Capacity Performance

Submersible micro-cloud deployments are expected to support multiple concurrent users or devices, requesting data processing resources for the completion of simultaneous tasks (i.e., multi-tenancy). To assess multi-tenancy capability of submersible micro-clouds, we next analyze the capacity of the micro-cloud to handle a workload of concurrent users submitting computing tasks to be processed in the micro-cloud.



(a) Baseline (b) Underwater (c) Baseline (d) Underwater

Fig. 7. Results of computation and communication latency for concurrent users, (a) Baseline computation latency, (b) Underwater computation latency, (c) Baseline communication latency, (d) Underwater communication latency.

Figure 7 shows the results. For comparison, we have included baseline results for computation and communication latency in Figure 7a and 7c, respectively. While we can observe an overhead in communication latency due to concurrent transmission when comparing Figure 7c and Figure 7d, we can observe the overhead to be small. For a workload of 50 users with Aeneas, an increase of only three seconds in communication latency is observed. For computation latency, shown in Figure 7a and Figure 7b, the overhead is larger. As an example, for the same Aeneas task, we can observe that a workload of 50 users requires in average 20 seconds outside the water, but it requires 30 seconds when the same setup is underwater, i.e., the overhead from submerging is similar to the overhead resulting from multiple users. In terms of individual users, the total delay is three times higher while submerging resulted only in an increase of 0.5 seconds in communication latency. The same pattern can be observed also with the other applications. The overhead percentage in computation latency when we consider baseline results and underwater results on all users for YOLO is 13 %, for PocketSphinx 6.63% and for Aeneas it was around 25%. The overhead in communication latency is more than computation latency. The overhead percentage in communication latency for YOLO is 41%, for PocketSphinx it is around 21% and for Aeneas it is 42%. Analysis of variance ANOVA test using baseline and underwater deployment as experimental conditions confirmed significant differences for computation ($\chi^2 = 11.27$, p< .001, Kendall's W= 0.99) and communication ($\chi^2 = 58.05$, p< .001, $\eta^2 = 0.996$) latency for concurrent users. The overhead is likely a result from device internal thermal management as the Raspberry PI uses throttling whenever temperature increases significantly. The differences in computational performance also suggest that submersible micro-clouds are best suited for small to moderate-scale deployments (e.g., up to 10 devices or users) and that surface-based edge is better for situations where a larger amount of simultaneous users needs support. Naturally, in a real aquatic deployment, the internal heat issue is easily overcome as the micro-cloud is submerged at higher depths (as demonstrated in Section 7).

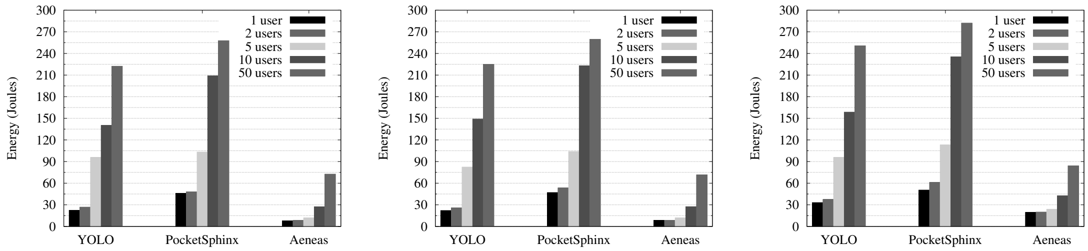
5.3 Underwater Energy Consumption

We also measured the energy consumption of the micro-cloud while executing applications underwater. Monitoring energy is important to offload computation and distribute the processing cost of a task among the available micro-clouds. We measure energy using the energy monitor implementation described in Section 4. We verified the accuracy of our monitor by using a multi-meter as the baseline. Figure 8 shows the results. The energy monitor is closely aligned with the multi-meter, providing fine-grained and accurate information on energy consumption. Non-parametric ANOVA test using multi-meter and energy monitor as experimental conditions confirmed there are no significant differences ($\chi^2 = 1.628$, p= 0.21, Kendall's W= 0.999) between the two sources of energy measurements.

While it has been demonstrated that the energy consumption of RPi4 does not differ between idle and active modes [48], it is still possible to observe an energy overhead when the RPi4 operates underwater. Results for underwater energy consumption are shown in Figure 8c, and show overhead caused due to transmission effort and heavier induced processing load in the computing resources. Conversely, when considering energy monitor deployment as experimental conditions, non-parametric ANOVA test verified that differences were statistically significant between energy monitors on surface and underwater ($\chi^2 = 15$, p< 0.001, Kendall's W= 0.999).

5.4 Submersed Fog to Cloud Performance

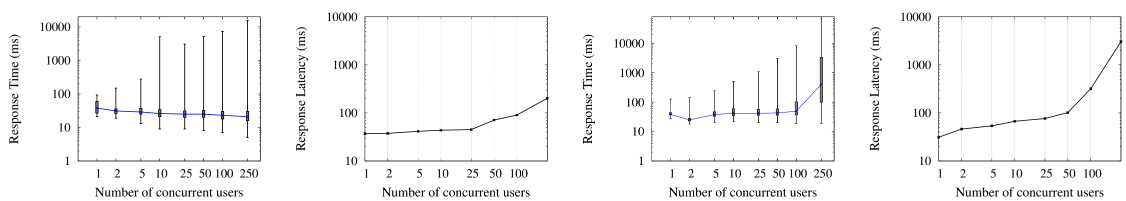
Since micro-clouds deployments are not isolated and need to synchronize with cloud, we also analyzed the communication from the submerged micro-cloud to cloud. To perform such step, we relied on iPokeman application. iPokeman application introduces an extra step in the execution of the task. Specifically, when iPokeman finishes the execution of a task in a micro-cloud, it also uploads the results to a server in the cloud. Figure 9 shows the results. We can observe from



(a) Baseline (multi-meter) (b) Energy monitor (c) Energy monitor (Underwater)

Fig. 8. Results of energy consumption underwater, a) Baseline with multi-meter, b) Energy monitor (outside the water) c) Energy monitor deployed underwater.

the results a higher overhead in communication latency. This extra overhead creates a bottleneck in the communication resources, which imposes a limitation on the number of concurrent users that can be handled. We can observe this when comparing Figure 9a and 9d. Conversely, from these figures we can also observe that the baseline still has enough available resources to handle more users beyond 250. In contrast, when the micro-cloud is underwater, we find an increment in computation latency due to resource over usage. Thus, micro-clouds need to be equipped with operation policies depending on whether they are on the water surface or submerged at a certain depth. The results also suggest that a submerged micro-cloud deployment is more effective at analyzing data underwater rather than at moving the data to surface infrastructure.

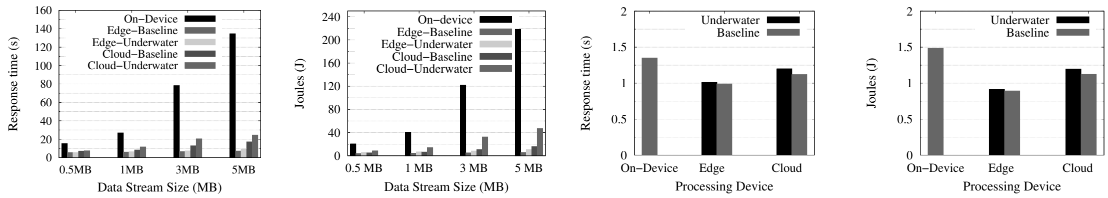


(a) Baseline surface (b) Baseline surface (c) Underwater (d) Underwater

Fig. 9. Impact of concurrent users using iPokeman, a) Response time, b) Response latency, c) Response time underwater, c) Response latency underwater.

5.5 Offloading from Underwater to the Surface

Figure 10 depicts results for offloading computing underwater. We measured the total energy consumption and response time of each application when (i) executed on the device, (ii) offloaded to an edge server (micro-cloud), and (iii) offloaded to a cloud server. From the results, we can observe that the Stream app consumes more energy when compared with the Code app. We also can observe that offloading to the cloud induces more overhead in both, response time and energy consumption when compared to the edge. We also find that same relative results are preserved when testing underwater. However, we also find that water induces higher energy consumption and response time for both applications. When offloading takes place between underwater devices in proximity, submerged micro-clouds become the edge, such that underwater IoT devices can obtain the same offloading benefits. Our results also indicate that while micro-clouds can be useful underwater, they can also be used to synchronize with surface-based infrastructure. However, given that communications are the main bottleneck, synchronization updates should be kept minimal.



(a) Response time (stream app) (b) Energy (stream app) (c) Response time (Code app) (d) Energy (Code app)

Fig. 10. Response time and energy consumption for both apps (a-b) Stream app, (c-d) Code app.

