Socio-Environmental Sensor Networks for Community Sensing

by

Andrés Rico Medina

B.Sc., Instituto Tecnológico y de Estudios Superiores de Monterrey (2019)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning in partial fulfillment of the requirements for the degree of Master of Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY May 2022

© Massachusetts Institute of Technology 2022. All rights reserved.
Socio-Environmental Sensor Networks for Community Sensing

by

Andrés Rico Medina

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on May 13, 2022, in partial fulfillment of the requirements for the degree of
Master of Science

Abstract

We are living in a time of extraordinary urban changes. Research has shown that cities can bring economic wealth and improved quality of life by fostering diverse economies, dense knowledge exchanges, and efficient district performance. However, it is also true that scientists have associated cities with crowding, segregation, environmental degradation, and other significant challenges. Sensors, Data, and Artificial Intelligence can lead to a better understanding of urban settings and their challenges by providing opportunities for insight into their social and environmental performance. Many of these sensing initiatives are carried out in a top-down fashion. Top-down sensing generates datasets that capture large-scale patterns across populations. This data could be complemented by bottom-up community-based approaches that capture more granular information emerging from the specific needs of individuals. Through a series of case studies, this thesis illustrates how to use a variety of community-scale sensor and machine intelligence implementations to measure aspects of socio-environmental cycles that emerge in different urban and environmental contexts. These studies explore possibilities for providing communities with access to localized information about socio-environmental systems that, if fully deployed, could enable bottom-up transformation of collective behavior, policies, and infrastructure to address the great challenges that future cities will face.

Thesis Supervisor: Kent Larson
Title: Principal Research Scientist
Socio-Environmental Sensor Networks for Community Sensing

by

Andrés Rico Medina

This thesis has been reviewed and approved by the following committee member:

Kent Larson ........................................................... Thesis Supervisor
Director, City Science Group MIT Media Lab
Socio-Environmental Sensor Networks for Community Sensing
by
Andrés Rico Medina

This thesis has been reviewed and approved by the following committee member:

Neil Gershenfeld ....................................................... Thesis Reader
Director, MIT’s Center for Bits and Atoms
Socio-Environmental Sensor Networks for Community Sensing
by
Andrés Rico Medina

This thesis has been reviewed and approved by the following committee member:

Ally Huang
Thesis Reader
Co-developer, BioBits
BioBits Program Lead at miniPCR bio
Acknowledgments

This thesis is a sum of support, conversations, lessons, ideas, and time. Thank you to all the amazing people that have been part of this in one way or another.

Thank you Kent for the opportunity of being a student in the City Science Group. I have learned so much from your leadership skills and vision about how we can build better cities. I’m very much looking forward to what the next phases in the lab will bring.

Thank you to my readers, Ally and Neil, I greatly appreciate your advice and hold your work as reference for how we can improve our society through democratic technological development.

Jason, Carson, Yasushi and Fadel, thank you for being such amazing friends and mentors and helping me to navigate the complexities of life and graduate studies.

Thank you to the entire City Science Team! It’s a great privilege to get to work alongside such a talented group of people. I have learnt a lot from everyone and have very much enjoyed being a part of the lab for the last couple of years.

Thank you to my professors, to the HTGAA team and to all of my colleagues throughout the Media Lab. Your perspectives and ideas are of enormous influence to my work.

Thank you to the ones that have always been there cheering and supporting me throughout this journey; Andy, Ana, Mora, Nico, and Guada.

Naroa, mi mayor inspiración, gracias por tanto. ¡Maite zaitut maitia!
Mamá, Papá y Sebastián, todo lo que soy se los debo a ustedes. ¡Los quiero mucho!
## Contents

1 Socio-Environmental Community Sensing ............................................................... 29
    1.1 The Challenge of Cities ....................................................................................... 29
    1.2 Understanding Cities Through Data ................................................................. 30
        1.2.1 A World of Sensors and Data .................................................................. 30
        1.2.2 City IoT and Data Systems ..................................................................... 30
        1.2.3 Today’s Top-Down City Sensing Systems ................................................. 31
    1.3 Cities as Socio-Environmental Systems ............................................................. 32
    1.4 Community Sensing ......................................................................................... 32
    1.5 Contribution ....................................................................................................... 33
    1.6 Methods ............................................................................................................ 34

2 Whispers Of The Mountain: Opportunistic Sensing ...................................................... 35
    2.1 Overture ............................................................................................................. 35
    2.2 Background ....................................................................................................... 36
        2.2.1 Context ...................................................................................................... 36
        2.2.2 Related Work ......................................................................................... 37
        2.2.3 Contribution ............................................................................................ 38
    2.3 System Design ................................................................................................... 38
        2.3.1 Hardware .................................................................................................. 38
        2.3.2 Software .................................................................................................. 47
    2.4 Validation .......................................................................................................... 49
        2.4.1 Deployment .............................................................................................. 49
        2.4.2 Data ......................................................................................................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2 Data</td>
<td>88</td>
</tr>
<tr>
<td>4.5 Discussion</td>
<td>91</td>
</tr>
<tr>
<td>4.5.1 Limitations &amp; Future Work</td>
<td>91</td>
</tr>
<tr>
<td>4.5.2 Conclusion</td>
<td>93</td>
</tr>
<tr>
<td>5 MAIA: Distributed Citizen Science</td>
<td>101</td>
</tr>
<tr>
<td>5.1 Overture</td>
<td>101</td>
</tr>
<tr>
<td>5.2 Background</td>
<td>102</td>
</tr>
<tr>
<td>5.2.1 Context</td>
<td>102</td>
</tr>
<tr>
<td>5.2.2 Related Work</td>
<td>103</td>
</tr>
<tr>
<td>5.2.3 Contribution</td>
<td>104</td>
</tr>
<tr>
<td>5.3 System Design</td>
<td>105</td>
</tr>
<tr>
<td>5.3.1 Hardware</td>
<td>105</td>
</tr>
<tr>
<td>5.3.2 Software</td>
<td>111</td>
</tr>
<tr>
<td>5.4 Validation</td>
<td>113</td>
</tr>
<tr>
<td>5.4.1 Deployment</td>
<td>113</td>
</tr>
<tr>
<td>5.4.2 Data</td>
<td>116</td>
</tr>
<tr>
<td>5.5 Discussion</td>
<td>121</td>
</tr>
<tr>
<td>5.5.1 Limitations &amp; Future Work</td>
<td>121</td>
</tr>
<tr>
<td>5.5.2 Conclusion</td>
<td>123</td>
</tr>
<tr>
<td>6 JettSen: Consensus Building Infrastructure</td>
<td>125</td>
</tr>
<tr>
<td>6.1 Overture</td>
<td>125</td>
</tr>
<tr>
<td>6.2 Background</td>
<td>126</td>
</tr>
<tr>
<td>6.2.1 Context</td>
<td>126</td>
</tr>
<tr>
<td>6.2.2 Related Work</td>
<td>127</td>
</tr>
<tr>
<td>6.2.3 Contribution</td>
<td>128</td>
</tr>
<tr>
<td>6.3 System Design</td>
<td>129</td>
</tr>
<tr>
<td>6.3.1 Hardware</td>
<td>129</td>
</tr>
<tr>
<td>6.3.2 Software</td>
<td>135</td>
</tr>
<tr>
<td>6.4 Validation</td>
<td>139</td>
</tr>
</tbody>
</table>
6.4.1 Deployment ........................................... 139
6.4.2 Data .................................................. 141
6.5 Discussion ............................................. 145
  6.5.1 Limitations & Future Work ....................... 145
  6.5.2 Conclusion ......................................... 146

7 Conclusions ................................................. 149
List of Figures

2-1 From left to right; Helmet module with microphone input and fiber optic light interface, Right ski module with connection to piezoelectric array and environmental sensors, and left ski module with GPS unit. 39

2-2 Full system architecture. The image shows the three main modules and the variables that each of the modules is responsible for. All of the modules communicate with each other through wireless radio signals. 39

2-3 Assembled helmet module with radio communication, voice input and peripheral feedback user interface (UI). 41

2-4 RM electronic diagram. The system has a central unit, a Teensy 3.2 micro controller and uses peripherals to access the piezoelectric array and external sensors. 42

2-5 Picture of both sections of the embedded ski sensor array. A) shows the front end of the ski. Labels 1–3 correspond to the front three piezoelectric sub-arrays. Label 4 shows the prototype area that contains all the embedded electronic circuits for signal pre-processing. Label 5 marks the waterproof plugs for accessing the sensor’s signals. B) shows the rear end of the ski with Label 1 indicating the location of the circuits for signal processing and Labels 2–3 marking the two rear sensor sub-arrays. 44
2-6 Electronic schematic for the individual sensor array. Labels T1–T5 show the five ceramic piezoelectric transducers connected in parallel. The signal is grounded through a 220kΩ resistor and is attenuated with a voltage divider of 1MΩ and 68kΩ. The output signal connection is the terminal that can be probed to obtain the signal from the entire sensor.

2-7 Electronic schematic for the complete piezoelectric array and its connections to the teensy microcontroller. The array is composed of five individual piezoelectric arrays containing five transducers each. The complete design has 25 transducers, which allow us to obtain measurements on different segments of the ski. Labels P1–P5 show the five sets of piezoelectric arrays, each of which follows the schematic of Figure 2-6. A 3.7V LiPo (Lithium-ion Polymer) battery powers the system. A commercial micro-controller board (Teensy 3.2) with a built-in SD card module measures and logs independent transducer signals.

2-8 Manufacturing sequence of the embedded sensors. A) shows how the piezoelectric transducers were aligned and attached to the ski surface. B) shows the layout for the circuit traces. Note that all components share a common digital ground reference. We place a resin layer on top of the sections that cross over different arrays to avoid short circuits. C) shows the pull-down resistor and voltage divider circuits composed of SMD components directly soldered onto the copper traces. D) shows the transducers after the final coating layer was applied to make the system waterproof.

2-9 LM electronic diagram. The system has an Adafruit Adalogger as its central unit and uses serial and SPI communication to access GPS and Radio Modules.

2-10 Prototype deployment on real mountain and use case conditions.
2-11 Frequency Spectrogram. The Figure shows the transformation of the signals coming from the five sub-arrays of the system into the frequency domain. .......................... 51

2-12 Fused data set visualization with piezoelectric sensor, pressure and geo-location data points plotted out on map. ......................... 52

2-13 Sensor fusion plots on multiple mountain runs. The visualizations help to see the insights that the system has the potential to give to stakeholders. .......................... 53

3-1 Illustration of sensing and communication nodes placed on fishing boats. The sensing module communicates with the communications module during each surfacing. Communication nodes will relay data onto the global network. Nodes can then be used to form a star network topology based on multiple fishing boats. .......................... 59

3-2 Sensor node component description with communication and power lines connections. .......................... 60

3-3 Illustration of mechanical and electrical implementation of the sensor node design. A) Electronic components mounted on acrylic plate. B) Acrylic plate inserted into cylinder body. .......................... 62

3-4 Screenshot of server data visualization on mobile phone. ............. 63

3-5 Finite state machine implemented on sensing node with three sequential states. .......................... 65

3-6 Data plot for 20 minute sensor deployment. Key variables for this experiment are depth and light. Plot shows testing of depth sensor at two different depths and intermittent occlusion of light sensor. .......................... 66

3-7 River deployment showing power cycling operation and intermittent data logging. Two different locations and depths were measured within a span of 18 minutes. .......................... 67
3-8 Collected data for a 5-minute experimental log deployed in a shallow water shoreline. The purpose of the experiment was to validate water acidity readings and the fusion of multiple sensors. The current pH sensor shows high precision but low accuracy, meaning that it can be used for relative measurements and not absolute ones.

3-9 Log from second ocean experiment. Experimented lasted 90 minutes. Data was sampled at two distinct locations and depths to evaluate the system’s mechanical and electrical performance. The graph shows successful operation of the device along with possible applications of collected data.

4-1 Full system architecture showing the sensor, data storage, machine intelligence, visualization, and actuation modules. The sensor module sends time-stamped sensor values every five minutes to the storage module. The machine intelligence module queries the storage module to extract seven days’ data to train a new model every week. The machine intelligence module locally stores classification models and sends live predictions back into the storage module. Classified states can also be sent to visualization and actuation modules to inform users about the environment or autonomously modify it.

4-2 Image of a sensor module with casing. The Figure shows the $CO_2$ sensor module connected to the WiFi-enabled microcontroller. We also show ventilation for the $CO_2$ sensor as well as an outlet for exposing the PIR sensor. The sensor can be mounted on any flat wall and has an outlet for powering the device through a micro USB port.
Hybrid machine learning implementation. Data is acquired for seven days (acquisition range can be varied depending on application) and saved on the data storage module. On the seventh day of acquisition, the machine intelligence module queries servers and uses spectral clustering to create clusters corresponding to different room activity states (e.g., high, medium, low). The clusters are then used to label a new data set, called DA, and fed for training into a Recurrent Neural Network that takes sensor data as inputs and clustered labels as its training objectives. Creating a new RNN from new spectral clustering results takes between 5 - 15 minutes (training time depends on the time range selected, our implementation uses seven days as the default time range). The new model is then used for live classification of room states. The process repeats every seven days to update new behaviors or activities within the space. The updating range can be varied depending on specific uses or rooms. When the system is in the live classification state, data is relayed into the data acquisition state to restart the process and readjust models.

Clustering results comparing K-means clustering and spectral clustering. We used five clusters as our standard to visualize the key differences between the two algorithms. Each color indicates the cluster to which the data point belongs. Values for PIR and $CO_2$ are normalized before being used by each of the algorithms.

Recurrent Neural Network architecture. The network is composed of 8 layers. The input takes two inputs corresponding to sensor variables, hidden layers are divided between LSTM’s and regular feed forward layers. The output layer has 5 neurons corresponding to the 5 clusters created by the spectral clustering section of the system.
4-6 Spectral clustering results for four weeks' worth of data. The plots visualize how clusters are properly created following sensor spikes and activation activity. We show a plot with CO$_2$ readings, one with PIR readings, and a final one that plots CO$_2$ vs. PIR to prove that clusters are well defined. It is relevant to note that differences in color scheme for each plot are due to the way that the spectral clustering algorithm assigns cluster centroids.