6 UNDERWATER OPTICAL COMMUNICATIONS

Since wireless communication is not reliable nor available for long distance communications in underwater environments, we next assess the potential of light communication as an alternative medium. Unlike radio frequencies, visible frequency spectrum has a lower attenuation in water. Thus, it is a promising technology to support communications over long distances for underwater systems. We next briefly describe the setup and results of our optical communication experiments.

Apparatus: We rely on a 650nm 5mW $3-5$V red laser diode as a light source, and an Arduino Mega ADK microcontroller (ATmega2560) to design a transmitter that uses light to transfer data. As a receiver, we use a solar panel (size 2.5cm x 2.5cm) connected to an Arduino board, which handles the received data. To induce water motion, we rely on two different sources to agitate the water continuously. We rely on a pond aeration pump (Ubbink Air 100[7]). The pump supplies air at the rate of 100 liters per hour with its three Watt air pump. Similarly, we also used a hand mixer with a potent motor (Model: House HB 1935, 200w) that generates high levels of turbulence.

Setup: Our testbed, illustrated in Figure 11, is built using a water tank of dimension 40 x 20 x 25 cm. We place the transmitter and receiver outside on opposite sides of the water tank. We then fixed the transmitter and receiver, such that the light emitted hit the solar panel in its center. We develop an application that transmits data using Morse code as an encoding mechanism. We rely on this mechanism as it is light for the constrained resources of the micro-controllers. We represent a dot by turning on the laser for 1 ms and a dash is represented by turning on the laser for 3 ms. These intervals can vary, but we found that speeding data transmission induces heavy processing in the devices. Thus, it causes a bottleneck when decoding the data. We used the optical configuration that does not induce much heavy processing.

Baseline: We placed the laser transmitter at one end of the corridor of the university building and the receiver at the other end of the corridor. The maximum transmission distance was approximately 100 m. With this distance, we were able to transfer a text file (1kB) to the destination within 5 s, and a text file (10kB) in about 50 s. More importantly, the light intensity of the transmitter was detected smoothly by the solar panel, such that there was no data loss at such a distance.

Experiment: We transmitted data in an interval of six minutes. In the first experiment, we captured the water motion experience when the water was calm. Next, we conduct an experiment in which the data transmission starts with calm

[7] www.ubbinkgarden.com

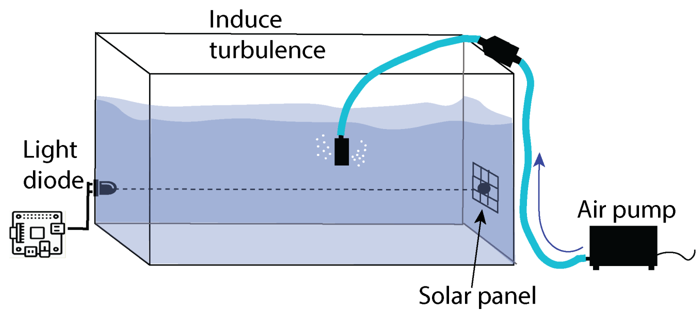
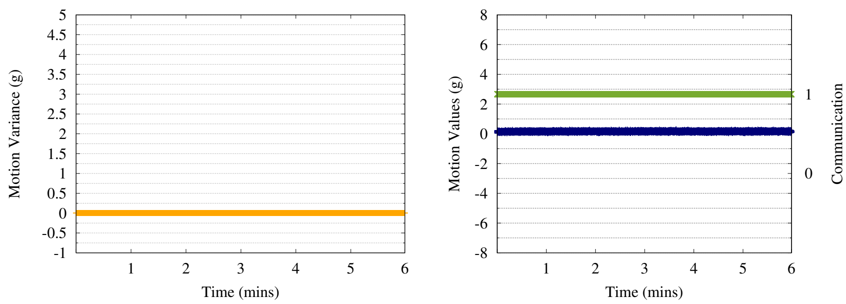


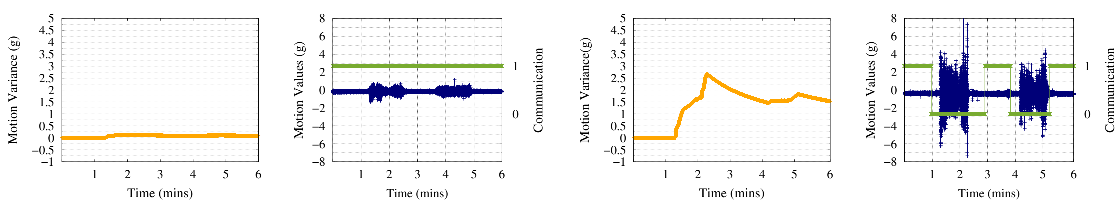
Fig. 11. Testbed for analysis of underwater communication with light.

water during the first minute, and then induced water motion is also generated for one minute. After that, the induced water motion is stopped, and we proceed to repeat the same experiment for the rest of the interval until the six minutes are completed.



(a) (b)

Fig. 12. Water motion (Calm water), a) Accumulated motion captured through all axes (variance), b) Experience motion (one axis only).



(a) (b) (c) (d)

Fig. 13. Light communication performance under different levels of induced water motion, a) Overall motion using AirPump (all axes) b) Motion using AirPump (one axis), c) Overall motion using a more potent motor (all axes) d) Motion using a more potent motor (one axis).

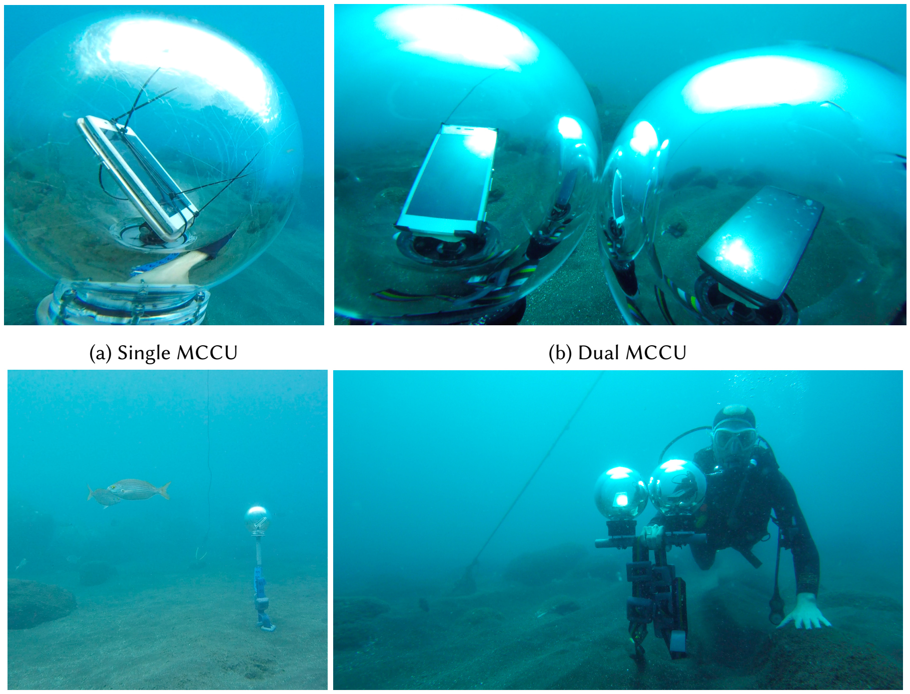
Results: Figure 12 shows the results of light communication within water. To quantify water motion, we used an accelerometer that is floating in the water surface and estimate the overall motion by calculating the variance of the three axes over a period of time. Figure 12a shows the overall motion in calm water. Similarly, Figure 12b shows the motion captured by the sensor (y axis only), and the success rate of communication, where 1 depicts a successful communication, and 0 a failure. Next, we analyze the performance of light communication when water motion is generated using the air pump. Figure 13a and Figure 13b shows the overall motion and the motion along the accelerometer y-axis. From the results, we can observe that the water motion induced by the pump is moderate, and does not disturb the communication with light. Baseline results are comparable with these two underwater cases, calm water, and induced water motion using the pump. Lastly, we also analyze the performance of light communication when water motion is generated using a more powerful motor (Ex.a hand mixer). Figure 13c shows the overall motion captured by the sensor. From the results, we can observe that water motion is higher using the mixer. More importantly, as shown in Figure 13d, we can observe that this type of induced water motion disrupts the communication with light. Non-parametric Spearman correlation [61] indicates a significant negative relation between high water motion and communication success ($\rho = -0.09$, $p < .05$). Interestingly, from the figure, we can observe that after stopping the communication using the mixer, a few seconds are required to re-establish the water's data communication link.

7 OCEAN DEPLOYMENT

The experiments thus far have focused on controlled benchmarks conducted using water containers or shallow depths. We next demonstrate that the solution is also feasible in actual underwater environments by considering experiments carried out as part of recreational scuba diving activity.

Apparatus: We consider a design where the devices forming the micro-cloud are placed inside a PVC supported acrylic sphere. As fog nodes, we consider (older model) smartphones that are repurposed to offer support for deep learning based object detection [56]. We perform two deployments, one with two phones inside the same sphere (Fig. 14a) and one considering two separate spheres that are attached to a horizontal bar and located next to each other (Fig. 14). As the spheres contain air inside, they would remain afloat, and we use diving weights to submerge them ($6kg$ for a single micro-cloud, and $12kg$ for dual micro-spheres) and to anchor them at the seafloor (Fig. 14c). We also consider a mobile scenario where a diver transports the micro-cloud while performing a dive transect survey (Fig. 14d). In all experiments, the phones placed inside the spheres were running deep learning based image recognition. As the runtime performance and resource use of the image classification can depend on the image that is given as input, we consider a fixed set of prerecorded images that were taken from underwater image feed [56]. Our motivation is that image data is the most common source for underwater data science as it can be collected without disrupting the environment. Indeed, image classification from image data is used, e.g., to support litter detection, fish school estimation, biodiversity monitoring, and pipeline leak estimation [56]. Thus, our experimental task is representative of the needs of real-world underwater data science applications.

Experimental Setup: Surface and underwater tests were performed at the Carlton diving reef in Madeira island. The former included a shade temperature of 25° Celsius while the latter with an underwater temperature of 21° Celsius at a depth of $8m$. Two dives were carried out (hereinafter Single and Dual MCCUs) with first having two mobile phones (a master and a worker) mounted inside the casing with screens on top of each other (Fig. 14a), and the second having three mobile phones (1 master and 2 workers) where worker phones were in the same container (Fig. 14b). The master device was a Sony Xperia M2 with Quad-core 1.2 GHz Cortex-A7 CPU, running Android 5.1. Two used workers were NOS NOVU II Android 5.1 and Alcatel Go edition running Android 11. For the task we consider image classification by having the previously trained ImageNet models on the device. As the focus was to test the underwater performance of



(a) Single MCCU (b) Dual MCCU

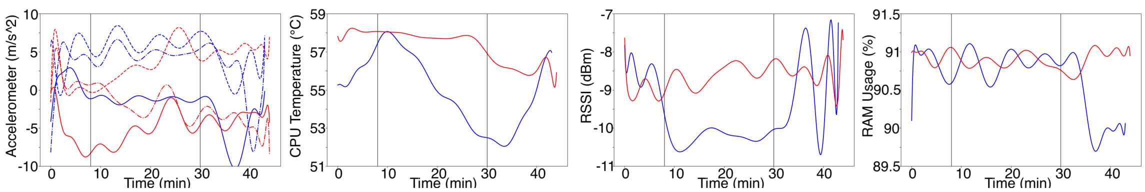
(c) Seabed deployment (d) Dive survey

Fig. 14. Underwater Micro-Cloud Computation Units (MCCUs) consisting of individual mobile devices, communicating using off-the-shelf integrated wireless interfaces while collaborating in the execution of a task, deployed in open sea at $8m$ depth. From left to right, top to bottom: (a) Master and worker within same unit, (b) Master to the left and two workers to the right, (c) Seabed deployment and (d) SCUBA diver survey.

the fog, the same 5 images were used for all tasks to ensure the memory requirements remained comparable throughout the experiment.

Single MCCU. This experiment included one master and one worker phone with data collected at the surface and in underwater setting at $8m$. Two phones ran separate mobile applications allowing them to communicate through WLAN and participate in computation task. As in previous experiments, the worker was connected to the master's access point, where the master was sending 5 images to the worker, following the Round-robin scheduling. The worker phone then performed image classification tasks on the CPU and transmitted the obtained accuracy back to the master mobile phone through WLAN. The duration of the experiment was $44min$ for both surface and underwater setups. Data inquiry included the environmental variables such as accelerometer, RSSI, RAM percentage and CPU temperature obtained from the worker phone, with a duty cycle of $1/15Hz$. The surface test was made by walking with the MCCU prior to the underwater test. The underwater test (SCUBA-dive time) was split into pre-dive ($11min$), dive ($22min$) and post-dive time ($11min$), assuring that the dive time is at a depth of $8m$.

Figure 15 depicts the accelerometer, CPU, RSSI and memory usage data from the experiments. In the figure we compare the underwater (blue) and surface (red line) setups in the same micro-cloud. The vertical lines highlight the transitions from pre-dive (i.e., surface tasks preparing for the dive) to the actual dive and from the dive to the post-dive period. During the dive the device was taken to 8 meter depth and remained there until surfacing at the end. The period where the micro-cloud was submerged can be clearly seen from the drop in RSSI values. Similarly, the temperature of the cloud starts to decrease rapidly as the dive begins, thus the temperature effects mainly relate to shallow water close to the surface and at deeper depths the heat transfer to the exterior is sufficient to normalize the operations of the prototype. The variations at the beginning and end (i.e., during the pre-dive and post-dive periods) are mostly due to obstructions caused by moving the objects. For example, the early difference in temperature between the two micro-clouds is simply a result of one device being in the shade whereas the other was exposed to direct sunlight. As both the master and the worker were located inside a container, the communication between the devices participating in the cloud remained stable throughout the experiment despite the RSSI weakening.



(a) Accelerometer (b) CPU Temperature (c) RSSI (d) RAM Usage

Fig. 15. Obtained sensor data during ocean deployment experiment time. From left to right: (a) Triaxial accelerometer, (b) CPU temperature, (c) RSSI, and (d) RAM Percentage. Red line depicts the time at the surface, while blue line is the time underwater. Vertical lines indicate the SCUBA diving time for underwater test.

Dual MCCU. Since the main bottleneck for using a single MCCU is a challenge to establish communication from a client located outside the container with a client that is located inside it, we perform a second experiment, submerging the two micro-clouds with the 3 phones (Fig. 14b). One master device was used in one micro-sphere, having the two workers in the second micro-sphere. Distance between the two micro-spheres was $5cm$, respecting the pre-established range of Wi-Fi communication and transmission of the data through salt water [67]. Obtained RSSI between the two micro-spheres at a depth of $8m$, indicated $-25dBm$ for surface and $-41dBm$ for underwater setting, indicating the stable data transmission in both settings. In Table 1 we showcase the obtained computation time when performing image classification on 301 images during $10min$. Using a single MCCU setup, 301 images were successfully classified within $19.26min$. Conversely, the dual MCCU setup shown the 301 images to be successfully classified in $12.3min$. This was a speed-up of nearly $7minutes$, being consistent with the previous work [40]. Results indicate that it is possible to perform robust computation in real ocean settings and that having more micro-clouds underwater can indeed speed-up the computation time.

Table 1. Comparison of underwater computation times during the image classification for 10 minutes. Increasing the number of workers expectedly speeds up computations which implies that the connectivity between devices is sufficiently stable to operate the micro-cloud despite adverse water conditions.

No. of phones	Comp. Time (s)	Mean (s)	St. Dev. (s)	No. of images	Setup
2	1156 (19.26 min)	3.84	0.3	301	Single MCCU
3	742 (12.36 min)	2.47	0.3	301	Dual MCCU

8 DISCUSSION

We have shown submersible micro-clouds to offer a potential way to support underwater data science applications and demonstrated that COTS components could be used to enable a submersible micro-cloud. Below we briefly discuss implications, possible extensions, the main limitations, and potential ways to overcome them.

Application Domains: The primary target for our research is underwater data science investigations that operate within a localized region. Examples of such applications include diverse monitoring tasks, such as pipeline integrity monitoring, reef condition monitoring, biodiversity monitoring, smart underwater navigation and litter monitoring [32, 52, 56]. At the same time, our solution can also support surface-based marine data science applications in areas that lack access to a traditional communication infrastructure (i.e., away from shore regions). Offering computational support for these kinds of applications is critical for scaling up such applications and reducing the delay between data collection and analysis. At the same time, the computing infrastructure needs to be environmentally sustainable to ensure it offers the required computational power but does not harm the underwater environment. Micro-clouds, as envisioned in our work, are well-suited for these needs, offering powerful and scalable computing support for applications that are characterized by high data velocity [40], while at the same time being easy to deploy on-demand. Indeed, our solution can be easily deployed as a temporary infrastructure instead of requiring a persistent deployment. As an example, we are currently using our solution to support scuba divers in underwater litter recognition by allowing the infrastructure to be deployed at the beginning of a dive and removed at the end of the time [56].