4-7 Accuracy plots for validation and test sets along 50 epochs of training for models generated from the office and classroom sensors throughout four weeks of experimentation. We can see that all models manage to get accuracies above 87% by their final training epoch.

4-8 Plotted results from the live classifications made by a trained model running on real-time data and control results for saved data clustered by spectral clustering. We can visualize that the machine intelligence module can properly classify as expected. Data density is lower for the live classification because live classification is carried out every 15 minutes as opposed to the 5-minute interval used when saving the data used by the spectral clustering algorithm.

5-1 Modular "plug and play" architecture allows for flexible design of experiments and growth setups. Modules are connected to shared power and communication lines and stacked accordingly to desired function and stimulation. Power, control, and temperature modules are considered the minimum assembly. Significant changes, dependant on cells and experiments, occur in the growth and multi-functional modules.

5-2 Base modules printed on different materials with PCB’s, growth medium and accessories mounted on them. The image illustrates the flexibility of the module and its capability to be adapted for the different functions that the reactor is required to perform.

5-3 Power PCB module placed on PLA base module.
5-4  Mother PCB module with WIFI and Bluetooth capabilities placed on PLA base module.

5-5  Temperature PCB module placed on PLA base module. The illustration shows the Peltier accessory that can be attached to module to hold the Peltier material on top of the device when connected.

5-6  Growth module assembly. The structure can house different types of cultures with different growth requirements. Future versions of the device will validate liquid and microfluidic capabilities.

5-7  Temperature PCB module placed on PLA base module. The illustration shows the Peltier accessory that can be attached to module to hold the Peltier material on top of the device when connected.

5-8  MAIAs setup for blue light stimulation experiments.

5-9  Students using the reactor to set up different experiments on the MAIA platform.

5-10 Experiment 1 results. Six out of seven reactors worked properly. Cell growth and gene expression happened as hypothesized.

5-11 Experiment 2 results.

5-12 A) Sketches of modular bioreactor modules for production of bio-tinted silk. B) Biocementation module prototypes. The designs include improved ventilation modules and a tool carrying module for portable experimentation. C) Bioreactor adaptation for testing temperature sensing bacteria grown on textiles.

6-1  Mobile Sensor Fusion System Architecture. Four sensor modules feed data into central unit. The central unit processes data, adds timestamps and generates three separate files labeled by a UUID. Data is later extracted and analyzed on an external computer.
6-2 System Implementation. Installation of the mobile sensor fusion system on prototype bicycle provided by Panasonic Cycle Technology. As the system components are placed at separate locations, cable lines run through the three main bike frame tubes (top, down and seat tubes) and connect directly to the central unit, placed in a case behind the bike’s seat.

6-3 Power Supply and Data Transmission Connections. Diagram shows electric connections and flow of information for the systems four sensor modules and the central unit. Note that all devices have a common ground provided by the bicycle’s battery pack, the diagram does not include ground connections to make it more simple to read. Red lines represent power (V+) connections and blue lines are all simplified data transmission lines.

6-4 Normalized elbow method analysis for determination of optimal clustering parameters for each one of the matrix variations (x acceleration, fusion and multiple trip). Results indicate that seven is the optimal number of clusters for each one of the matrix variations. The point on each curve was chosen by analyzing the moment at which the error rate change decreases for the first time.

6-5 Sensor variables Knowledge Impact. The diagram allows us to visualize the way in which different trip layers are related to the different levels of abstraction. The different layers of abstraction allow for exploration of highly specific and technical issues and their relationship with higher level values that the community might hold. Focusing on a specific area within the diagram can yield entirely different knowledge, even if the same data source and algorithm is used, as can be seen in section 4.

6-6 Single Trip X-Acceleration Clustering. The graph plots normalized acceleration data on the y axis and relative time (each unit represents 0.5 ms) on the x axis. The K-means algorithm is capable of effectively clustering different acceleration points.
6-7 Single Trip Fusion Data Clustering. The graph plots normalized data for 3 axis acceleration, speed, torque, temperature, light, humidity and pressure on the $y$ axis and relative time (each unit represents 0.5 ms) on the $x$ axis. The K-means algorithm is capable of effectively clustering different instances in a single trip in a sequential manner.

6-8 Multiple Trip Fusion Data Clustering. The graph plots normalized data for 3 axis acceleration, speed, torque, temperature, light, humidity and pressure for 16 different trips on the $y$ axis and relative time (fitted to start from zero) on the $x$ axis. The K-means algorithm is capable of effectively clustering different trips in a sequential manner.
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Ski sensor fusion system stored variables and units.</td>
<td>40</td>
</tr>
<tr>
<td>2.2</td>
<td>Helmet’s voice interface commands and control actions.</td>
<td>41</td>
</tr>
<tr>
<td>3.1</td>
<td>Saved Data Structure: collected variables, selected sensors, and data units.</td>
<td>64</td>
</tr>
<tr>
<td>4.1</td>
<td>Performance benchmark for state of the art activity and occupancy classification systems.</td>
<td>91</td>
</tr>
<tr>
<td>5.1</td>
<td>Available commands processed by mother module to control light and temperature states.</td>
<td>112</td>
</tr>
<tr>
<td>5.2</td>
<td>Experimental setup and results for the first experiment to evaluate cell ground and gene expression on bioreactor. Light value key: B = Blue, G = Green, R = Red, N = No Light, E = Exposed (white light). Recator key: M = MAIA reactor, O = Outside (no incubator), I = Commercial incubator. Temperatures are expressed in degrees Celsius.</td>
<td>117</td>
</tr>
<tr>
<td>5.3</td>
<td>Experimental setup and results for the second experiment to evaluate cell ground and gene expression on bioreactor carried out by students. Light value key: B = Blue, G = Green, R = Red, N = No Light, E = Exposed (white light). Recator key: M = MAIA reactor, O = Outside (no incubator), I = Commercial incubator. Temperatures are expressed in degrees Celsius.</td>
<td>118</td>
</tr>
<tr>
<td>6.1</td>
<td>Bicycle State Variable Description and Units.</td>
<td>132</td>
</tr>
<tr>
<td>6.2</td>
<td>Environmental State Variable Description and Units.</td>
<td>132</td>
</tr>
</tbody>
</table>
6.3 Geo-spatial Variable Description and Units. . . . . . . . . . . . . . . 132
6.4 Selected Variables for Clustering Analysis . . . . . . . . . . . . . . . 138
Chapter 1

Socio-Environmental Community Sensing

1.1 The Challenge of Cities

In 2015, the United Nations defined inclusive, safe, resilient, and sustainable human settlements as one of its Sustainable Development Goals (SDGs) [129]. In establishing this goal, the UN acknowledges that while existing urban systems have produced significant societal progress, they did so with high social and environmental costs.

Research has shown that cities can bring economic wealth and improved quality of life by fostering stable economies of scale with dense knowledge exchanges and efficient district performance [13,14,60]. However, it is also true that scientists have associated urbanization with crowding, segregation, and environmental degradation, amongst other adverse effects [13,19,55]. Therefore, creating sustainable, livable, and equitable cities and communities requires a radical transformation in the social and technological structure that urban landscapes currently have.

Urban growth in the future is expected to follow three primary patterns: (1) informal settlements, (2) modernization of existing cities, and (3) the creation of entirely new cities [23,39,67,68,176]. In contrast to prior periods of rapid urbanization, where environmental and social impacts were of minimal concern, the present moment requires significant attention to the potential negative impacts that might be caused
by the growth of cities. Some technology development has shifted its focus towards "rurbanization" as a way of reversing negative forces of urbanization [24]. Other strategies emphasize new tools and research frameworks enabled by advances in digital technologies to optimize a better balance between the benefits to city dwellers and the social and environmental costs. This thesis explores possibilities for providing communities with access to bottom-up sensing and machine intelligence infrastructure that complements today’s systems.

1.2 Understanding Cities Through Data

1.2.1 A World of Sensors and Data

The 1990s brought the first mentions of the terms "Ubiquitous Computing" and the "Internet of Things" along with rosy visions of the future that might be enabled by networks of sensors and actuators [56, 161, 167, 185]. The subsequent decades have produced enormous innovations in Ubiquitous Computing (UbiComp) and the Internet of Things (IoT). The fields mainly study how large networks of devices can be embedded into our social and environmental fabrics to deeply understand health, behavior, performance, amongst other aspects of our lives [185].

Decades ago, scientists were aware that computing power would continue to increase as prices and size of chips continued to decrease at exponential rates [123]. This would, in turn, lead to computation being embedded, interconnected, and ingrained into our everyday lives [58]. Such advancements have made it possible to have more than 20 billion connected devices around the world as of 2020, in the form of smart thermostats, phones, and wearables amongst others. The total number of these devices is expected to increase to 50 billion by 2030 [117].

1.2.2 City IoT and Data Systems

Connected devices have been introduced across many disciplines, and the operation and study of cities is no exception. Aided by novel artificial intelligence techniques,
UbiComp practitioners have deployed sophisticated IoT sensor networks across cities in an effort to bring a higher quality of life and sustainability to urban environments [70,131].

The central premise of this work is that interconnected devices can enable the collection of data to inform insights into human behavior, social patterns, environmental conditions, and other aspects of urban life. This information could then be used to modify systems to consume fewer resources, more rapidly satisfy the needs of citizens, and create informed public policy that is more closely aligned with environmental goals and community needs. Nevertheless, realizing the potential of these systems to improve cities is not a trivial undertaking as it requires the development of a mix of top-down and bottom-up technologies [11,163].

1.2.3 Today’s Top-Down City Sensing Systems

As the fields of UbiComp and IoT enter maturity, research achievements range from the use of sensor networks for detecting activities within homes, schools, and offices [84,100,100,152,173] to the monitoring of mobility infrastructure in neighborhoods and on highways [81,151,159]. Sensor and data networks have enabled better understanding of mobility patterns in cities [59], explained relationships between historical segregation and current economic income [124], and tracked the environmental degradation of water bodies near human settlements [20].

Most of these projects were developed and operated by research institutions and other large, well-funded organizations. Benefits from the collected data therefore tend to flow mostly towards these institutions instead of individuals and communities [198]. The fact that smart infrastructure tends to be deployed at city and national scales and is operated by a few large entities, either public or private, makes it hard for the systems, their data, and their insights to be used to adequately address pressing societal challenges. Furthermore, the scale of deployment for the systems is unable to engage communities in the use and upkeep of these technologies, leading to system decay, as citizens do not see value in maintaining them [15,37,64]. Current top-down deployments for sensing infrastructure and algorithms do not allow them to be as
adaptive, responsive, dynamic, flexible, and modular as urban spaces are [64].

1.3 Cities as Socio-Environmental Systems

Socio-environmental systems encompass the relationships among humans and the places they inhabit. To understand cities, this means understanding how humans collectively interact with the natural and built environment. Clear cycles emerge where social decisions induce significant changes in the environment, and environmental changes in turn shape future collective behavior. As intuitive as this might sound, the challenges of representing all the variables at play in such relationships are exponential and beyond the capabilities of current computing systems.

Consequently, researchers studying cities typically use techniques for data reduction to minimize the number of variables under consideration at any one time. Understanding only a subset of the factors, however, often leads to failed policies and interventions that can have deteriorating environmental and social impacts [94]. In contrast, a granular assessment of the cycle could bring better resource management practices, adaptive systems, and positively guided behavior change [30,142].

1.4 Community Sensing

Probing and understanding the complexity of socio-environmental systems calls for a re-conceived approach to sensor and algorithm implementation. Our understanding of socio-environmental systems may be well complemented through opportunistic, low-cost, modular, distributed, and community-based infrastructure. These characteristics allow sensors and algorithms to represent pictures of reality that are closely aligned with what communities value [93], something that is difficult to achieve with centralized top-down data infrastructure [198].

The work of this thesis proposes to expand the definition of community sensing beyond data collected about communities [93] to encompassing data pipelines that intrinsically stem from community challenges, needs, infrastructure, and desires.
Community sensing is thus not only about the information that can describe the community *per se*, but also the relationships between the community, their challenges and the environment. This enhanced knowledge can, in turn, lead to behavior, policies, and decisions that are more sustainable.

Operating within complex socio-environmental ecosystems requires integration of citizen goals, the natural environment, devices, algorithms, and private and public organisms. The central contribution of this thesis is to demonstrate community-based socio-environmental devices and algorithms that are specifically designed to address the lack of community-scale, bottom-up sensing infrastructure.

### 1.5 Contribution

The potential for community sensing to capture complex socio-environmental relationships is examined through five case studies that help to understand key characteristics of such systems. The first two studies show how sensors can be embedded in skiing and fishing equipment to create opportunistic data sources that help users and stakeholders understand how their activities both contribute to and respond to changing environmental conditions. Next, a study of carbon dioxide levels in buildings explores how adaptive algorithms can help to ease the implementation of technologies aimed at having multiple use cases and problem statements. A fourth study involving the implementation of a low-cost bioreactor, shows how creativity and scientific skill-enhancing devices can help to decentralize the development and benefits of the technologies that enable community sensing. Lastly, a sensor fusion system for bicycles is used to study how shared mobility infrastructure can be designed to incorporate community values in decision making processes.

The case studies help to illustrate how to sense variables of socio-environmental cycles that emerge in different urban and environmental contexts using a variety of opportunistic, low-cost, open-source, distributed, and community-based sensor and machine intelligence implementations. These implementations are evaluated as complementary to current urban IoT and as potential tools to address concerns about
community engagement and data centralization in top-down "smart city" infrastructure. Simply put, the contribution of this thesis can be understood as a set of tools that enable bottom-up collection of information about socio-environmental dynamics that emerge within cities.

1.6 Methods

The thesis presents a diverse set of projects that were developed to explore what characteristics socio-environmental sensors and algorithms must have to be successful and drive transformation in communities. The goal of the thesis is not only to provide concrete examples of systems but to also expose the process that leads to their formulation.