Implications for Fog Computing: A fundamental challenge in adopting any computational infrastructure is the need for a static and permanent physical deployment location. Indeed, available space, suitable deployment facilities, e.g., rack, cooling system, energy supplies, and sufficient computing hardware are critical to providing services to a large number of users. Our work offers a way to alleviate these issues by adopting small scale data-centres that can be submerged on-demand and potentially moved taking advantage of currents. Our results also suggest that old computing devices could be recycled for underwater settings to provide computing infrastructure near users. Naturally, it is important to ensure the micro-clouds are safely attached so that they do not get lost and end up polluting the underwater environment.

Towards large-scale deployments: The present paper demonstrated how micro-clouds could be easily deployed in the open sea for short-term deployments, e.g., included as part of scuba diving missions. For longer term deployments that could provide a broader range of underwater scenarios, such as sensors deployed onto the seabed, there are further challenges that need to be addressed. First, the waterproofing we used was designed to protect the computing units, not to offer a long-term solution. Indeed, we used acrylic spheres for the encasing of computing resources. These can only handle moderate depths (e.g., around 20 meters which is a common recreational diving depth) before pressure accumulation would break the spheres. As a result, better encasing solutions are needed for longer-term deployments and for operating deeper. Optimally, the casing should also offer adjustable buoyancy, enabling it to operate at different parts of the water column instead of being limited to the seabed. Finally, adverse environmental conditions, such as heavy turbidity, salinity or turbulence also pose challenges that require further research. Nevertheless, our research shows how it is possible to support underwater data science applications through low-cost micro cloud based fog designs, offering the first steps in developing broader computing support for underwater deployments.

Communication Interfaces: Ideally, a submersible micro-cloud has to be equipped with multiple communication interfaces, and a context-aware mechanism to decide which communication interface to use based on water conditions, e.g., high turbulence. While we have shown that submersible micro-clouds are feasible and useful (with close contact using wireless and longer range connectivity using optical connectivity), practical deployments would need to support distances of a few meters at a minimum. This can only be achieved through the use of communication mediums that are suited for underwater environments, such as optical, acoustic, laser, or electromagnetic communications. Currently, these technologies have not yet reached a level where standardized communication interfaces – let alone low-cost ones – would be available. For these reasons, we omitted their use within the benchmarks, as integration with proprietary interfaces would result in additional overheads and make it difficult to separate the computational performance of the offloaded task and applications from the overheads caused by the communication interface.

Water Conditions: The experiments were conducted both in a river and in an ocean environment with varying water motions. The results were stable across these experiments, and showed stable computational performance even in the presence of significant water motion. The main challenge, thus is not the computational aspect of the fog node, but to having robust enough connectivity. In practice, salinity, turbidity, level of pollution and extent of algae can affect the performance of both the micro-cloud and the communication interface. In particular, water characteristics affect propagation of signals as well as the thermal conductivity of the water, which is critical for cooling and thermal management of the submersible micro-clouds. Additionally, algae or particles in the water can accumulate on the surface of the casing hosting the fog and this can reduce access from the outside of the casing. Overcoming these issues requires improved material designs (that are beyond the scope of our work) for casings besides more robust and affordable communication interfaces.

Surface-Based Micro-Cloud deployments: We demonstrated that both extents of water motion and depth of the deployment affect computational performance. Besides being highly relevant to submersed deployments, these results are also highly relevant for surface-based deployments, e.g., other computing infrastructure attached to buoys or sea vessels. Waves and other water motions can cause surface-based deployments to be momentarily submerged, which can cause disruptions in handling and processing requests. The reliability of such deployments can be improved by integrating motion-based techniques, such as accelerometers used in our work.

On-demand Fogs: While submersing and deploying permanent micro-clouds in critical areas that require continuous monitoring is important, other areas that are monitored occasionally can rely on on-demand infrastructure that is carried and deployed temporally in a location. For instance, aerial and underwater autonomous vehicles can be used for this purpose. Similarly, other transportation means can be envisioned, such as hot air balloons, airships and mobile buoys.

Multi-Modal Energy Harvesting: Tidal harvesting and solar cells are two of the most promising technologies for generating sustainable energy for underwater devices. While several works have demonstrated that energy can be harnessed using these techniques, the energy gains they offer remain small and are unlikely to suffice the needs of underwater devices facing continuous and resource intensive processing. This suggests that a multi-modal approach to preserving battery life underwater can be more effective in fostering longer explorations. We demonstrate the usage of computation offloading which complements that vision. Indeed, by using computation offloading underwater, it is possible to preserve devices underwater for longer periods of time.

Thermal Management and Casing: Our implementations of micro-clouds used sealed waterproof containers for the micro-controllers. The lack of heat exhaust can result in heat accumulation inside the container, which in turn can trigger device-internal thermal management which throttles the CPU performance. We have observed this phenomenon in earlier experiments that were conducted near the surface, but in the ocean experiments this did not occur. This is potentially a result of better thermal management in the devices that are used as fog nodes and from the deeper water being able to cool the container effectively. Further improving the performance of submersible micro-clouds requires research on casing solutions that are sufficiently lightweight to allow attaching the infrastructure into underwater devices or objects, while at the same time having sufficient cooling capacity. Another limitation of the casing that we used in the experiments is that it suffers from the fact that it houses air inside it, requiring separate weights for submerging it. Removing – or at least reducing – the air pockets is thus needed to make the overall platform easier to deploy.

Recycling opportunities for e-waste: Electronic waste from smart devices is a global concern as it pollutes natural ecosystems and fosters climate change. In our work, we demonstrate that micro-clouds can be made from aggregated smart and IoT devices. We envision that computing resources from old phones can be recycled to create portable computing racks, which then can be deployed on edge underwater to provide public services to users. For instance, a video streaming service for tourists about the sightseeing places in a city.

AUVs for fog delivery: In our work, we demonstrated the design and development of micro-clouds that can be used for underwater edge deployments. One important insight of this work is related to the weight of the micro-cloud. Indeed, micro-clouds are lighter in weight and have compact size after being encased. Thus, micro-clouds can then be transported underwater easily by AUVs. This suggests that it is possible to provide mobile fog computing services underwater.

9 RELATED WORK

Our work draws inspiration from edge and fog computing research and from underwater IoT and data science. We next review relevant works in these fields to highlight key requirements and challenges for providing computing support. Table 2 provides a summary of this comparison, providing the proposed name of the solution (Proposed); the type of deployment supported by the solution (Deployment); the type of hardware used as underlying processing infrastructure per (fog) processing unit (Underlying hardware); the available communications that the deployment provides (Available communications); the flexibility of the solution to be used underwater (Ready for underwater deployment) and whether the solution exploits COTS components to provide high replication rate and large-scale adoption (COTS components). The key novelties of our work are solutions to several practical challenges resulting from the underwater environment, and the provisioning of an underwater computing infrastructure (micro-cloud) that is low-cost and easy to implement. Achieving this is critical for ensuring that computing support can be easily deployed and used to support diverse underwater (pervasive) data science applications. Indeed, unlike our work, existing solutions are limited to offering access to external, surface-based infrastructure, or augmenting the capabilities of individual devices, without offering a general purpose platform that can simultaneously support multiple devices and serve the needs of a broad range of underwater (pervasive) data science applications.

Computational Augmentation: Edge and fog computing are the main paradigms for augmenting computing by offering processing, intelligence, storage, and other functionality close to the data source [79]. Edge computing provides services that are in the vicinity of the data sources, such that there are no oscillating changes in communication latency that hamper the battery and performance of applications. For example, improving battery saving by caching data on the edge has been investigated [77]. A key limitation with edge computing is the lack of dense and ubiquitous deployments to provide continuous support to end-applications. Fog computing, in turn, assumes the support covers data storage and is able to integrate intelligence [79]. Fog computing can be delivered from any device with enough processing resources that is blended within the environment. For example, common devices acting as fog nodes, include, access points, IoT

Table 2. Summary of most relevant work for fog provisioning. Our work is the first to offer fog computing support consisting of multiple devices that is ready for underwater deployment and that uses low-cost COTS components.