With this in mind, all projects have been made available online along with documentation and specific build files. Each of the projects includes files that can help future researchers to replicate, re-implement and re-evaluate all of the chapters of the document. Mechanical build files, electronic schematics, PCB milling codes, post-processing, artificial intelligence and embedded software have all been made available through each project repository.

The repository for Chapter 2 can be found in [148]. Similarly, the repository for Chapter 3 can be found in [149]. Chapter 4’s files can be found in [145], Chapter 5’s repository is located in [147]. Lastly, the files for the system described in Chapter 6 are in [146]. Readers are encouraged to explore the repositories to learn, remix and improve upon the presented work.

Author’s note: I view the creation process as valuable and insightful as the finalized polished versions of the projects. It is in the process rather than in the product that you can truly understand the drivers and constraints that shaped the technological outcome. With this I hope not only to outline technical and speculative aspects of a new breed of sensors but also to narrate their practical evolution.
Chapter 2

Whispers Of The Mountain: Opportunistic Sensing


2.1 Overture

It is well demonstrated that sensors can be used to collect large amounts of data about different aspects of our lives and environments. Defining the types of sensors that we use to gather insights about cities is just as relevant as determining the placement of those sensors and the specific variables they will record. Placement, embodiment, and usage of the device dictates the type of insights that the system will be able to offer. This chapter proposes the development of systems that can opportunistically acquire information about a diverse set of parameters that allow the reconstruction of a complex behavioral and socio-environmental reality as opposed to having fragmented bits of isolated information.
This chapter describes the development of a sensor system embedded and mounted on snow skis for distributed data collection. The system crowd-sources environmental and behavioral data through a distributed system as opposed to using traditional fixed sensor stations. The main focus is to outline relevant characteristics that we must consider when designing and implementing distributed, modular, and low-cost sensor systems that allow for multipurpose and opportunistic sensing.

This project was developed as part of a research collaboration with the government of the Principality of Andorra. The system’s key objective is to facilitate sensing of environmental and snow conditions (temperature, pressure, etc.) as well as behavioral aspects such as skiing patterns. The opportunistic and multipurpose capabilities make the system ideal for gaining insight into a complex socio-environmental system like the ones that emerge, in this case, around the activities and operations of ski resorts.

2.2 Background

2.2.1 Context

Instrumenting commonly used devices is a promising approach to the deployment of sensors at the community scale. For example, instead of using a fixed weather station to monitor environmental changes like humidity, temperature, and snow conditions, we can use multiple distributed systems to crowd-source a large terrain’s ecological profile. This allows stakeholders to have a more complete view of hyper-local changes and opportunistically sense environments at the exact point in which they intersect with human activity.

This chapter presents a voice-controlled sensor fusion system mounted on commercial skiing equipment. The device can record 6-axis acceleration, GPS coordinates, altitude, ambient temperature, atmospheric pressure, ambient humidity, and distributed ski/snow surface forces. The system leverages skiers’ motion and the skis’ response against the snow to help evaluate skier performance, equipment design, and how terrain, environment, and snow conditions interact with each other.
These types of devices allow for monitoring environmental variables and, at the same time acquiring information about skier behavior on the mountain. Field interviews with mountain and ski experts allowed us to assess that this data can help stakeholders, communities, and experts answer questions such as: What are the sections of the mountain that people most use? Where do people spend their time? What are the most dangerous areas? How are skiers using their equipment? Where should resorts focus their resources to avoid wasting energy from snow-making and grooming operations? Furthermore, environmentalists can use the data to research how alpine micro climates evolve over the years. Environmentalists can then inform industries and mountain users on practices they can avoid or embrace to help preserve the natural ecosystem.

2.2.2 Related Work

A survey of existing work on sensor systems for skis reveals a variety of research objectives and implementation strategies. For example, readings from accelerometers have been used to detect alpine ski turns [86], as well as for analyzing an athlete’s biomechanics of ski jump landings [17]. In addition, for understanding forces within the ski, a custom-made surface-mounted strain gauge has been proposed for measuring turning direction, angle, and deflection [193].

Sensor fusion systems that take input from sensors mounted on the skier, their boots, and their skis have been used to compute athlete exertion power [95]. Approaches that use embedded electro-mechanical systems include an embedded dynamometer installed to measure torque and forces at the ski-binding interface [118], and embedded piezoelectric components that serve as active actuators to help dampen ski vibrations to enhance performance [178]. However, the systems shown in prior work do not include the capability of fusing multiple sensor inputs to enable parallel insights about various aspects of the mountain, ski, or user.
2.2.3 Contribution

As part of a comprehensive study of human-alpine relationships called "Whispers Of The Mountain", this sensor fusion system was mounted on commercial skiing equipment. The system is unique in that it can record forces acting across a majority of the surface area of the ski. At the same time, it records variables that are directly related to environmental conditions like humidity, temperature, atmospheric pressure, and position. It does so with a three-module system mounted on the skier’s equipment. As an example of "opportunistic sensing", this platform represents an emerging class of participatory sensing, where sensors are embedded in artifacts that are already commonly used, allowing people to collect data without changing their existing activity patterns.

The prototype sensor system was tested in actual skiing conditions to validate the design and analyze the types of information obtained from its output. Results show that the tool can act as a crowd-sourced alternative for monitoring mountain and behavioral conditions within skiing resorts. In addition, the system shows potential for helping to balance out economic benefits, user experience, and environmental impacts that come from the skiing industry.

2.3 System Design

2.3.1 Hardware

The system’s three separate modules allow for flexibility in deployment. Each module can function independently from the others, allowing researchers, resorts, and users to tune their settings according to their needs. All of the modules have built-in radio communication capabilities to coordinate data collection between the units. Control of the system is done by the helmet module (HM), the second module corresponds to the rights ski’s module (RM), and the third to the left ski module (LM). Figure 2.3.1 shows the implementation of the three modules.

As illustrated in Figure 2.3.1, each module is responsible for accessing different
sensors and independently storing their data. The modules communicate to coordinate clocks, file names, and operations. Data is stored locally on Secure Digital (SD) cards on both ski modules. Post-processing and multi-frequency data fusion is carried out on an external computer. The complete system is capable of recording the variables listed in Table 2.1.

Figure 2-2: Full system architecture. The image shows the three main modules and the variables that each of the modules is responsible for. All of the modules communicate with each other through wireless radio signals.
Table 2.1: Ski sensor fusion system stored variables and units.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>°C</td>
<td>Right</td>
</tr>
<tr>
<td>Humidity</td>
<td>% Relative</td>
<td></td>
</tr>
<tr>
<td>Pressure</td>
<td>% Pa</td>
<td></td>
</tr>
<tr>
<td>X Axis Acceleration</td>
<td>m/s²</td>
<td></td>
</tr>
<tr>
<td>Y Axis Acceleration</td>
<td>m/s²</td>
<td></td>
</tr>
<tr>
<td>Z Axis Acceleration</td>
<td>m/s²</td>
<td></td>
</tr>
<tr>
<td>Ski Surface Forces</td>
<td>m/s²</td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>DD</td>
<td>Left</td>
</tr>
<tr>
<td>Longitude</td>
<td>DD</td>
<td></td>
</tr>
<tr>
<td>Altitude</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Voice Labels</td>
<td>1 - 5 (User Defined)</td>
<td>Helmet</td>
</tr>
</tbody>
</table>

**Helmet Module**

The helmet module (HM) is attached to the side of the helmet, and it serves as the primary control interface for the system. HM acts as a voice-activated interface that sends commands through built-in radio frequency antennas (RF24). The unit is based on a commercial Teensy 3.2 board connected to a voice processing board. The HM interface supports five different voice commands listed in Table 2.2. Users activate the commands to control the system and label specific data points.

The system gives feedback to the user with a Red-Green-Blue (RGB) Light-Emitting Diode (LED) indicator that shines different color animations through a fiber optic line placed at the front of the helmet where the user can easily detect it. The terminal of the fiber optic line transmits information through to the periphery of the user’s line of sight. The interface is designed to be minimally distracting to the skier while maintaining effective communication. Different light patterns and colors indicate that the voice commands have been recognized and sent to the sensor modules mounted on the skis. Figure 2.3.1 shows the helmet module installed and in operation.
<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate Ski</td>
<td>Activates full system and sets it on standby for recording command.</td>
</tr>
<tr>
<td>Begin Log</td>
<td>Begins logging data on all modules.</td>
</tr>
<tr>
<td>End Collection</td>
<td>Stops data collection on all modules.</td>
</tr>
<tr>
<td>Flag Moment</td>
<td>Flags specific moment on data log for future identification on post processing.</td>
</tr>
</tbody>
</table>

Table 2.2: Helmet’s voice interface commands and control actions.

Figure 2-3: Assembled helmet module with radio communication, voice input and peripheral feedback user interface (UI).

**Right Ski Module**

The right ski module (RM) tracks all variables related to the interaction between user, ski, and terrain, and environmental variables like ambient temperature, humidity, and atmospheric pressure. The module is based on a Teensy 3.2 development board [138]. It uses the central Teensy board to communicate with a custom research board known as a "TerMITe" [166]. The TerMITe sensor board is an environmental sensor with a built-in 9-axis IMU, temperature, light, and atmospheric pressure sensors. Figure 2-4 shows how the module interconnects devices through power and communication lines.

RM operates at a sampling frequency of $f_{LM} = 45HZ$. The sampling frequency was chosen to capture subtle changes in movement while not exceeding the processing capabilities of the 32-bit ARM Cortex M4 microcontroller used by the Teensy 3.2 board. The board saves the data locally in an SD card and communicates through
Figure 2-4: RM electronic diagram. The system has a central unit, a Teensy 3.2 microcontroller and uses peripherals to access the piezoelectric array and external sensors. Its radio frequency module with the helmet module for coordinating collection times and labeling data. One of the most important connections on the right module is the connection to the ski embedded piezoelectric array. This array collects data that describes strains and vibrations on multiple points of the surface of the ski. The array’s data can infer skiing conditions, snow conditions, and overall performance.

Piezoelectric Sensor Array Description

Distributed forces on a surface can reveal relevant information about the object’s interaction with its surrounding environment. In the world of sports, there is significant interest in analyzing the forces that impact sporting equipment, as they may offer insights into how different materials perform, how different athletes interact with their apparatus, or how other external forces influence performance [2,110]. Furthermore, recent developments in sensing and computational technologies have made it possible to seamlessly instrument these devices in a low-cost manner for a wide range of purposes [143].

Within the winter sports industry, there is great interest in the various ways in which the design of snow skis – their shape and materials – can alter the experience or
performance for a given user [6]. Distributed sensing on a ski can improve the design process by revealing the response characteristics that different material combinations have to external forces resulting from user action, terrain, snow conditions, or a variety of each.

The surface array is composed of five longitudinally-mounted sub-arrays that log data independently to obtain information at five distinct locations within the top plate of a commercial Stöckli® Laser GS Race Ski. The prototype has three sub-arrays on the front and two on the back, as seen in Figure 2-5. Components are mounted on the surface of the ski using a layer of commercial rubber coating which protects the circuitry from low temperatures and moisture.

Electro-mechanical connections are accessed through waterproof plugs placed on the front of the boot binding. The plugs have six contacts, five correspond to signals coming from each sub-array, and the remaining one corresponds to the common GND (ground) signal. Contacts and signal outputs coming from the back end of the ski are traced and reinforced through the ski bindings so that they can withstand rough forces during use.

**Array Electronic Design**

Each sub-array comprises five piezoelectric transducers connected in parallel. Connecting the transducers in parallel allows for the voltage generated on each transducer to be equal [104], meaning that \( v(t) = v_1(t) = v_n(t) \) where \( v \) represents the voltage drop and the subscripts correspond to each connected element. In this design, \( n = [1, 5] \) to ground the signal a pull-down resistor of 220k\( \Omega \) is connected between the positive terminal and ground (GND) on the transducer arrays. Each set of five transducer is connected to a voltage divider, as can be seen in Figure 2-6. The voltage divider remaps the signal domain to a maximum of 5 volts so that the output is compatible with most commercial micro-controllers.

The system consists of five individual sub-arrays, each containing five transducers. Three arrays are placed on the front of the ski and the remaining two on the rear end. Signals derive their power from the vibrations and movement of the skis themselves.
Figure 2-5: Picture of both sections of the embedded ski sensor array. A) shows the front end of the ski. Labels 1–3 correspond to the front three piezoelectric sub-arrays. Label 4 shows the prototype area that contains all the embedded electronic circuits for signal pre-processing. Label 5 marks the waterproof plugs for accessing the sensor’s signals. B) shows the rear end of the ski with Label 1 indicating the location of the circuits for signal processing and Labels 2–3 marking the two rear sensor sub-arrays.

Figure 2-6: Electronic schematic for the individual sensor array. Labels T1–T5 show the five ceramic piezoelectric transducers connected in parallel. The signal is grounded through a 220kΩ resistor and is attenuated with a voltage divider of 1MΩ and 68kΩ. The output signal connection is the terminal that can be probed to obtain the signal from the entire sensor.

They do not require an external power source to send readable signals to a microcontroller. This aspect of piezoelectric transducers makes for an attractive use case in low-power sport sensor applications [157].

As seen in Figure 2-7, the five sub-arrays are connected to a commercial microcontroller\(^1\) [138] with a commercial SD card reader/writer [139] attached to it to sample the signals and save the recordings. The voltage source seen in Figure 2-7 is

\(^1\)It is relevant to note that almost all commonly found micro-controllers with GPIO (General Purpose Input Output) pins can work with this system.
Figure 2-7: Electronic schematic for the complete piezoelectric array and its connections to the teensy microcontroller. The array is composed of five individual piezoelectric arrays containing five transducers each. The complete design has 25 transducers, which allow us to obtain measurements on different segments of the ski. Labels P1–P5 show the five sets of piezoelectric arrays, each of which follows the schematic of Figure 2-6. A 3.7V LiPo (Lithium-ion Polymer) battery powers the system. A commercial micro-controller board (Teensy 3.2) with a built-in SD card module measures and logs independent transducer signals.

only used to power the micro-controller for reading the signals. The micro-controller is set to sample at $f = 45Hz$. A high sampling frequency is required to capture sudden changes found in high-speed sports.

Embedded Array Fabrication

The manufacturing process for this system uses low-cost and low-tech components, which allows for the system to be easily replicable for iterative experimentation and design. As shown in Figure 2-8, the process begins by aligning and attaching individual piezoelectric transducers on the ski’s surface. The first front transducer is placed 110mm away from the front of the ski binding to give enough room for the embedded circuits, mounting connectors, and the logging device. Similarly, the first transducer placed on the rear end of the ski has a distance of 40mm from the binding to provide enough room for the embedded circuit. Next, each transducer is attached to the ski’s surface with a methyl cyanoacrylate-based adhesive. The pieces are left to set until adhered.
Figure 2-8: Manufacturing sequence of the embedded sensors. A) shows how the piezoelectric transducers were aligned and attached to the ski surface. B) shows the layout for the circuit traces. Note that all components share a common digital ground reference. We place a resin layer on top of the sections that cross over different arrays to avoid short circuits. C) shows the pull-down resistor and voltage divider circuits composed of SMD components directly soldered onto the copper traces. D) shows the transducers after the final coating layer was applied to make the system waterproof.

Copper tape with a 6mm width is used to connect the transducers. The first trace works as digital GND and connects all the transducers. After that, anything connected to the transducers’ outer rings (gold-tinted section) acts as GND. For this reason, a mix of nitrocellulose dissolved in butyl acetate is used to cover the outer ring sections that are crossed by the tape connecting the positive terminals of each transducer (white tinted circles). This allows us to lay down the traces for signals without any short circuits to GND².

Once all traces are placed on the surface, SMD resistors with a 0603 package can be soldered directly on the top side of the tape and transducers. Once every connection is tested for continuity and resistance, seven layers of a multipurpose rubber coating is applied across the surface. Finally, connections are added between the copper traces and waterproof plugs to make it easy to access the terminals for all five sensors and GND. Traces located in ski areas that are known to experience the

²All connections are secured by soldering the upper side of the tape with the transducer.
highest strain (middle section, front, and rear of binding) are reinforced with moldable silicon rubber. It is crucial for all materials to be flat and of small proportions. This maintained the overall thickness of the system at a level that did not interfere with the user’s motion or notably alter the performance characteristics of the sporting equipment.

**Left Ski Module**

The left ski module (LM) collects GPS coordinates. This module is based on an Adafruit Adaloggger board with built-in SD card capabilities [1]. The board samples information at $f = 1Hz$. This is the standard sampling frequency for GPS modules. Similar to the other modules, the device has a radio frequency module that allows it to communicate with the other modules to coordinate file names, clocks, and recording times.

Figure 2-9 shows the electronic diagram of the module, detailing power and communication lines. The system has enough I/O pins to add more sensors to collect data at lower frequencies. Future versions of the device can build on this capability to lower the burden on the right module and add more sensory capabilities like capacitive sensing on the ski’s core structure.

### 2.3.2 Software

**Module Programs**

Each module has an independent embedded program developed in C++. The modules are programmed to access their peripheral sensors at the outlined frequencies. Internal processing verifies that data packets are structured correctly. The program is also responsible for generating random file names and assigning the same file names to the two ski modules to make post-processing easier.
Figure 2-9: LM electronic diagram. The system has an Adafruit Adalogger as its central unit and uses serial and SPI communication to access GPS and Radio Modules.

**Data Fusion Algorithm**

Data is recorded independently on the RM and LM. LM has a slower sampling frequency of $f_{LM} = 1\, Hz$ compared to the $f_{RM} = 45\, Hz$ that RM has. LM operates at a lower frequency because it has to operate at the rate at which GPS satellite data updates, which is once every second.

RM’s data describes interactions between the user, equipment, and terrain; therefore, the sampling frequency must be high enough to capture subtle differences over minute lapses of time. As previously noted, LM’s geospatial data can add a layer of information essential for extracting richer insights from visualizing and analyzing the data.

The difference in sampling frequency for each module and the need for proper fusion of the data sets to obtain richer information calls for an adequate algorithm for fusing the data sets from each module. The chosen fusion algorithm takes the fast frequency data set as the base and aligns data points from the slower frequency set to it. It uses coordinated timestamps to align data points that are taken at the same time.

As the higher frequency data set has more data points for the same amount
of time, spline interpolation is used to fill in data points that are not collected on the low-frequency data set. Interpolation between existing latitude, longitude, and altitude coordinates is used to get prime coordinates for each data packet on the high-frequency set.

2.4 Validation

2.4.1 Deployment

The prototype system was iteratively tested at three different downhill ski sites in the Northeast United States on three different days in the winter of 2020. The system was used by four different users during experimentation. The goal of the tests was to validate the robustness and performance of the system under typical skiing conditions. As a result, the system could withstand rough climatic conditions (low temperatures and humidity), showing robust and appropriate operation during all days of testing.

Figure 2-10: Prototype deployment on real mountain and use case conditions.
2.4.2 Data

Piezoelectric Array Data

Figure 2-11 shows a visualization of data recorded for two ski runs through the use of a frequency heatmap or spectrogram. The relative intensities of different frequency components are plotted over time in this case. Differences in intensity can be seen over time, as evidenced by the vertical orange bands during the "ski run" in contrast to the red areas when the ski was stationary or otherwise not in contact with the ground. Additional variation in frequency components is revealed by a greater presence of lower frequencies toward the front of the ski, indicating that this part of the ski is subject to different forces than the rear.

The data for the spectrogram was taken from a test run that lasted 967.012 seconds. The runs are composed of two downhill slides, which we can see in the lighter regions of the spectrogram, and moments of rest, which can be detected in the darker areas. It is relevant to note how intensities also vary depending on the area of the ski. We can see that the front end of the ski receives forces with larger magnitudes as color regions on P1–P3 have more considerable contrast than P4 and P5.

At present, the piezoelectric array is not calibrated to provide absolute measures of force, accelerations, or vibrations, so data output is restricted to measures of relative variation. Nevertheless, relative measurements can allow for comparisons of different materials, patterns, and conditions using appropriate tools for visualization and statistical analysis.

Fusion Algorithm Data

Running the data fusion algorithm produces a single high-frequency data set. Interpolation generates granular information at the level of the turns that the skier does during a run. Figure 2-12 shows a map plot made with an interpolated and fused data set. We can see that the algorithm can expand the low-frequency data set and fill in the missing data points with accurate interpolated points.
Figure 2-11: Frequency Spectrogram. The Figure shows the transformation of the signals coming from the five sub-arrays of the system into the frequency domain.

Evaluating the fusion algorithms allows us to see that we can have highly accurate geo-localized information as all of the right module’s data points are embedded into a single geo-localized point. This allows the distributed system to function as a mountain sweeping and profiling device, making it an interesting alternative to fixed weather stations. The image also plots the values from the embedded array to show how forces are entirely different at certain moments of the downhill run. For example, we can see low points colored in magenta when the user goes up a lift and yellow spikes at turns during ski use. Additionally, the visualization shows a blue color gradient on the data points representing the atmospheric pressure data; we can see that the gradient changes accurately as the user goes up and down the mountain.

Multipurpose Sensing Data

The two subsections above illustrate how we can get specific insights into different variables and actions during a ski run. The system shows its capability to fuse data points into geo-localized data instances with multiple data layers. Each of these layers can be used to address different questions. With the help of a visualization interface, ski resorts, environmental activists, recreational and professional skiers can look into
the data and understand the complex socio-environmental systems that emerge in the mountains.

Figure 2-13 illustrates different runs and temporal moments mapped into another visualization of a skiing resort. Geo-localized points are shown with 3D bar plots representing variables collected by the piezoelectric array, pressure sensor, temperature, system clocks, accelerometers, and gyroscopes. The plots help to see sensor values and calculate meta values such as time spent at a given spot, historical temperature changes at a given location or infer variables such as skiing speed or turn forces. In addition, the visualization allows us to extrapolate how the value of the system would increase if it were to be embedded on the rental equipment of the resort. It also validates how the system can be used to inform multiple stakeholders with distributed geo-localized data, information that can help monitor mountains, behavior, and equipment use. An example of such implementation would be to use the system to help ski resorts monitor locations that require the most artificial snow operations and assess degradation of rental equipment.
Figure 2-13: Sensor fusion plots on multiple mountain runs. The visualizations help to see the insights that the system has the potential to give to stakeholders.

2.5 Discussion

2.5.1 Limitations and Future Work

The current implementation of the prototype was installed on one prototypical pair of skis. Future work will center on standardizing the system to be easily replicated and embedded in more equipment. It is important to keep in mind that the system’s potential can be further enhanced through swarm dynamics that would emerge if it is to be used on multiple skis within the same mountain. Multiple and distributed systems would allow us to gather granular information that is too expensive or time-consuming to collect with traditional approaches.

Proper calibration and labeling of the data can help us develop more possible use cases for the system. For example, if we label data while skiing, we could train algorithms to detect certain types of snow, certain types of skiing dynamics, or even to detect mountain accidents automatically. As we interconnect and label the data to enable artificial intelligence to help us gather deeper insights than possible today, it is also relevant to think about low-power modules that would allow for lasting operation without recharging batteries or changing power supplies constantly. Similarly, the communications and data transference modules could be improved so that the skis regularly feed their data into fixed stations located at chair lifts. This would eliminate the operational burden of manually extracting the data from the local SD cards and make it easier for the mountain communities to map out their environment.
collectively.

2.5.2 Conclusion

Opportunistic, distributed and multipurpose sensing will become key to implementing systems that can provide valuable insights to a range of stakeholders. As cities and the natural environment have tangled interactions between variables, the availability of systems that can gather information on different variables becomes crucial for reaching the sensing infrastructure's true potential.

The system and its insight visualizations allow stakeholders to ask questions such as: What were the snow conditions at the places where the skier spent the most time? What temperature changes are observed at a given location? What sections of the mountain induce the most strain on the skiing equipment? These questions naturally emerge and can be answered from the data that the presented system can help collect. As we will see in the next chapter, such questions and insights can help us drive the transformation of collective behavior and our surrounding environments to promote more sustainable living.
Chapter 3

Ocean Logger: Collective Behavior Transformation

A version of this chapter has been submitted for review and publication at the IEEE Sensors and Applications Conference 2022.

3.1 Overture

The last chapter showed us how opportunistic, low-cost, and multipurpose systems can help us gain insights that are not possible with traditional top-down sensor stations. In addition to having proper functional characteristics, finding key feedback loops into which the systems can feed information is critical to driving collective behavior transformation.

Sensors deployed at the community scale can be used to inform collectives on specific issues of interest as the sensors become an extension of the collective’s knowledge. They can inform decisions and behaviors that transform the natural and built environment in which the subjects are immersed. This chapter outlines the development of a second system. The system is used for ocean profiling in coastal waters. The goal of the device is to give fishing communities capabilities for actively and oppor-
tunistically monitoring the waters where they fish without interrupting or negatively impacting their activities.

The project builds on ideas coming from New Zealand’s famous MOANA project. It expands the idea of using technological enhancements to improve the sustainability of industrial fishing vessels into the realm of small community-owned boats in coastal fishing villages. Communities using these sensor nodes could gather information and change their collective behaviors according to the information they receive from the sensors. These community feedback loops can lead to more efficient and productive economic activities while keeping environmental degradation at a minimum.

This chapter describes the characteristics of the ocean sensor node. It also outlines a network topology that could be used to inform fishing communities about the state of their waters. This can, in turn, allow them to decide on the collective practices that will allow them to maximize their fishing yields while preserving balance in the ecosystems that they interact with. For example, the system can be used to inform fishing communities about the likelihood that a specific fish species will inhabit certain areas. In the case that the community has already fished too many individuals from a species, they might choose to look for a fishing spot that is less likely to be populated by that species in order to give the species’ population time to regenerate. The sensors can be used to avoid over-fishing and coral reef degradation while helping fishers maximize their efforts economically.

3.2 Background

3.2.1 Context

Ocean exploration is a key area of research for the United Nations within the current decade [75]. Ocean instrumentation will be the means to provide essential insights into how climate change affects our natural ecosystems, and modern society [140].

While a large amount of instrumentation exists within our oceans, critical coastal waters are still not yet adequately mapped [44]. Coastal waters are defined as bodies
of water that lie next to land masses with a maximum depth of $200m$. These regions house the most diverse and dynamic phenomena observed in the oceans [113]. In particular, coastal waters are home to the majority of the planet’s coral reef ecosystems [36]. These ecosystems have been proven essential to the holistic balance of marine environments. Therefore, mapping the temperature and acidity of coral reefs is vital to understanding the implications of climate change on such environments in the short and long term [73] as well as understanding ways in which human activity impacts said environments.

Challenges like underwater communication, localization, pressure, power constraints, and saltwater make distributed ocean sensing difficult to implement [71]. A large number of efforts have been made to profile the ocean through buoys or temporary instruments that tend to be deployed and rescued within a period of weeks [103]. Amongst the most successful implementations we can find the Argo floats [153] and the Low-Cost Miniature Isopycnal Floats (Minions) [168]. However, these floats require sophisticated equipment and trained personnel for proper operation, significantly increasing deployment costs.

High costs and technological sophistication make ocean monitoring equipment and ocean profile data hard to access. This impedes information from the systems from being available for people in constant interaction with the ocean. Designing low-cost and distributed network topologies could aid in democratizing access to ocean technologies, thereby benefiting populations such as fishing communities, which would be empowered to better understand their relationship with the ocean and engage in more sustainable activities.

3.2.2 Related Work

Researchers have identified clear opportunities to use existing ocean vessels for monitoring coastal waters [177]. Apart from gathering data for scientific research purposes, establishing a relationship between fishing vessels could create a valuable data ecosystem that would benefit communities relying on the blue economy for survival. [87]. For example, the MOANA project in New Zealand has attached sensor nodes to large
shipping vessels and proved that the collected data is useful for both ocean research and shipping vessel operators [106] as it can inform vessels about types of species that are abundant at any given location.

While the MOANA project’s sensors work well with commercial shipping vessels and have proven to crowdsourcing helpful ocean profiles successfully, their nodes have not been implemented on vessels owned by non-industrial fisheries [177]. Leveraging smaller boats for ocean device implementation could increase the mapping capacity for projects like MOANA around the globe.