Proposed	**Deployment**	**Underlying hardware (per node)**	**Available communications**	**Ready for underwater deployment**	**Off-the-shelf components**
VM-Based Cloudlets [60]	edge/fog	individual	Wireless LAN	no	no
OREO [77]	edge	multiple	Wireless LAN	no	no.
Fog micro datacenter [1]	datacenter	multiple	Wireless LAN	no	no
Pocket Cloudlets [38]	mobile-device	multiple	Wi-Fi/Cellular	no	yes
Pervasive Data Science [40]	edge	multiple	Wireless LAN	no	yes
Smartphone cluster [8], MISCO [22]	mobile devices	multiple	Wi-Fi	no	yes
FemtoClouds [29]	edge	multiple	Wi-Fi	no	yes
Collaborative processing methods [41, 69, 74]	edge	multiple	Wi-Fi	no	yes
Cloudrone [59]	edge	individual	Wi-Fi	no	yes
Geographical relocation methods [70], [4], [14]	data center	multiple	co-axial/fiber optic	yes	no
Natick [17], [51]	data center/ cloud	multiple	co-axial/fiber optic	yes	no
IoUT [37], [21], [35]	wireless sensor networks	multiple	Wi-Fi	yes	no
Underwater exploration and monitoring [2], [64], [65], [30]	wireless sensor networks/AUVs	individual	Wi-Fi	yes	no
Underwater infrastructure [33], Aqua-Fi [62]	WSN, drones	individual	Wi-Fi/optical	yes	no
Penguin [26]	micro-cloud/AUV	individual	Wi-Fi	yes	yes
POSEIDON [53], [11], [34]	edge	individual	acoustic/wired	yes	yes
Deep learning in Oceans [56]	AUV/ROV	individual	Wi-Fi	yes	yes
Our work	fog	multiple	Wi-Fi	yes	yes

devices, cloudlets and edge servers. Our work explores a new frontier for fog computing, developing and deploying micro-clouds in underwater environments to increase the ubiquity of access to processing resources for underwater applications. Existing works cannot be directly adopted in underwater environments as there are unique challenges when operating underwater. Our work addresses some of the key challenges, including water motion, poor wireless propagation, and the need for sufficient waterproofing, and highlights their impacts on delivering computational support. We also demonstrate how off-the-shelf devices can be harnessed for underwater needs. Our work serves to pave the way for real-time data analytics in underwater environments.

Cloudlets and Micro-clouds: Cloudlets provide computing power close to users [60] and are the foundations of edge and fog computing paradigms. A micro-cloud, in turn, depicts an extended form of a cloudlet, whose underlying resources are aggregated using multiple distributed devices [23, 39]. Since smart and IoT devices have increased computational capabilities, approaches to create dynamic micro-data centres with them have been proposed [1, 38, 40]. It has been demonstrated that a rack of smartphones can be used to create a cloud computing-like infrastructure [8, 22]. Moreover, collaborative processing and federated learning between devices can be used to create dynamic and elastic computing infrastructures on the edge [29, 41, 69, 74]. Also, micro-clouds can operate on aerial drones at the edge [59]. Our work draws inspiration from the possibilities offered by micro-clouds, addressing key challenges and developing the necessary support to deploy and operate them in an underwater environment, as well as demonstrating the benefits micro-clouds can bring to underwater data science.

Data Centre Deployments: Among numerous ecological challenges, reduction of data centres emissions is an issue that has been investigated widely to overcome the impact on climate change [5, 70]. Data centres have been moved to different geographical locations, in order to improve cooling and provisioning of services to end-users [4, 14]. While dunking the data centers has been explored by Microsoft in its Naptik [17], underwater data centres can take advantage of low temperature sea floor and reduce cooling power, improving energy efficiency [73]. Optimal deployment of marine cloud computing have been investigated [51]. These all suggest there is significant potential in underwater computing infrastructure that warrants further investigation. Our work explores the design of (small size) micro-clouds that can be deployed underwater near where the computations are performed. These micro-clouds can provide localized computing support for underwater data science applications, complementing the deployment of dunked data centres and providing broader coverage of computing resources over targeted areas.

Internet of Underwater Things: IoT has increasingly large scope in underwater scenarios, and many of the application scenarios have been covered in previous works [21, 35, 37, 56]. Most of the existing work has focused on developing applications for specific purposes, including marine pollution monitoring [2], aquaculture [64, 65] and study of marine life [6, 18, 35]. Other work has focused on adopting existing technologies and developing new ones for underwater usage [37]. Also, underwater sensors networks have been studied in detail [24, 30] and several other technologies have been integrated into them, e.g., drones [33], and floating infrastructure [62]. In particular, different communication technologies have been explored [13, 54, 55, 58]. Despite the increasing amount of research, very little work exists on augmenting the processing resources of IoUT applications. Other works have attempted to facilitate data gathering from oceans using underwater sensor networks (UWSNs) by proposing efficient routing protocols [81]. A programmable Internet of Underwater Things (IoUT) project called SEANet has also been developed to make it convenient to add , remove and replace both hardware and software [20]. Other works have focused on making the IoUTs scalable [49]. Our work addresses the gap in the availability of processing resources in underwater environments, developing a practical solution that addresses challenges in operating external computing infrastructure in an underwater environment and that is beneficial for augmenting the processing capabilities of these types of IoT applications and deployments.

Underwater Data Science: The most common method to analyze underwater data is to use passive analysis, where data is collected from underwater and then taken out to be processed by surface-based infrastructure [19]. Large amounts of data are collected underwater that require on-site data analysis, e.g., deep learning [56]. Multiple tools have been developed to support deep learning applications for underwater data analysis [11, 34, 53], e.g., whales with CurvRank [9], dolphins with finFindR [68], NNPool [44] and PhotoID Ninja [7], and turtles with MYDAS [12], among others. Despite these solutions targeting analysis of underwater data, they do not operate underwater and require surface based infrastructure. In parallel to this, autonomous vehicles have been equipped with augmented computing payloads to perform processing underwater using deep learning [26, 56]. Autonomous vehicles have limited operational time underwater and its difficult to extend their design for augmentation of computing resources [43]. Thus, alternative deployment to obtain additional computing support is required. Our work provides one such alternative, developing micro-clouds that can operate underwater and demonstrating how they can serve the localized processing needs of underwater applications. This makes it possible for applications to augment their processing capabilities without relying on surface-based deployments.

Summary of Literature Review: Underwater environments remain highly challenging for computing and, currently most underwater data science applications rely on scenarios where either surface-based computing support is reachable (e.g., buoys close to access point stations or tethered drones) or there is a significant delay between the collection and analysis of data [56]. Our work addresses technical challenges in developing a general-purpose solution for improving access to computing resources, offering an easy-to-implement and low-cost solution for delivering computing support to underwater applications. Our work is the first to provide such a platform and to solve technical challenges in deploying and operating the platform underwater. At the same time, our work is firmly grounded on the current state-of-the-art in fog and edge computing, including the delivery of micro-clouds, but extending these solutions to highly challenging underwater environments. While fog and edge computing technologies have been explored previously, translating existing solutions from surface-based infrastructure to underwater computing support is non-trivial and requires addressing challenges resulting from this shift in the environment. We address these challenges, including limited wireless connectivity, water motion, and need for water-proofing, developing solutions and analyzing their effects on the computing support that can be offered.

10 CONCLUSIONS

We developed a novel submersible fog computing approach that offers a general-purpose solution for augmenting computational capability for a wide range of underwater data science applications. We demonstrated the feasibility of our approach through a proof-of-concept offloading framework that has been implemented on micro-controllers and off-the-shelf devices placed in waterproof containers. Off-the-shelf devices make our approach suitable for large-scale deployment and easy to adopt in different contexts. We performed rigorous experiments to analyze micro-cloud performance in underwater settings, demonstrating that water motion and depth are critical factors affecting computational performance. We also demonstrated the practical feasibility of our approach by deploying our solution at 8 meter depth on the ocean seabed. We also demonstrate the benefits of our approach when using multiple micro-clouds at once. Taken together, our results show that submersible micro-clouds can offer general purpose support for a wide range of computational tasks, but that additional intelligence is required to manage how and when to best utilize these resources. We presented one such solution, showing how accelerometer-based sensing can aid in deciding suitable communication times by detecting water motion levels. We also provide lessons learned and experiences from our experiments conducted in the open sea. Our work serves as an important first step toward enabling underwater data science applications to access more powerful computing resources than what is currently possible.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their insightful feedback. Research was partially supported by the grants LARGESCALE (PTDC/CCI-CIF/32474/2017), INTERWHALE (PTDC/CCI-COM/0450/2020) and INTERTAGUA (MAC2/1.1.a/385) by FCT, MCTES, PIDDAC, and MAC-INTERREG funding respectively. This study had the support of FCT through the strategic project UIDB/04292/2020 awarded to MARE and through project LA/P/0069/2020 granted to the Associate Laboratory ARNET. This research has been also financed by European Social Fund via "ICT programme" measure and the Academy of Finland project 339614.

REFERENCES

[1] Mohammad Aazam and Eui-Nam Huh. 2015. Dynamic Resource Provisioning Through Fog Micro Datacenter. In *IEEE International Conference on Pervasive Computing and Communication Workshops*. IEEE, IEEE, St. Louis, MO, USA, 105–110. https://doi.org/10.1109/PERCOMW.2015.7134002

[2] Mutiu Adesina Adegboye, Wai-Keung Fung, and Aditya Karnik. 2019. Recent Advances in Pipeline Monitoring and Oil Leakage Detection Technologies: Principles and Approaches. *Sensors* 19, 11 (2019), 2548. https://doi.org/10.3390/s19112548

[3] Mohammed Al-Khafajiy, Thar Baker, Hilal Al-Libawy, Zakaria Maamar, Moayad Aloqaily, and Yaser Jararweh. 2019. Improving Fog Computing Performance Via Fog-2-Fog Collaboration. *Future Generation Computer Systems* 100 (2019), 266–280. https://doi.org/10.1016/j.future.2019.05.015

[4] Howraa Mehdi Mohammad Ali, Taisir EH El-Gorashi, Ahmed Q Lawey, and Jaafar MH Elmirghani. 2017. Future Energy Efficient Data Centers With Disaggregated Servers. *Journal of Lightwave Technology* 35, 24 (2017), 5361–5380. https://doi.org/10.1109/JLT.2017.2767574

[5] Maria Avgerinou, Paolo Bertoldi, and Luca Castellazzi. 2017. Trends in Data Centre Energy Consumption Under the European Code of Conduct for Data Centre Energy Efficiency. *Energies* 10, 10 (2017), 1470. https://doi.org/10.3390/en10101470

[6] Scott Bainbridge, Damien Eggeling, and Geoff Page. 2011. Lessons From the Field—Two Years of Deploying Operational Wireless Sensor Networks on the Great Barrier Reef. *Sensors* 11, 7 (2011), 6842–6855. https://doi.org/10.3390/s110706842

[7] Soren Bouma, Matthew DM Pawley, Krista Hupman, and Andrew Gilman. 2018. Individual Common Dolphin Identification Via Metric Embedding Learning. In *International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, IEEE, New Zealand, 1–6. https://doi.org/10.1109/IVCNZ.2018.8634778

[8] Felix Büsching, Sebastian Schildt, and Lars Wolf. 2012. DroidCluster: Towards Smartphone Cluster Computing–The Streets are Paved With Potential Computer Clusters. In *International Conference on Distributed Computing Systems Workshops*. IEEE, IEEE, Macau, China, 114–117. https://doi.org/10.1109/ICDCSW.2012.59

[9] John Calambokidis, Jay Barlow, Kiirsten Flynn, Elana Dobson, and Gretchen H Steiger. 2017. Update on Abundance, Trends, and Migrations of Humpback Whales Along the US West Coast. *IWC Report SC A* 17 (2017), 18–21. https://www.cascadiaresearch.org/files/RS6796_SC_A17_NP_13.pdf

[10] Zachary R Caldwell, Brian J Zgliczynski, Gareth J Williams, and Stuart A Sandin. 2016. Reef Fish Survey Techniques: Assessing the Potential for Standardizing Methodologies. *PLOS ONE* 11, 4 (2016), 1–14. https://doi.org/10.1371/journal.pone.0153066

[11] Shuo Cao, Dean Zhao, Xiaoyang Liu, and Yueping Sun. 2020. Real-Time Robust Detector for Underwater Live Crabs Based on Deep Learning. *Computers and Electronics in Agriculture* 172 (2020), 105339. https://doi.org/10.1016/j.compag.2020.105339

[12] Steven JB Carter, Ian P Bell, Jessica J Miller, and Peter P Gash. 2014. Automated Marine Turtle Photograph Identification Using Artificial Neural Networks, With Application to Green Turtles. *Journal of Experimental Marine Biology and Ecology* 452 (2014), 105–110. https://doi.org/10.1016/j.jembe.2013.12.010

[13] Charles J Carver, Zhao Tian, Hongyong Zhang, Kofi M Odame, Alberto Quattrini Li, and Xia Zhou. 2020. AmphiLight: Direct Air-Water Communication with Laser Light. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association, USA, 373–388. https://www.usenix.org/conference/nsdi20/presentation/carver

[14] Jakob Dybdal Christensen, Jens Therkelsen, Ivo Georgiev, and Henrik Sand. 2018. *Data Centre Opportunities in the Nordics: An Analysis of the Competitive Advantages*. Nordic Council of Ministers, Copenhagen, Denmark.

[15] Alastor M Coleby and Eric PM Grist. 2019. Prioritized Area Mapping for Multiple Stakeholders Through Geospatial Modelling: A Focus on Marine Plastics Pollution in Hong Kong. *Ocean & Coastal Management* 171 (2019), 131–141. https://doi.org/10.1016/j.ocecoaman.2018.12.021

[16] Jun-Hong Cui, Jiejun Kong, Mario Gerla, and Shengli Zhou. 2006. The Challenges of Building Mobile Underwater Wireless Networks for Aquatic Applications. *IEEE Network* 20, 3 (2006), 12–18. https://doi.org/10.1109/MNET.2006.1637927

[17] Ben Cutler, Spencer Fowers, Jeffrey Kramer, and Eric Peterson. 2017. Dunking the Data Center. *IEEE Spectrum* 54, 3 (2017), 26–31. https://doi.org/10.1109/MSPEC.2017.7864753

[18] Almir Davis and Hwa Chang. 2012. Underwater Wireless Sensor Networks. In *2012 Oceans*. IEEE, IEEE, Hampton Roads, VA, USA, 1–5. https://doi.org/10.1109/OCEANS.2012.6405141

[19] Matthew Dawkins, Linus Sherrill, Keith Fieldhouse, Anthony Hoogs, Benjamin Richards, David Zhang, Lakshman Prasad, Kresimir Williams, Nathan Lauffenburger, and Gaoang Wang. 2017. An Open-Source Platform for Underwater Image and Video Analytics. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, IEEE, Santa Rosa, CA, USA, 898–906. https://doi.org/10.1109/WACV.2017.105

[20] Emrecan Demirors, Jiacheng Shi, Anh Duong, Neil Dave, Raffaele Guida, Bernard Herrera, Flavius Pop, Guofeng Chen, Cristian Casella, Sayedamirhossein Tadayon, et al. 2018. The SEANet Project: Toward a Programmable Internet of Underwater Things. In *Underwater Communications and Networking Conference (UComms)*. IEEE, IEEE, Lerici, Italy, 1–5. https://doi.org/10.1109/UComms.2018.8493207

[21] Mari Carmen Domingo. 2012. An Overview of the Internet of Underwater Things. *Journal of Network and Computer Applications* 35, 6 (2012), 1879–1890. https://doi.org/10.1016/j.jnca.2012.07.012

[22] Adam Dou, Vana Kalogeraki, Dimitrios Gunopulos, Taneli Mielikainen, and Ville H Tuulos. 2010. Misco: A MapReduce Framework for Mobile Systems. In *International Conference on Pervasive Technologies related to Assistive Environments*. ACM, New York, NY, USA, 32. https://doi.org/10.1145/1839294.1839332

[23] Yehia Elkhatib, Barry Porter, Heverson B Ribeiro, Mohamed Faten Zhani, Junaid Qadir, and Etienne Rivière. 2017. On Using Micro-Clouds to Deliver the Fog. *IEEE Internet Computing* 21, 2 (2017), 8–15. https://doi.org/10.1109/MIC.2017.35

[24] Fatemeh Fazel, Maryam Fazel, and Milica Stojanovic. 2011. Random Access Compressed Sensing for Energy-efficient Underwater Sensor Networks. *IEEE Journal on Selected Areas in Communications* 29, 8 (2011), 1660–1670. https://doi.org/10.1109/JSAC.2011.110915

[25] Huber Flores, Naser Hossein Motlagh, Agustin Zuniga, Mohan Liyanage, Monica Passananti, Sasu Tarkoma, Moustafa Youssef, and Petteri Nurmi. 2021. Toward Large-Scale Autonomous Monitoring and Sensing of Underwater Pollutants. *IEEE Internet of Things Magazine* 4 (2021), 40–45. https://doi.org/10.1109/IOTM.0011.2000057

[26] Huber Flores, Agustin Zuniga, Naser Hossein Motlagh, Mohan Liyanage, Monica Passananti, Sasu Tarkoma, Moustafa Youssef, and Petteri Nurmi. 2020. PENGUIN: Aquatic Plastic Pollution Sensing Using AUVs.. In *DroNet@ MobiSys*. ACM, New York, NY, USA, 5–1. https://doi.org/10.1145/3396864.3399704

[27] Huber Flores, Agustin Zuniga, Leonardo Tonetto, Tristan Braud, Pan Hui, Yong Li, Sasu Tarkoma, Mostafa Ammar, and Petteri Nurmi. 2022. Collaboration Stability: Quantifying the Success and Failure of Opportunistic Collaboration. *IEEE Computer Magazine* 55 (2022), 70–81. 10.1109/MC.2021.3112850