3.2.3 Contribution

The proposed system is a low-cost, open-source sensor fusion device that small fisheries can use for crowdsourcing data from coastal waters and coral reef environments. The device is equipped with depth, temperature, pH, and light sensors. Sensor data is stored locally on the device and transmitted via radio signals onto an onboard logger or an online server when the sensor is on the sea’s surface.

This chapter presents (1) the electronics and mechanical design of the device, (2) an implementation outline for scaling sensor networks based on two versions of the system (local storage and networked prototypes), and (3) field testing and characterization of the system. The device was manufactured using standard digital manufacturing techniques and tested in three different bodies of water (a pool, a river, and in the open ocean).

3.3 System Design

3.3.1 Hardware

Community Network Design

Building on knowledge about the benefits of opportunistic sensing [99], this project intends for fisheries that use the system to gather data about the oceanic regions they occupy collectively. Hence, the system’s value would increase proportionally
with the number of users joining the network. Further, system data could inform their fishing decisions such that fisheries can optimize for yield without overfishing the marine ecosystems. For this purpose, a multi-node distributed network will be employed. Each node will have two separate components, one for logging the data (sensing module) and another for uploading it and sharing it with the entire network (communication module). Figure 3.3 illustrates each node and the entire network topology.

The network topology allows fishing collectives to share insights that can help peers have more sustainable fishing practices. In addition, the network could be locally controlled, ensuring that the community wholly owns its data. These types of networks can bring communities power for improving their immediate environments and tools for them to put data-based pressure on higher institutions or governments to make policies and interventions that further aid local economic and preservation efforts.

Figure 3-1: Illustration of sensing and communication nodes placed on fishing boats. The sensing module communicates with the communications module during each surfacing. Communication nodes will relay data onto the global network. Nodes can then be used to form a star network topology based on multiple fishing boats.
The sensor module is a device based on an ATTINY 3216 microcontroller [119]. As shown in Figure 3-2, the module is powered by a 2500mAh lithium-ion battery pack. The current prototype has three sensor modules connected to the microcontroller: the TE-Connectivity Bar30 depth/temperature sensor [25], the e-Gizmo E-201 pH sensor [38] and an Everlight visible light photo-transistor [43]. The sensors enable the extraction of information for profiling temperature, depth, altitude, atmospheric pressure, light intensity, and acidity levels. While in collection mode, the device queries each sensor at a frequency of .5 Hz. This frequency was chosen to balance power consumption, sensor adjustment, and temporal coverage of the ocean profile.

![Sensing Module Circuit Design](image)

Figure 3-2: Sensor node component description with communication and power lines connections.

**Power management**

The sensor module cycles between high power and low power modes of operation. The module accesses information from all peripherals in high power mode and saves the data onto the SD card. In addition, the module checks the current depth value every minute for submersion on low power mode. Switching through power modes allows for extending the nodes’ battery life.

On low power mode, the device has a current draw of 15mA, meaning that a fully-charged battery would roughly operate for one week. On high power mode, the
device draws 55mA, meaning that a fully charged battery would operate for roughly two days. By cycling through these two modes, battery life is extended significantly, increasing the ease of use for fisheries. In addition, device battery life could increase further when using the microcontroller’s deep-sleep functionalities, but this is not yet implemented. Adding transistor switches for each component could further improve the device’s power consumption, but is beyond the scope of the present research.

**Mechanics and waterproofing**

The mechanical design of the node housing prioritizes ease of access to the electronics while maintaining waterproofing. Easy access allows for efficient debugging and scalability of the module. Making the node hard to maintain would add an additional burden on the users, and would not allow it to reach its full potential. Additionally, since the node is intended for use in the coastal water region, the maximum depth it will be submerged to is 200m, which was taken into account when choosing sensor modules.

The main PCB, sensor probes, and battery are placed onto an acrylic mount. See Figure 3-3-A for visual details. Each component is fastened to the acrylic with bolts, standoffs, and nuts. This creates modularity in case there is a need to replace individual components, and also reduces the amount of vibration experienced by the electronics.

The acrylic plate slides into slots inside a custom-made cylindrical body, see Figure 3-3-B. The cylinder has two holes for placing the pH probe and the depth and temperature sensors, allowing direct contact with water. The cylinder is enclosed by a lid that is mechanically fastened by nuts and bolts to keep the closure as tight as possible and able to withstand high pressure.

The device’s current version requires commercial acrylic silicon for waterproofing the sensor through holes and the lid-cylinder connection. First, silicon is applied to the cylinder’s edge and lid. Next, the screws are placed, and the enclosure is set to dry for eight hours before use. The cylinder body and the device lid are manufactured using 3D stereolithography printers (Form3). This type of print does not compromise
the structure’s impermeable characteristics and can be used for creating waterproof enclosures [137].

Once the device has been sealed with silicone, a set of weights is attached to make it negatively buoyant. Weights are attached directly to the cylindrical body using a fishing line. Figure 3-3-C shows the device prepared for ocean deployment.

![Image](image.jpg)

**Figure 3-3:** Illustration of mechanical and electrical implementation of the sensor node design. A) Electronic components mounted on acrylic plate. B) Acrylic plate inserted into cylinder body.

**Networking Module**

Having the capability to view data in semi real-time from various nodes across various locations (nearby and far) increases the accessibility of the data to non-scientists and informs users of currently occurring natural marine processes while they are in the field.

The hardware of the communications module is based on an ESP-WROOM-32 microcontroller manufactured by ESPRESSIF [42] capable of WiFi connection and...
equipped with 4MB flash storage. The same system architecture as the one illustrated in Figure 3-2 is used. However, an ESP32 microcontroller is used in place of the ATTINY 3216 as the microcontroller.

The ESP32’s AsyncWebServer library is used to establish a server connection through which sensor readings are graphed online. The microcontroller demonstrated the ability to visualize data after connecting to a phone’s hot spot. See Figure 3-4 for an example semi real-time mobile visualization interface. Visualizing the data increases the overall convenience and accessibility of the data to fisheries and other non-scientists who would use the node.

Future work will integrate the sensor and communications nodes into the network topology described above. In this application, the ability to visualize data in real-time at various locations can inform fisheries and scientists of natural processes while working in the field.

![Ocean Iot Sensor Logger](image)

Figure 3-4: Screenshot of server data visualization on mobile phone.

### 3.3.2 Software

As stated in section 3.3.1, the protocol is classified as a finite state machine. The two central states of operation are the standby state (low power mode) and the data
Table 3.1: Saved Data Structure: collected variables, selected sensors, and data units.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sensor</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Internal MC Clock</td>
<td>milliseconds</td>
</tr>
<tr>
<td>Water Acidity</td>
<td>GAOHOU PH0-14</td>
<td>pH</td>
</tr>
<tr>
<td>Temperature</td>
<td>BAR30</td>
<td>Celsius</td>
</tr>
<tr>
<td>Pressure</td>
<td>BAR30</td>
<td>mbars</td>
</tr>
<tr>
<td>Depth</td>
<td>BAR30</td>
<td>meters</td>
</tr>
<tr>
<td>Altitude</td>
<td>BAR30</td>
<td>meters</td>
</tr>
<tr>
<td>Light</td>
<td>1080-1379-1-ND</td>
<td>Analog 3.3V</td>
</tr>
</tbody>
</table>

collecting state (high power mode). When the device registers submersion (the depth sensor value is greater than zero), it enters the data collecting state. The device accesses each sensor within this mode, builds a data packet with shape [1 : 7], and saves it into a new line within a .txt file on the SD Card. Each packet is timestamped to facilitate the post-processing of the data. Table 3.1 outlines the sensor variables and their units.

When individual nodes integrate with a more extensive network of nodes, a data transfer state is needed. This state would be entered once the device registers it is at the surface. Here, the device establishes radio or WiFi communication and transfers data to the global network’s servers. After data transfer, the device returns to the initial standby state. Figure 3-5 describes the finite state machine model that our implementation follows. The following section outlines a WiFi-enabled node’s design, connectivity, and performance.

### 3.4 Validation

#### 3.4.1 Deployment

The device was validated in three different bodies of water (pool, open ocean, and river). Testing occurred in different bodies of water to validate different capabilities: (1) sensor reliability and performance, (2) housing enclosure reliability, and (4) power cycling.
3.4.2 Data

Pool

The device was first deployed in a freshwater pool. This experiment validates the mechanical design and electronics performance. As illustrated in Figure 3-6, the device was submerged at two distinct depths (.9 m and .5 m). The device was underwater for 20 minutes (15 minutes at .9 meters and 5 minutes at .5 meters).

In the graph, two distinct steps in depth can be observed. There are some variations in the depth readings because the node was not perfectly fixed at depth underwater. Evaluation of the light sensor was done by covering and uncovering the device. The actions are translated onto the graph at approximately 4, 9, and 16 minutes.

River

The device was deployed in the Charles River in Boston MA, USA. This deployment was intended to evaluate and certify the power cycling algorithm. The experiment lasted 18 minutes. Samples were collected at two distinct locations and two different water depths (3 m and 4 m). The river depth limited these depths. The device activated and deactivated the collection states successfully for both locations.
Figure 3-6: Data plot for 20 minute sensor deployment. Key variables for this experiment are depth and light. Plot shows testing of depth sensor at two different depths and intermittent occlusion of light sensor.

Figure 3-7 shows that the device only collects data when it is submerged. As soon as the device resurfaces, data collection stops. The straight slope lines indicate interpolation of the last data point with the first after reactivation of data collection. The time axis indicates the minutes that passed in between each cycle. Detailed information can be gleaned from the collected data, such as minimal temperature differences at the different depths.

This experiment was carried out after sunset. We can see this on the light sensor due to readings that indicate minimal exposure to light. This can also be attributed to the muddy waters typical of the Charles River in Boston, Massachusetts, USA. It is relevant to note that these readings also illustrate the device’s potential to capture water turbidity levels.

**Open Ocean**

Finally, the device was tested in two scenarios within the open ocean. The deployment was carried out in the Gulf of Mexico in Cabo Catoche. The first deployment, shown
Figure 3-7: River deployment showing power cycling operation and intermittent data logging. Two different locations and depths were measured within a span of 18 minutes.

in Figure 3-8, was a 5-minute shore test used to confirm that the pH probe gave reliable data. According to NOAA (The National Oceanic and Atmospheric Administration), ocean acidity averages are currently at 8.1. We would thus expect to see a value close to that average. Unfortunately, the device recorded values at around 5.5 units\(^1\). Therefore, the absolute value of data readings from the probe were not validated. However, the readings are stable and on-site calibration of the probe could allow for more accurate and precise results. The experiment allows us to conclude that the pH probe is highly precise but not accurate, meaning that the current version can be used only for relative and not for absolute profiles.

A second open water test was conducted to validate the entire system design in the open ocean. The device was deployed from a fishing boat for 90 minutes at two distinct locations and depths to validate the mechanical and electrical design. As illustrated in Figure 3-9, the first submersion lasted about 40 minutes and was conducted at 3.9 meters. The second submersion lasted around 20 minutes and was

\(^{1}\)This does not mean that the sensor is off by 2.6 units as the ocean can have areas of PH below average. Future work will use a commercial device to set a true baseline value to evaluate absolute performance of the device.
Figure 3-8: Collected data for a 5-minute experimental log deployed in a shallow water shoreline. The purpose of the experiment was to validate water acidity readings and the fusion of multiple sensors. The current pH sensor shows high precision but low accuracy, meaning that it can be used for relative measurements and not absolute ones.

Conducted at 5.9 meters (the maximum tested depth for the device).

In the first submersion, the device became stuck in a position such that the phototransistor could not operate properly, leading to inconsistent readings. In the second submersion, the device turns underwater due to tides and currents. As a result, the sensor readings can be used to understand how light dissipates with depth in this environment and observe the changes in temperature and pH at different depths.

Overall, the collected data illustrates how the device can be used to profile temperature, acidity, and light readings at different locations and depths in the ocean. The collected data can be of great use to scientists and fisheries alike [32].
Figure 3-9: Log from second ocean experiment. Experimented lasted 90 minutes. Data was sampled at two distinct locations and depths to evaluate the system’s mechanical and electrical performance. The graph shows successful operation of the device along with possible applications of collected data.

3.5 Discussion

3.5.1 Limitations & Future Work

Modular Waterproofing

The current module needed to be sealed with silicone to ensure absolute waterproofing. Improving the mechanical design to prevent the need for silicone sealant will significantly increase the device’s ease of use and maintenance.

pH Sensor Probe Calibration

As stated above, the pH sensor probe did not give correct absolute values. Accurate readings can be obtained through the implementation of an on-site calibration protocol. An easy calibration setup would provide both accurate and precise results. Calibration protocols would need to use the device’s sensors to make on-site calibration quick and user-friendly.
Geolocalizing Data

Another shortcoming of the current system is that the collected data does not include geo-localization. However, a low-power GPS device could be used for localization once the node has resurfaced. Data would then be tagged with coordinates and a timestamp, which could later be used for temporal and spatial analysis.

Ultra Low Power Mode and Wireless Charging

The current implementation does not use the microcontroller’s deep sleep capabilities. Enabling such capabilities would allow for a significant extension of battery life. In addition, complementing the deep sleep feature with independent and automated switching of peripherals could be crucial for long-term missions. Finally, a wireless charging node could improve the device’s power dynamics by allowing fisheries to charge sensors overnight without opening the entire enclosure. These additions would make the device more manageable and comfortable to deploy in varying contexts.

Multi Node Deployment

With the current evaluation, we can now replicate sensor nodes to experiment and validate the star network topology. Further development of the design and implementation of the communication nodes will be required. Communication could be done through WiFi stations, GSM, or LoRa radio modules. Each option would introduce trade-offs in range, availability of signals, and delay in uploading collected data. The network architecture could allow multiple nodes to work in tandem to make transmission more efficient and introduce fewer delays in the data collection chain.

Community Deployment

Solving some of the key issues outlined above can facilitate the deployment of the device in a fishing community setting. The deployment would allow evaluation of the collective benefits of such a device. Communities can use the device to tackle
local problems according to their desires. Building on top of the presented work, a community deployment would fully demonstrate the potential of the proposed network topology and sensor node design.