[28] Michael Fulton, Jungseok Hong, Md Jahidul Islam, and Junaed Sattar. 2019. Robotic Detection of Marine Litter Using Deep Visual Detection Models. In *International Conference on Robotics and Automation (ICRA)*. IEEE, Montreal, QC, Canada, 5752–5758. https://doi.org/10.1109/ICRA.2019.8793975

[29] Karim Habak, Mostafa Ammar, Khaled A Harras, and Ellen Zegura. 2015. Femto Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge. In *IEEE International Conference on Cloud Computing*. IEEE, New York, NY, USA, 9–16. https://doi.org/10.1109/CLOUD.2015.12

[30] John Heidemann, Milica Stojanovic, and Michele Zorzi. 2012. Underwater sensor networks: applications, advances and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370, 1958 (2012), 158–175. https://doi.org/10.1098/rsta.2011.0214

[31] Eduardo Iscar, Corina Barbalata, Nicholas Goumas, and Matthew Johnson-Roberson. 2018. Towards Low Cost, Deep Water AUV Optical Mapping. In *OCEANS MTS/IEEE Charleston*. IEEE, Charleston, SC, USA, 1–6. https://doi.org/10.1109/OCEANS.2018.8604772

[32] Mohammad Jahanbakht, Wei Xiang, Lajos Hanzo, and Mostafa Rahimi Azghadi. 2021. Internet of Underwater Things and Big Marine Data Analytics: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 904–956. https://doi.org/10.1109/COMST.2021.3053118

[33] Imad Jawhar, Nader Mohamed, Jameela Al-Jaroodi, and Sheng Zhang. 2013. An Efficient Framework for Autonomous Underwater Vehicle Extended Sensor Networks for Pipeline Monitoring. In *IEEE International Symposium on Robotic and sensors Environments (ROSE)*. IEEE, Washington, DC, USA, 124–129. https://doi.org/10.1109/ROSE.2013.6698430

[34] Leilei Jin and Hong Liang. 2017. Deep Learning for Underwater Image Recognition in Small Sample Size Situations. In *OCEANS 2017-Aberdeen*. IEEE, Aberdeen, UK, 1–4. https://doi.org/10.1109/OCEANSE.2017.8084645

[35] Chien-Chi Kao, Yi-Shan Lin, Geng-De Wu, and Chun-Ju Huang. 2017. A Comprehensive Study on the Internet of Underwater Things: Applications, Challenges, and Channel Models. *Sensors* 17, 7 (2017), 1477. https://doi.org/10.3390/s17071477

[36] Mohammad-Ali Khalighi, Chadi Gabriel, Tasnim Hamza, Salah Bourennane, Pierre Leon, and Vincent Rigaud. 2014. Underwater Wireless Optical Communication; Recent Advances and Remaining Challenges. In *International Conference on Transparent Optical Networks (ICTON)*. IEEE, Graz, Austria, 1–4. https://doi.org/10.1109/ICTON.2014.6876673

[37] Ruhul Khalil, Mohammad Babar, Tariqullah Jan, and Nasir Saeed. 2020. Towards the Internet of Underwater Things: Recent Developments and Future Challenges. *IEEE Consumer Electronics Magazine* 10 (2020), 32–37. https://doi.org/10.1109/MCE.2020.2988441

[38] Emmanouil Koukoumidis, Dimitrios Lymberopoulos, Karin Strauss, Jie Liu, and Doug Burger. 2011. Pocket Cloudlets. In *ACM SIGARCH Computer Architecture News*, Vol. 39. ACM, New York, NY, USA, 171–184. https://doi.org/10.1145/1961296.1950387

[39] Eemil Lagerspetz, Huber Flores, Niko Mäkitalo, Pan Hui, Petteri Nurmi, Sasu Tarkoma, Andrea Passarella, Jörg Ott, Peter Reichl, Marco Conti, et al. 2018. Pervasive Communities in the Internet of People. In *International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Vol. 1. IEEE, Athens, Greece, 40–45. https://doi.org/10.1109/PERCOMW.2018.8480273

[40] Eemil Lagerspetz, Xin Li, Jonatan Hamberg, Huber Flores, Petteri Nurmi, Nigel Davis, and Sumi Helal. 2019. Pervasive Data Science on the Edge. *IEEE Pervasive Computing* 18 (2019), 40–49. https://doi.org/10.1109/MPRV.2019.2925600

[41] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network* 32, 1 (2018), 96–101. https://doi.org/10.1109/MNET.2018.1700202

[42] Kaiqiang Lin, Tong Hao, Wuan Zheng, and Wenchao He. 2019. Analysis of LoRa Link Quality for Underwater Wireless Sensor Networks: A Semi-empirical Study. In *IEEE Asia-Pacific Microwave Conference (APMC)*. IEEE, Singapore, 120–122. https://doi.org/10.1109/APMC46564.2019.9038666

[43] Mohan Liyanage, Farooq Dar, Rajesh Sharma, and Huber Flores. 2021. GEESE: Edge computing enabled by UAVs. *Pervasive and Mobile Computing* 72 (2021), 101340. https://doi.org/10.1016/j.pmcj.2021.101340

[44] Rosalia Maglietta, Rocco Caccioppoli, Emanuele Seller, Stefano Bellomo, Francesca Cornelia Santacesaria, Roberto Colella, Giulia Cipriano, Ettore Stella, Karin Hartman, Carmelo Fanizza, et al. 2020. Convolutional Neural Networks for Risso's Dolphins Identification. *IEEE Access* 8 (2020), 80195–80206. https://doi.org/10.1109/ACCESS.2020.2990427

[45] Ricardo Martins, Joao Borges de Sousa, Renato Caldas, Chiara Petrioli, and John Potter. 2014. SUNRISE project: Porto University Testbed. In *Underwater Communications and Networking (UComms)*. IEEE, Sestri Levante, Italy, 1–5. https://doi.org/10.1109/UComms.2014.7017143

[46] Jonathan McChesney, Nan Wang, Ashish Tanwer, Eyal de Lara, and Blesson Varghese. 2019. DeFog: Fog Computing Benchmarks. In *ACM/IEEE Symposium on Edge Computing*. ACM, New York, NY, USA, 47–58. https://doi.org/10.1145/3318216.3363299

[47] David K Mellinger and Christopher W Clark. 2006. MobySound: A Reference Archive for Studying Automatic Recognition of Marine Mammal Sounds. *Applied Acoustics* 67, 11-12 (2006), 1226–1242. https://doi.org/10.1016/j.apacoust.2006.06.002

[48] Roberto Morabito. 2016. A Performance Evaluation of Container Technologies on Internet of Things Devices. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, San Francisco, CA, USA, 999–1000. https://doi.org/10.1109/INFCOMW.2016.7562228

[49] Nils Morozs, Paul D Mitchell, and Roee Diamant. 2020. Scalable Adaptive Networking for the Internet of Underwater Things. *IEEE Internet of Things Journal* 7, 10 (2020), 10023–10037. https://doi.org/10.1109/JIOT.2020.2988621

[50] Yiming Peng, Peipei Wu, Amina T Schartup, and Yanxu Zhang. 2021. Plastic Waste Release Caused by COVID-19 and its Fate in the Global Ocean. *Proceedings of the National Academy of Sciences* 118, 47 (2021), 1–6. https://doi.org/10.1073/pnas.2111530118

[51] AA Periola, AA Alonge, and KA Ogudo. 2020. Architecture and System Design for Marine Cloud Computing Assets. *Comput. J.* 63, 6 (2020), 927–941. https://doi.org/10.1093/comjnl/bxz169

[52] Tie Qiu, Zhao Zhao, Tong Zhang, Chen Chen, and CL Philip Chen. 2019. Underwater Internet of Things in Smart Ocean: System Architecture and Open Issues. *IEEE transactions on industrial informatics* 16, 7 (2019), 4297–4307. https://doi.org/10.1109/TII.2019.2946618

[53] Marko Radeta, Nuno J Nunes, Dinarte Vasconcelos, and Valentina Nisi. 2018. Poseidon-Passive-Acoustic Ocean Sensor for Entertainment and Interactive Data-Gathering in Opportunistic Nautical-Activities. In *Designing Interactive Systems Conference*. ACM, New York, NY, USA, 999–1011. https://doi.org/10.1145/3196709.3196752

[54] Marko Radeta, Miguel Ribeiro, Dinarte Vasconcelos, Jorge Lopes, Michael Sousa, João Monteiro, and Nuno Jardim Nunes. 2019. SeaMote-Interactive Remotely Operated Apparatus for Aquatic Expeditions. In *IFIP Conference on Human-Computer Interaction*. Springer, New York, NY, USA, 237–248. https://doi.org/10.1007/978-3-030-29387-1_14