3.5.2 Conclusion

This chapter presented the design, implementation, and evaluation of a distributed, low-cost ocean profiling sensor node. Experiments were carried out within three different water bodies. Collectively, the experiments allowed us to characterize and evaluate the proposed design. The experimental data reveals that the system can: (1) collect and fuse multi-sensor data, (2) cycle through different operational modes, and (3) not allow for any water leakage or damage when submerged.

Results show a promising avenue for research to create a low-cost, distributed sensing infrastructure that can inform various interested parties about how to understand and preserve marine environments. Further, these networks could become crucial mechanisms for communities to understand how their activities impact the environment collectively.

Collected data can also be aggregated and used by scientists to develop better models for climate change and possible mitigation interventions. The presented work has been left as an open-source resource to scale up the implementation of networks in future work, and to encourage community adoption of the system.

Functional socio-environmental systems that are opportunistic can be used to sense behavior and environmental conditions without disrupting the activities in which they are embedded. This extended intelligence in daily activities helps close feedback loops of insight and behavioral transformation. Such loops are essential for driving change in communities. The community network presented in this chapter can be expanded into different use cases within oceans and outside of them. Finding opportunities and designing the proper hardware and community dynamics to foster virtuous behavioral cycles is essential for balancing socio-environmental systems.

This idea is further extended in the next chapter, which explores how adaptability and scalability for multi-functional systems can be addressed not only through
hardware design but also through implementations of machine intelligence. These im-
plemetations add adaptability to hardware devices, increasing their flexibility when
faced with changing or novel use cases.
Chapter 4

Chameleon: Adaptive Sensor Intelligence


4.1 Overture

Allowing communities to actively change their behaviors can help remediate immediate behavioral impacts. In addition to transforming behavior and decision-making, predictive algorithms can help to foster sustainable actions by giving communities information about the possible future states of a given environment. This chapter presents Chameleon. The project proposes a software architecture that can be used to create adaptive sensor intelligence for buildings. This adaptive intelligence can impact how we use spaces in the near future by improving building energy consumption, improving user experience within built environments, and allowing for efficient use of building resources.
As workplaces, study spaces, and general public spaces become more dynamic and diverse, it is crucial for users and communities to have tools that can adapt to changing behaviors and give predictions that can better inform building managers, planners, teachers, and household leaders about the current and future states of the spaces we inhabit. Furthermore, beyond implications for the built environment, climate change will require scalable, privacy-preserving systems that can quickly adapt to changing environments and behavior patterns.

The chapter presents the complete system architecture detailing information about the prototype hardware and software. The full system was evaluated in two distinct locations to validate the adaptability, scalability, and accuracy that it can yield. The system was used to monitor room activity in an office and a classroom. Nevertheless, it is essential to note that the same adaptive sensor intelligence can be adapted to different use cases such as micro-climate forecasting, predicting public space usage, natural disaster and resource management, and monitoring of infrastructure degradation.

### 4.2 Background

#### 4.2.1 Context

Understanding urban infrastructure and its use can help us create city spaces that improve citizen well-being and a city’s sustainability efforts [85, 164]. The fields of Ubiquitous Computing, Ambient Intelligence, and the Internet of Things, in the context of smart cities, aim to do so through the use of sophisticated sensor network infrastructure installed at varying city scales [70, 131]. As the field enters maturity, research achievements range from the use of sensor networks for detecting specific activities being carried out within homes [84, 100, 173] to the monitoring of street and mobility infrastructure states in neighborhoods and highways [81, 151, 159].

Classrooms, offices, and homes represent some of the most relevant spatial scales for sensing in cities. A large body of research has been dedicated to activity recogni-
tion and occupancy estimation within these places because these are spaces that can have a great impact on sustainability and citizen well-being [35]. Systems that can identify activities and occupancy of a space properly could help dynamically improve the comfort of the space by changing light hue, ventilation, temperature, and even their structure. In addition to bringing more comfort, the spaces can also become more sustainable by reducing their energy consumption footprint [70].

Understanding human behavior along with its impacts within classrooms, offices, and homes is complex because spaces and people are continuously evolving and changing in significant ways [74]. A person or a group of persons will not interact with a space in the same way they interacted with it a year, a month, or even a week before. Furthermore, trends in multipurpose and flexible living [31] and working spaces [76] make behaviors and spaces change even more drastically [5, 69]. Lastly, significant and unexpected changes like the ones triggered by natural disasters or pandemics can strongly change the way collectives interact with the built environment [65]. Changes in human behavior and patterns of use of a space make it clear that there is a need for highly dynamic and robust systems to monitor internal activity states within said spaces.

4.2.2 Related Work

The last decades have seen the development of a large body of work that uses machine learning and live multi-sensor input to estimate or classify activity states in buildings [169]. Models created on most publications use supervised learning techniques, physics based modeling, artificial neural networks (ANN), or statistical methods such as Markov Chains, Bayesian Networks, Support Vector Machines and Logistic Regressions [22, 29, 80, 88, 144, 187, 199]. Hybrid models have also attracted interest from the research community. Implementations include the use of hybrid feature selection models [115], the combination of swarm optimization with ANNs [45] and merging ANNs with physics models [165].

In terms of sensors used to create the training and input data, the most commonly used ones are temperature (external and internal) [128, 181, 197], humidity [128], $CO_2$
light and passive infrared (PIR) sensors. Furthermore, several studies make use of sensor fusion to combine multiple sensor inputs and make predictions more robust.

While the field has yielded remarkable advances for highly specialized systems, there are still challenges remaining in terms of deploying sensor networks that are not constrained by particular working conditions. Specific working conditions include sensor placement, room size, space layout, seasonal and geographical locations, and types of activities carried out within the space in which they are deployed. Challenges tend to arise when implementing the same system in different rooms or use cases. The previously mentioned constraints make current deployments heavily reliant on intensive sensor calibration, time-consuming labeling of the data, maintenance, and costly multi-node infrastructure.

Multiple implementations have solved some of these challenges, but they all come with undesirable trade-offs. For example, synthetic data trained algorithms have been used to minimize the required amount of training data. Synthetic data can solve the problem of high volumes of data for training but is not easy to apply to multiple rooms.

Other major approaches are based on physics models. Like synthetic data, physics modeling can reduce data volumes and achieve high classification accuracies but are not easy to transfer into different spaces or settings as equations need to be adjusted according to each new environment. Similarly, statistics-based approaches are heavily reliant on parameters that are specific to the deployment space. Hybrid architectures have collected interest due to their improved robustness and accuracy but have also not been demonstrated to adapt to different rooms and use cases.

Lastly, sensor fusion approaches that incorporate multiple sensor nodes, such as humidity sensors, window status sensors, wind flow analysis, and HVAC flow sensors, tend to be highly reliable and transferable to multiple spaces but are expensive to scale and hard to maintain. Literature review makes it clear that there is a trade off between adaptability, scalability and accuracies that each of this systems
can offer.

4.2.3 Contribution

In this chapter, I present the work I conducted with research collaborators on a system called "Chameleon", which uses $CO_2$ concentration and PIR data with a hybrid machine learning model to dynamically classify room activity states with an accuracy of up to 99%. The model’s novelty lies in using both supervised and unsupervised machine learning (Recurrent Neural Networks and Spectral Clustering, respectively) to create clusters, training labels, and live classifications corresponding to different levels of activity within a room.

This hybrid approach enables the system to adapt accordingly to different rooms and routines across time. In addition to adapting over time, the system does not require the storage of large amounts of data to function, making it an effective way to monitor occupancy scenarios within office buildings, homes, or schools. Furthermore, the system is based only on two types of sensors, making it a scalable alternative when it comes to installation and maintenance.

The system was evaluated by gathering data sets corresponding to two distinct locations for four weeks, resulting in a total of 8 weeks of data. The locations vary in shape, size, layout, and location. The first location is a closed office space, and the second location is a classroom within a school. The collected data is used to train and verify the performance of our models.

We first evaluate the capability of the clustering algorithm to create clusters representative of activity states. We define activity states as the different cases that a room can have depending on the number of people using it and the physical actions that are being carried out within it. Secondly, we evaluate the performance of the Neural Network in terms of its prediction accuracy and error in classifying activity clusters. Lastly, we evaluate the performance of the system as a whole.

We conclude that the proposed system can (1) define activity clusters (See section 4.4.2), (2) self train on them, and (3) achieve a high classification accuracy (See sections 4.4.2 and 4.4.2). Furthermore, the system can keep a high classification
accuracy when training on small data sets (seven days’ worth of data) and shows the capability of adapting to rapidly changing environments in a minimally expensive way (hardware and software-wise). The above conclusions allow us to demonstrate Chameleon’s potential for becoming a reliable alternative for scaling room activity sensing infrastructure and building intelligence. The system proves to be promising in industry and academic applications such as occupancy monitoring for automated HVAC systems, adaptive lighting and ventilation for classrooms, and monitoring the health of a space for user well-being.

4.3 System Design

The Chameleon sensor system is composed of three main parts. The first is the sensor board, the second is the database, and the third is the computation. The sensor board module is installed within the room of interest and is responsible for collecting and saving the data on the data storage module. The machine intelligence module then queries the data storage module for training new models and making live classifications. The proposed architecture can allow for more modules to be aggregated in the future to add capabilities for actuating external systems (robotic furniture and built infrastructure, HVAC and ventilation, etc.) and visualization tools to the system. Figure 4-1 outlines the general architecture of the system along with possible modules that can be connected to it, and the data structures shared between them.

4.3.1 Hardware

Sensor Board Module

The sensor board module is a custom research board based on the ESP-12S WiFi chips (MIT TerMITes). The device is capable of measuring three-axis accelerations, ambient temperature, relative humidity, light intensity, atmospheric pressure, and infrared proximity and has been designed specifically for ambient intelligence experi-
Figure 4-1: Full system architecture showing the sensor, data storage, machine intelligence, visualization, and actuation modules. The sensor module sends time-stamped sensor values every five minutes to the storage module. The machine intelligence module queries the storage module to extract seven days’ data to train a new model every week. The machine intelligence module locally stores classification models and sends live predictions back into the storage module. Classified states can also be sent to visualization and actuation modules to inform users about the environment or autonomously modify it.

As the device does not have built-in \( CO_2 \) and PIR sensors, we use its peripherals to add \( CO_2 \) and PIR sensing capabilities onto it. While the \( CO_2 \) modules are generic, the TerMITes do have a dedicated purpose-built circuitry to power and handle signals from specified PIR modules.

**PIR Sensor**

We make use of a generic PIR module based on the AM312 low power PIR sensor module manufactured by NANYANG SENBA OPTICAL AND ELECTRONIC CO. A PIR sensor activation is triggered by changes in infrared radiation within the field of view of the sensor (100°). Changes in radiation are usually caused by a moving body that emits thermal radiation, such as humans.

It is important to note that the PIR sensor can activate with the movement of
animals such as rodents and can have false activations if placed in environments that have direct light exposure or high temperatures (above 35°C). Therefore, we mounted the sensors within spaces that meet optimal operating standards for the PIR to minimize the probability of errors produced by the environment and ensured that our mechanical casing and our board’s LEDs would not interfere with the sensor.

Correct placement was tested by allowing sensors to run in empty rooms and validating that their reported data was close to zero activations. Empty rooms show no activations on our experimental results. We define the saved PIR value by an internal counter that sums all activations that happened within a fixed reporting range of five minutes. Oversampling helps us to minimize the impact of false activations.

**CO2 Sensor**

In addition to the PIR sensor, the board is connected to a digital optical CO₂ sensor (SCD30 - Sensor Module for HVAC and Indoor Air Quality Applications) manufactured by Sensirion [172]. The CO₂ value sent by the sensor corresponds to the direct parts per million (ppm) concentration reading. The CO₂ module automatically calibrates itself for 14 days, as specified by the manufacturer [172]. Values above 400ppm correspond to changes in the ppm detected within the space where the sensor is deployed.

We note that, while studies have made use of multiple types of sensors, we make use of CO₂ and PIR because they are known to be the most privacy-preserving [3, 155]. The modularity of our system allows for future implementations to make use of more sensors or different types of sensors depending on specific use case conditions. However, future implementations should keep in mind that adding more sensor inputs can affect the system’s scalability and should only be done when necessary.

**Casing Design**

The sensor board is cased within a custom-made capsule that can be easily mounted on any flat wall. The casing was manufactured using a clear resin SLA 3D printer. The case is designed in a way that allows for the correct operation of the CO₂ and
PIR sensors. Proper ventilation and ambient exposure of the sensors minimize the possibilities of an error in the data being collected. Figure 4-2 shows the sensor module as manufactured and installed.

![Image of sensor module with casing](image.png)

Figure 4-2: Image of a sensor module with casing. The Figure shows the $CO_2$ sensor module connected to the WiFi-enabled microcontroller. We also show ventilation for the $CO_2$ sensor as well as an outlet for exposing the PIR sensor. The sensor can be mounted on any flat wall and has an outlet for powering the device through a micro USB port.

### 4.3.2 Software

**Data Storage Module**

Data is stored within a PostgreSQL database running on a Linux-based operating system. The module is queried for (1) training the hybrid model and (2) for live classification once a model has been trained. As indicated on Figure 4-1, the database stores all the data coming from the sensor board. This module can be cleaned every time a new model is created to keep privacy and security concerns at a minimum. It is relevant to note that localized computing would further tackle such concerns.

As we needed access to historical data for our system evaluation, we did not erase any data instance but we note that deleting data would not affect the system’s performance. The data storage module can save data from multiple sensors and allows for querying based on sensor module ID and specific date ranges. We only used two
sensor inputs to prove proper system operation, but note that more sensors can easily be added in future versions of the system. Keep in mind that sensors must be mounted on the same board so that the system maintains its scalability characteristics.