[55] Marko Radeta, Miguel Ribeiro, Dinarte Vasconcelos, Hildegardo Noronha, and Nuno Jardim Nunes. 2020. LoRaquatica: Studying Range and Location Estimation Using LoRa and IoT in Aquatic Sensing. In *International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Austin, TX, USA, 1–6. https://doi.org/10.1109/PerComWorkshops48775.2020.9156088

[56] Marko Radeta, Agustin Zuniga, Naser Hossein Motlagh, Mohan Liyanage, Ruben Freitas, Moustafa Youssef, Sasu Tarkoma, Huber Flores, and Petteri Nurmi. 2022. Deep Learning and the Oceans. *Computer: A publication of the IEEE Computer Society* 55 (2022), 39–50. https://doi.org/10.1109/MC.2022.3143087

[57] Christopher Rasmussen, Jiayi Zhao, Danielle Ferraro, and Arthur Trembanis. 2017. Deep Census: AUV-based Scallop Population Monitoring. In *International Conference on Computer Vision Workshops*. IEEE, Venice, Italy, 2865–2873. https://doi.org/10.1109/ICCVW.2017.338

[58] Nasir Saeed, Abdulkadir Celik, Tareq Y Al-Naffouri, and Mohamed-Slim Alouini. 2019. Underwater Optical Wireless Communications, Networking, and Localization: A Survey. *Ad Hoc Networks* 94 (2019), 101935. https://doi.org/10.1016/j.adhoc.2019.101935

[59] Arjuna Sathiaseelan, Adisorn Lertsinsrubtavee, Adarsh Jagan, Prakash Baskaran, and Jon Crowcroft. 2016. Cloudrone: Micro Clouds in the Sky. In *Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, New York, NY, USA, 41–44. https://doi.org/10.1145/2935620.2935625

[60] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing* 8, 4 (2009), 14–23. https://doi.org/10.1109/MPRV.2009.82

[61] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia & Analgesia* 126, 5 (2018), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

[62] Basem Shihada, Osama Amin, Christopher Bainbridge, Seifallah Jardak, Omar Alkhazragi, Tien Khee Ng, Boon Ooi, Michael Berumen, and Mohamed-Slim Alouini. 2020. Aqua-Fi: Delivering Internet Underwater Using Wireless Optical Networks. *IEEE Communications Magazine* 58, 5 (2020), 84–89. https://doi.org/10.1109/MCOM.001.2000009

[63] Michelle Sigler. 2014. The Effects of Plastic Pollution on Aquatic Wildlife: Current Situations and Future Solutions. *Water, Air, & Soil Pollution* 225, 11 (2014), 1–9. https://doi.org/10.1007/s11270-014-2184-6

[64] Dan A Smale, Gary A Kendrick, Euan S Harvey, Timothy J Langlois, Renae K Hovey, Kimberly P Van Niel, Kris I Waddington, Lynda M Bellchambers, Matthew B Pember, Russ C Babcock, et al. 2012. Regional-Scale Benthic Monitoring for Ecosystem-based Fisheries Management (EBFM) Using an Autonomous Underwater Vehicle (AUV). *ICES Journal of Marine Science* 69, 6 (2012), 1108–1118. https://doi.org/10.1093/icesjms/fss082

[65] Swarnalatha Srinivas, P Ranjitha, R Ramya, and G Kumar Narendra. 2012. Investigation of Oceanic Environment Using Large-Scale UWSN and UANETs. In *International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, Shanghai, China, 1–5. https://doi.org/10.1109/WiCOM.2012.6478552

[66] Milica Stojanovic and James Preisig. 2009. Underwater Acoustic Communication Channels: Propagation Models and Statistical Characterization. *IEEE Communications Magazine* 47, 1 (2009), 84–89. https://doi.org/10.1109/MCOM.2009.4752682

[67] Kay Strama, Daniel Weber, and Helge Renkewitz. 2021. Evaluation of Wifi Data Transmission Algorithms for Short Distance Underwater Communication. In *OCEANS 2021: San Diego–Porto*. IEEE, San Diego, CA, USA, 1–6. https://doi.org/10.23919/OCEANS44145.2021.9705847

[68] Jaime W Thompson, Victoria H Zero, Lori H Schwacke, Todd R Speakman, Brian M Quigley, Jeanine SM Morey, and Trent L McDonald. 2019. finFindR: Computer-assisted Recognition and Identification of Bottlenose Dolphin Photos in R. *bioRxiv* 598 (2019), 1–18. https://doi.org/10.1101/825661

[69] Nguyen H Tran, Wei Bao, Albert Zomaya, Nguyen Minh NH, and Choong Seon Hong. 2019. Federated Learning Over Wireless Networks: Optimization Model Design and Analysis. In *IEEE INFOCOM Conference on Computer Communications*. IEEE, Paris, France, 1387–1395. https://doi.org/10.1109/INFOCOM.2019.8737464

[70] Thiago Lara Vasques, Pedro Moura, and Aníbal de Almeida. 2019. A Review on Energy Efficiency and Demand Response with Focus on Small and Medium Data Centers. *Energy Efficiency* 12 (2019), 1–30. https://doi.org/10.1007/s12053-018-9753-2

[71] R Venkatesan, VR Shamji, G Latha, Simi Mathew, RR Rao, Arul Muthiah, and MA Atmanand. 2013. In Situ Ocean Subsurface Time-Series Measurements from OMNI Buoy Network in the Bay of Bengal. *Current Science* 1, 1 (2013), 1166–1177. https://www.jstor.org/stable/24092396

[72] Sébastien Villon, David Mouillot, Marc Chaumont, Emily S Darling, Gérard Subsol, Thomas Claverie, and Sébastien Villéger. 2018. A Deep Learning Method for Accurate and Fast Identification of Coral Reef Fishes in Underwater Images. *Ecological informatics* 48 (2018), 238–244. https://doi.org/10.1016/j.ecoinf.2018.09.007

[73] Jianxiong Wan, Xiang Gui, Shoji Kasahara, Yuanyu Zhang, and Ran Zhang. 2018. Air Flow Measurement and Management for Improving Cooling and Energy Efficiency in Raised-Floor Data Centers: A Survey. *IEEE Access.* 6 (2018), 48867–48901. https://doi.org/10.1109/ACCESS.2018.2866840

[74] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221. https://doi.org/10.1109/JSAC.2019.2904348

[75] Jun-Ichiro Watanabe, Yang Shao, and Naoto Miura. 2019. Underwater and Airborne Monitoring of Marine Ecosystems and Debris. *Journal of Applied Remote Sensing* 13, 4 (2019), 044509. https://doi.org/10.1117/1.JRS.13.044509

[76] Hannah J Williams, Lucy A Taylor, Simon Benhamou, Allert I Bijleveld, Thomas A Clay, Sophie de Grissac, Urška Demšar, Holly M English, Novella Franconi, Agustina Gómez-Laich, et al. 2020. Optimizing the Use of Biologgers for Movement Ecology Research. *Journal of Animal Ecology* 89, 1 (2020), 186–206. https://doi.org/10.1111/1365-2656.13094

[77] Jie Xu, Lixing Chen, and Pan Zhou. 2018. Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks. In *IEEE Conference on Computer Communications- INFOCOM*. IEEE, IEEE, Honolulu, HI, USA, 207–215. https://doi.org/10.1109/INFOCOM.2018.8485977

[78] Orhan Yaman, Turker Tuncer, and Beyda Tasar. 2021. DES-Pat: A Novel DES Pattern-based Propeller Recognition Method Using Underwater Acoustical Sounds. *Applied Acoustics* 175 (2021), 107859. https://doi.org/10.1016/j.apacoust.2020.107859

[79] Shanhe Yi, Cheng Li, and Qun Li. 2015. A Survey of Fog Computing: Concepts, Applications and Issues. In *Proceedings of the Workshop on Mobile Big Data*. ACM, New York, NY, United States, 37–42. https://doi.org/10.1145/2757384.2757397

[80] Zhaoquan Zeng, Shu Fu, Huihui Zhang, Yuhan Dong, and Julian Cheng. 2016. A Survey of Underwater Optical Wireless Communications. *IEEE Communications Surveys & Tutorials* 19, 1 (2016), 204–238. https://doi.org/10.1109/COMST.2016.2618841

[81] Zhangbing Zhou, Beibei Yao, Riliang Xing, Lei Shu, and Shengrong Bu. 2015. E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things. *IEEE Sensors Journal* 16, 11 (2015), 4072–4082. https://doi.org/10.1109/JSEN.2015.2437904