**Machine Intelligence Module**

The machine intelligence module runs the hybrid machine learning architecture designed to (1) cluster different activity states, (2) train a Recurrent Neural Network (RNN) with clusters made from incoming data, and (3) make live classifications of the state of a room using the trained models. This module periodically queries data from the storage module to create models that can classify activity states within a room. Our implementation queries the database every seven days. We chose seven days to properly visualize changes in our models and room usage patterns every week. We note that query intervals can be modified depending on the function and schedule of a given room, keeping in mind that the smaller the interval, the more accurate the representation will be.

Lastly, the machine intelligence module would be the component that would interact with the actuation and visualization modules seen in Figure 4-1. Interaction with visualization and actuation modules would allow the system to become a closed-loop control system for dynamically regulating comfort and efficiency within different use contexts and rooms.

**Hybrid Machine Learning Architecture**

This section introduces the hybrid machine learning system for dynamically understanding the state of a room. The system can continuously monitor and classify different activity states of a room according to sensor data coming from a CO$_2$ and PIR sensor. We define activity states as the different cases that a room can have depending on the number of people using it and the physical actions that are being carried out within it. The cases are represented by different combinations of PIR and CO$_2$ data (eg. High CO$_2$ and High PIR, Low CO$_2$ and High PIR, etc.).

The current demonstration uses only two sensor types, but it could allow for more
sensor inputs through minor modifications to the clustering algorithm parameters and the neural network input layer. Adding more types of inputs could further improve reliability but is considered to be beyond the scope of the current contribution and is left for future work. Nevertheless, it is crucial to integrate extra sensors keeping in mind the trade-offs between system accuracy, sensor inputs, and scalability.

The system uses small amounts of data (maximum of 2,016 independent instances corresponding to seven days worth of data) to create clusters and classification models automatically.

We define the system as hybrid because it uses supervised and unsupervised learning techniques. At a high level, the system utilizes spectral clustering to create a set of clusters that represent environmental instances or combinations of $CO_2$ and PIR data that represent different activity states of a room. The relationship between variables is dependent on the sensor’s data and on the activities that are being carried out within the space used for testing the system. Clusters are then used to create a new data set of labeled data and fed into a Recurrent Neural Network that takes sensor data as inputs and clustered labels as training outputs. Figure 4.3.2 shows a detailed diagram of how the architecture functions over time.

**Spectral Clustering**

Spectral clustering has received great attention over the last few years as a highly reliable clustering algorithm [180]. We chose spectral clustering due to the algorithm’s capabilities to create clusters that are not only based on basic similarity calculations but can include convex boundaries, which are related to relationships between data points with higher complexities [109]. This allows us to weigh the model properly when using multiple sensors as inputs.

Our implementation of spectral clustering takes an input as matrix $M$ with shape $[2, n]$ where $n$ represents the number of examples taken into account to train the model. Each of the rows or examples of the input matrix corresponds to the normalized $[0.0\,\text{to}\,1.0]$ sensor values for $CO_2$ concentration and PIR activations.

For example, knowing that a new data instance is saved on the data storage module
every five minutes, if we train the model using seven days of data, we would have an
\( n = 2016 \). The size of \( n \) is a parameter that can be used to adjust the days of past
data stored and used by the system. A large \( n \) would mean that we can access older
data instances, making our training set larger, but it would compromise on privacy
aspects. A small \( n \) would make for a highly reactive and privacy-preserving system
but would compromise on performance. The size of \( n \) can be modified depending on
the use case for the system or user preferences. In section 4.3.2, we refer to \( n \) as the
time range used for training and state that our current implementation is set up to
draw seven days of information every seven days.

We use SciKit learn’s spectral clustering function [101] which uses a Gaussian
Kernel to create the affinity matrix, which is then used for computing similarity
between points and defining clusters. The Gaussian Kernel is defined by

\[
s(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}
\]  

(4.1)

Where \( x_i \) and \( x_j \) represent individual data instances and \( \sigma \) is the width of the
kernel and is used at its default library value [101]. The Gaussian Kernel that we
use computes the euclidean distance between data points to get a similarity score for
each pair of data instances based on proximity.

The algorithm uses the affinity matrix generated with the Gaussian Kernel to
create clusters that minimize similarity between each other. Our implementation
uses a target of five clusters to illustrate the system’s capabilities. However, cluster
numbers can be varied according to specific use cases (See section 4.4.2 for types of
use cases). For example, some applications might require more granularity (a higher
number of clusters).

We compared results obtained through spectral clustering against the K-means
clustering algorithm to visualize the difference between the results. Our results proved
that spectral clustering was superior to K-means because it could fuse information
from the two sensors with proper distribution.

Figure 4-4 visualizes results obtained from running spectral clustering and K-
means clustering on seven days’ worth of data. From the first four graphs [1-4], we can see that both algorithms can distinguish unique states within a room, guided by PIR and CO₂. The spikes on CO₂ and PIR activations seen in [1-4] are related to the activity in the room. Both algorithms can detect moments of high activity and cluster inactivity properly on days that correspond to weekends (days 4 - 6 on the horizontal axis). As values are normalized, we can talk about high, medium, or low activity independent of the type of room that the system is being deployed in. The last row of graphs (5-6) gives us insight into how spectral clustering combines multiple sensor inputs better than K-means. From graph 5 (K-means clustering), we can see that at low levels of PIR activations, we have three distinct clusters (purple, yellow, and green); at medium levels, we only have one cluster (blue), and at high levels one cluster (red). In contrast, within graph 6, we can see that low PIR activations have two clusters (purple and yellow), medium PIR activations have two clusters (green and blue), and high PIR activations have one cluster (red).

Having clusters evenly distributed along PIR activations suggests that the algorithm has a more balanced result and can weight the information from both sensors evenly. This is relevant because the algorithm would be better suited for clustering special cases like having people within a space with good ventilation and open windows (high PIR, low CO₂) or having a poorly ventilated space not being used by anyone (high CO₂, low PIR). In this way, we conclude that spectral clustering can make the most of fusing multiple sensors.

While spectral clustering is superior to other clustering algorithms in terms of its results, it has a trade-off in performance. Some of the spectral clustering techniques are not optimal for large data sets nor for live classification or execution tasks [109]. This is why we use the results from spectral clustering to train a Recurrent Neural Network instead of directly using spectral clustering for live classification of new data instances.
Recurrent Neural Network

A Recurrent Neural Network is a type of neural network that allows for the inclusion of the concepts of time and memory to be taken into account for classification and prediction [108]. The last couple of years have seen the adoption of these systems primarily for natural language processing (NLP) and financial time series applications. In general, RNNs are optimal for classification and prediction tasks that involve sequential data [194]. Time-series data is crucial to understanding how people interact with a space. How a space is currently being used will affect how it is likely to be used in the upcoming future. This was the driving assumption for using an RNN as our classification mechanism.

Network Architecture and Hyperparameters

We employed the popular TensorFlow library [33] for our RNN implementation. Our architecture consists of six hidden layers, in addition to input and output layers. The input layer receives a tuple of values corresponding to the normalized sensor values with shape $[\text{PIR, CO}_2]$. The first three hidden layers are long-term short-term memory machines (LSTMs) with 256 neurons and one hidden state. LSTMs have been shown to have high performance when learning from sequential data through the implementation of memory cells and gate units [195]. The final three hidden layers correspond to three fully connected feedforward layers with 500 neurons each. All hidden layers have a rectifier linear unit (RELU) as their activation function. The RELU activation function is defined by

$$f(x) = \max(0, x)$$ (4.2)

where $x$ is the input that each neuron receives. RELU outputs the same number that it receives as output as long as it is larger than 0.

Lastly, the output layer has five neurons corresponding to the five clusters defined by spectral clustering. The number of clusters and number of output neurons should always be equal. The output layer uses a softmax activation function that creates a
vector with the same size as the number of clusters and assigns a probability between 0 and 1 that represents the likelihood of a given example belonging to a class (1-5). Figure 4-5 illustrates the Neural Network architecture.

We use manual search to define our network’s hyperparameters. Our implementation runs training with a 70%-30% training and test data split respectively. The models are trained for 50 epochs with a batch size of 64 instances. The Adam optimization algorithm is used as our weight optimizer with a learning rate of .001 and a decay rate of $1 \times 10^{-6}$. We initialize all weights randomly at the beginning of every training sequence. Future implementations could save the current model and use the current weights when initializing training. This change could bring better processing times for the system but was not tested. Dropout layers are incorporated after every regular layer with a dropout rate of 0.2; the layers are placed to avoid overfitting.

4.4 Validation

4.4.1 Deployment

Identical versions of Chameleon were deployed in two distinct scenarios to evaluate their performance. We chose two distinct locations to evaluate different aspects of the system. The first location was an office space with low usage (maximum of five persons used the space within a week). The office space is situated within a research lab at a university in the USA. The second location was a school classroom with high usage (the headcount within the space would typically be 15-20 students). The classroom is located in a public school within the Principality of Andorra. The rooms are entirely different in location, use, capacities, layouts, people flow, and ventilation.

Sensors were placed at the height of 1.5 meters above the floor, next to the entrances of the room in a way that the PIR could have a field of view that overlapped with the center of the room and that the $CO_2$ sensor could be properly exposed to the environment. The systems were deployed for four weeks to capture naturally occurring behaviors within the space. Set up of the devices is carried out with a
living laboratory approach to induce the least amount of measurement bias into the behavior of the people using the space [83].

4.4.2 Data

We first evaluate the capability of spectral clustering to create clusters that are relevant to space usage and activity within the two rooms. Next, we evaluate accuracies obtained by the Neural Network being trained on the clusters coming from the spectral clustering algorithm. Lastly, we visualize the live classification outputs and model confidence for each classification. Classification outputs would be the outputs going to the visualization and actuation modules as explained in section 4.3.

Clustering Algorithm Performance

We first evaluate the capability of the clustering algorithm to create clusters representative of activity states. To do so, we plot out results corresponding to 4 weeks of operation for both of the devices. Figure 4-6 illustrates clusters normalized plots showing data collected for the two sensors along with clustering results for each. We can see periods of inactivity are correctly labeled with a single color, whereas spikes in PIR and \( CO_2 \) contain different color clusters corresponding to different states in a room.

The same system was used for the different rooms without changing the deployment procedure. From Figure 4-6 we can conclude that the system can adequately cluster and map different activity states within rooms with different usage, ventilation, and layout. We can also see how the different cluster models can evolve according to the different patterns of use and to different \( CO_2 \) baseline levels. Experimental results along with literature supporting the correlation of \( CO_2 \) with room occupancy and activity [22, 80, 88, 187, 199] allow us to validate the capability of the system to identify activity states and to adapt to multiple environments and scenarios.
Neural Network Performance

Along with proper cluster performance, we evaluate the capability of the system to retrain itself with new clusters every week. Our results show that the Neural Network can retrain itself properly with accuracy within 87%-99% on test and training sets. We evaluate each generated model with a 60%-10%-30% data split (training-validation-testing). As mentioned above, each model is trained from scratch with a week worth of data. Future implementations can retrain by using weights generated by previous models. We believe this will allow us to maintain valuable information and make training more efficient and accurate. Figure 4-7 shows the plotted accuracy results on validation and test sets for models generated throughout the same weeks of testing as the ones used for our cluster evaluation. We can also conclude that the training epochs and model parameters are optimal for making proper use of limited resources in terms of data and energy used by the system.

Full System Performance

Lastly, we evaluate how the entire system works together to actively monitor the state of a room. As we outline in Figure 4-3, we run the system continuously and retrain a new classification model every week. The model is then used to classify incoming live data. Live classification results would later be fed into the visualization or actuation modules to change the state of the room (improve ventilation, send user safety alerts, etc). During our four-week experiment, we saved all of the classifications made by each sensor module, collecting eight weeks’ worth of data. We then plot out the live classification results against a control data set constructed by only running the spectral algorithm on the data from the same time period from which the live classifications were saved. We would expect the clustering decisions made by the live classification model (trained neural network) to be highly similar to running spectral clustering on the same data. Figure 4-8 shows both plots for one week worth of data for the sensor deployed in the classroom.

With the visualization, we can conclude that clusters are being generated live
almost identically to our control set. This implies that the system can effectively use small amounts of historical data (one week) to train a hybrid machine learning model and use the model to monitor different states within a room actively.

**Benchmarking Results**

Our experimental results showed that our system can achieve classification accuracies that lie between 87% and 99%. Our system can perform in spaces with completely distinct layouts, geography, and use and can perform without the need to deploy a hard-to-scale sensor infrastructure. In Table 4.1, we compare the strengths and results of our system with the results of other state-of-the-art systems.

We evaluate each approach and method depending on the accuracy and the number of sensors used. Adaptability is defined depending if the method allows for easy operation in different environments. As stated above, physical, statistical, and synthetic data methods are not easy to adapt because they rely on knowing specific characteristics of the environment where the sensors are placed.

Scalability is evaluated depending on the number of sensor inputs and devices required to make the system work. Lower amounts of sensors make systems more scalable by reducing installation and maintenance costs. We see that multiple sensors can yield high accuracy and make models adaptive but are hard to scale, especially when sensors are spread out around a room instead of being compacted onto a single board.

From this literature review and benchmark of the system, we conclude that Chameleon performs comparably well in terms of accuracy range. Furthermore, it offers an adaptive system that is easy to scale, making it a highly viable alternative for adaptive building intelligence in the future.
<table>
<thead>
<tr>
<th>System</th>
<th>Approach</th>
<th>Method/Algorithm</th>
<th>Sensor Inputs</th>
<th>Sensor Types</th>
<th>Adaptability</th>
<th>Scalability</th>
<th>Accuracy Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chameleon</td>
<td>Hybrid Machine Learning</td>
<td>Spectral Clustering RNN</td>
<td>2</td>
<td>CO₂, PIR</td>
<td>High</td>
<td>High</td>
<td>89%-99%</td>
</tr>
<tr>
<td>[199] Physical Modeling</td>
<td>Statistical Modeling</td>
<td>Dynamic Mass Balance ANN SVM</td>
<td>3</td>
<td>CO₂</td>
<td>Low</td>
<td>High</td>
<td>70%-76%</td>
</tr>
<tr>
<td>[22] Statistical Modeling</td>
<td>Statistical Modeling</td>
<td>LDA RF CART</td>
<td>4</td>
<td>Light Temperature, Humidity CO₂</td>
<td>Low</td>
<td>Low</td>
<td>95%-99%</td>
</tr>
<tr>
<td>[22] Statistical Modeling</td>
<td>Statistical Modeling</td>
<td>LDA</td>
<td>1</td>
<td>Temperature, CO₂</td>
<td>Low</td>
<td>High</td>
<td>83%-85%</td>
</tr>
<tr>
<td>[187] Physical Modeling</td>
<td>Physical Modeling</td>
<td>Dynamic Mass Balance</td>
<td>1</td>
<td>CO₂</td>
<td>Low</td>
<td>High</td>
<td>88%-94%</td>
</tr>
<tr>
<td>[115] Hybrid Machine Learning</td>
<td>Physical Modeling</td>
<td>PI-PRM</td>
<td>2</td>
<td>CO₂, Temperature</td>
<td>Low</td>
<td>High</td>
<td>97%</td>
</tr>
<tr>
<td>[45] Hybrid Machine Learning</td>
<td>Hybrid Machine Learning</td>
<td>Particle Swarm Optimization ANN</td>
<td>4</td>
<td>Light Temperature, Humidity CO₂</td>
<td>High</td>
<td>Low</td>
<td>81%-98%</td>
</tr>
<tr>
<td>[107] Machine Learning (Simulation)</td>
<td>Hybrid Machine Learning</td>
<td>SVR RNN</td>
<td>1</td>
<td>Temperature, CO₂</td>
<td>High</td>
<td>High</td>
<td>96%</td>
</tr>
<tr>
<td>[184] Synthetic Data</td>
<td>RNN</td>
<td>RNN</td>
<td>1</td>
<td>CO₂</td>
<td>Low</td>
<td>High</td>
<td>89%-98%</td>
</tr>
</tbody>
</table>

Table 4.1: Performance benchmark for state of the art activity and occupancy classification systems.

4.5 Discussion

4.5.1 Limitations & Future Work

Adaptively monitoring activity states within a room can be a challenging problem involving high-performing hardware and software design. We believe Chameleon is a big step, from a system perspective, towards making building sensing more resilient and scalable. However, the current version of the system is still limited in some functions.

Granular Activity Recognition

Chameleon’s implementation can detect general room states, but it cannot recognize specific activities being carried out within the room. Doing this requires using ground truth labels that can help indicate the correlation between a specific action or number of people with the sensor readings. While our clustering results show great potential, granular activity recognition with Chameleon remains a challenge to be solved and an important milestone to tackle in the future.

Approaches to reaching granular activity recognition could require using a larger array of sensor types and the number of clusters that the spectral clustering algorithm
Connection to Actuation and Visualization Systems

The current version of Chameleon does not have visualization or actuation modules. Creating these modules would allow Chameleon to become a complete closed-loop control system. Such a system would need to be connected so that safety mechanisms and a larger array of inputs for decision-making are taken into account. Creating a central decision hub that can take as inputs information coming from a system like Chameleon and other building systems could be an essential tool for making building sensing as adaptive, modular, and scalable as possible.

Multipurpose Adaptability

We presented Chameleon and the use case of detecting activity intervals within a room. While this is a relevant problem, many other room monitoring applications are relevant in the context of smart buildings. We did not explore any other monitoring but note that other types of monitoring could be carried out with the same system. Chameleon’s modular architecture can allow for integrating different types of sensors. Different sensor combinations could make a more granular class set or monitor things like environmental performance, structural behavior, or granular human behavior.

Specialized sensors could also be used to use our contribution to open space monitoring; such experiments could help us to better instrument public spaces in cities. It is relevant to note that the CO$_2$ sensor used for this research does not work for outdoor environmental monitoring. Hardware would have to be significantly adapted to adjust the system to outdoor environments.

Interactions with HVAC and Ventilation Systems

The system cannot accurately characterize the precise interactions that it might encounter with different HVAC and ventilation systems. While these effects were not
evaluated in the current implementation, we realized that they were observable in the
decay times for \( CO_2 \) concentration seen in the clusters that the system can create.

Further knowledge about specific interactions could make the system more adapt-
able to spaces and countries with distinct ventilation and equipment regulations. The
authors note that further development of the system could take into account the decay
rate of \( CO_2 \) and infer ventilation dynamics from raw data and cluster outputs.

**Local Data Privacy**

Our implementation does not address any encryption or data security aspects. While
the system uses small amounts of data for training, it does not encrypt any infor-
mation. Adding an encryption layer could make the system more privacy-preserving
and safe against malicious operators. One of the system’s major values is that it uses
small quantities of data to create classification models. This allows computation to
be carried out locally on the deployment side.

Local deployments and encryption could make for a privacy-preserving and safe
building activity recognition system. In addition, making modifications to the initial-
ization of training weights could free up processing power to handle encrypted data
while keeping the same performance level.

4.5.2 Conclusion

Adaptive monitoring of the activity and use of a space has been a major challenge
for sensing in smart buildings. Current implementations face a big challenge when
making systems that will function within different environments and patterns of use.
Long calibration protocols along with acquisition of labeled data sets make for major
hurdles in scaling these systems. Put simply and based on our literature review, we
identify a major obstacle as the trade-off between accuracy, scalability, and adapt-
ability of current systems. Therefore, a system that can scale well, adapt to different
deployment spaces, and maintain high classification accuracy is valuable to enable
these systems to become more ubiquitous.
Making these systems increasingly available for many types of uses can significantly impact user comfort and sustainability efforts. Smart spaces can enable efficient actuation operation for HVAC systems, motorized windows, blinds, and even completely transformable architecture.

With these issues and goals in mind, we present a hybrid machine learning architecture along with a sensor fusion system that is able to (1) define activity clusters within two types of rooms, (2) self-train on them and (3) achieve a high classification accuracy, with a range of 87%-99%. The system can keep a high classification accuracy when training on small data sets (seven days’ worth of data) and shows the capability of adapting to deployments on distinct environments in a cost-effective way (hardware and software-wise). The system uses an integrated sensor board to keep its scalability viable and introduces a modular architecture that allows for well-directed future improvements.

We evaluated the system through a living laboratory setting where sensors were placed in an office space and classroom with distinct window and space layouts to test the system’s adaptability. The relationship between readings and activities has been justified by literature, and the hybrid machine learning algorithm was evaluated quantitatively and qualitatively through its accuracy metrics on different weeks, a benchmark with current state of the art, and through visualization of clusters being made by the system in real-time.

The present chapter expands characteristics that community-scale systems must have at the software and hardware levels. It also demonstrates how machine intelligence can be used to make systems that are highly adaptive and scalable for multiple use cases. As explained in the last chapters, systems that are deployed at the community scale are closely ingrained into our everyday lives. This fact calls for the need of tools that can bring technological literacy to communities. The next chapter outlines how literacy can allow communities to control and develop their own technological interventions through the implementation of a low-cost and modular bioreactor.
Figure 4-3: Hybrid machine learning implementation. Data is acquired for seven days (acquisition range can be varied depending on application) and saved on the data storage module. On the seventh day of acquisition, the machine intelligence module queries servers and uses spectral clustering to create clusters corresponding to different room activity states (e.g., high, medium, low). The clusters are then used to label a new data set, called DA, and fed for training into a Recurrent Neural Network that takes sensor data as inputs and clustered labels as its training objectives. Creating a new RNN from new spectral clustering results takes between 5 - 15 minutes (training time depends on the time range selected, our implementation uses seven days as the default time range). The new model is then used for live classification of room states. The process repeats every seven days to update new behaviors or activities within the space. The updating range can be varied depending on specific uses or rooms. When the system is in the live classification state, data is relayed into the data acquisition state to restart the process and readjust models.
Figure 4-4: Clustering results comparing K-means clustering and spectral clustering. We used five clusters as our standard to visualize the key differences between the two algorithms. Each color indicates the cluster to which the data point belongs. Values for PIR and $CO_2$ are normalized before being used by each of the algorithms.
Figure 4-5: Recurrent Neural Network architecture. The network is composed of 8 layers. The input takes two inputs corresponding to sensor variables, hidden layers are divided between LSTM’s and regular feed forward layers. The output layer has 5 neurons corresponding to the 5 clusters created by the spectral clustering section of the system.
Figure 4-6: Spectral clustering results for four weeks’ worth of data. The plots visualize how clusters are properly created following sensor spikes and activation activity. We show a plot with $CO_2$ readings, one with PIR readings, and a final one that plots $CO_2$ vs. PIR to prove that clusters are well defined. It is relevant to note that differences in color scheme for each plot are due to the way that the spectral clustering algorithm assigns cluster centroids.
Figure 4-7: Accuracy plots for validation and test sets along 50 epochs of training for models generated from the office and classroom sensors throughout four weeks of experimentation. We can see that all models manage to get accuracies above 87% by their final training epoch.
Figure 4-8: Plotted results from the live classifications made by a trained model running on real-time data and control results for saved data clustered by spectral clustering. We can visualize that the machine intelligence module can properly classify as expected. Data density is lower for the live classification because live classification is carried out every 15 minutes as opposed to the 5-minute interval used when saving the data used by the spectral clustering algorithm.
Chapter 5

MAIA: Distributed Citizen Science

5.1 Overture

The last three chapters showed how community based sensors and algorithms can feed insights into predictive and transformation processes that change our behavior, how we use our spaces and how we interact with the environment. Equally important, is the idea of making devices and technological implementations easy to deploy in a decentralized fashion. As urbanization proceeds and challenges continue to arise, deployment of technologies will need to be carried out in a distributed fashion. Distributed deployments allow for responses that would otherwise be hard to achieve. Current systems have reached levels of complexity that make it hard for small institutions or non-specialized collectives to implement or even understand them. It is therefore essential to create tools that can allow for complex technological interventions to be carried out by young citizens and communities.

One of the fields that will have tremendous impact on how we sense and understand urban life will be the field of Synthetic Biology (SynBio). SynBio is growing at a fascinating pace. As the field holds great promise and challenges, many efforts are being made to allow it to be as democratized as possible from its beginnings. This chapter builds on those efforts by outlining the development of a low-cost and modular platform that can be used to democratize synthetic biology experimentation.
and education.

The chapter introduces MAIA, a low-cost, modular and portable bioreactor. Specifically, the device can accelerate experimentation pipelines for DIY bio-sensing projects and become a technology development catalyst within the expanding anti-disciplinary synthetic biology community. The modular bioreactor allows for quick iteration and validation of different bio-sensor designs. This chapter describes the development, operation and validation of the reactor. Details are presented about the characteristics that allow systems to reduce barriers of entry to technological development, such as being easy to replicate, low-cost, having moderate learning curves, and having mechanisms that nurture creativity and experimentation skills.

5.2 Background

5.2.1 Context

The "Internet of Bio-Nano Things" is an emerging field that aims to develop new applications for sensing enabled by biology [4]. Publications suggest that the application of biology-based sensors can complement our current sensing infrastructure. Furthermore, it can expand our capabilities for understanding the complexity of cities [4,90,97,135,158].

Advances in bio-sensing at the scale of the city have shown promising alternatives to understanding new ways in which we can instrument our cities [63]. These novel approaches have the potential to help us better understand citizens’ needs, behavioral patterns, and their interaction with the environment [154]. Furthermore, research has shown that it is possible to understand complex socio-environmental dynamics within cities through the use of these types of systems [77,92,116]. An example of such work is the use of sewage sensing to monitor viral spread in communities during the COVID-19 pandemic [8,48].

As exponential technologies like artificial intelligence, robotics, and synthetic biology make their way into our everyday lives, democratizing their development and
deployment becomes increasingly important. Technological literacy has been shown to be key in enabling impact of technologies within bottom-up processes. Well known projects like the Scratch programming language [112], LEGO Mindstorms [91], and the Arduino platform [96], have enabled global communities to engage and develop highly complex technological implementations. This engagement can lead to the creation of hyper-local solutions to problems that a given community interacts with on a daily basis.

Similarly, a large number of foundations, institutions, devices, and programs that seek to democratize the development of Synthetic Biology are emerging. Examples include the BioBuilder Educational Foundation [51], the International Genetically Engineered Machine Competition (iGEM) [82], the Community-Bio-summit, the global How To Grow (Almost) Anything course [78], and the BioBits Educational Kit [79], amongst many others. It is widely recognized that building devices that allow us to democratize different aspects of technological development will be essential for us to rely on distributed creativity and talent to solve the great challenges that the future will have for communities around the globe.

5.2.2 Related Work

Low-cost incubators and synthetic biology equipment have been of great interest to the scientific community in the recent past [21,160,174]. Research has been carried out to develop low-cost but highly specialized machines. Some examples include human incubators [175], incubators for carrying out field studies in remote or underserved communities [186], laser-cut incubators for optogenetic bacterial culture [188], flexible microfluidic incubators for affordable, safe, and sterile, hands-on experimentation with live microorganisms [61], and hypoxia chambers for cell culture [182].

In addition to highly specialized and research-oriented incubation devices, we are also beginning to see the development of low-cost, multipurpose devices aimed at the DIY synthetic Biology and Synthetic Biology Education communities. For example, companies like miniPCR [122], AminoLabs [98], Opentrons [133], Foldscope Instruments [28], and Bioexplorer [98] are coming up with DIY and educational kits
that allow communities and students to engage with essential SynBio techniques such as DNA modification, expression, translation and assembly, thermal reactions, and electrophoresis. The proposed work aims to fill a gap currently not filled by existing systems by making a device that costs under a hundred dollars. In addition to having a low cost, the device’s modularity will allow it to have multiple functions making it adaptable for a wide range of experiments.

5.2.3 Contribution

The proposed research aims to prove the feasibility of a bio-based, living cell sensing system that is deployable outside a sophisticated wet lab setting. Such a device could lead to further developments in the field of city bio-sensing in a community-based, bottom-up fashion. The device has the goal of enabling organic community contributions and growth similar to the one shown by digital fabrication communities such as Arduino and the Global Fab Lab Network [57,96].

The chapter shows the development of a low-cost, modular and portable bio-reactor. The reactor controls bacterial growth variables conducive to rapidly deploying cell-based biosensor experiments and designs outside of expensive laboratory settings. A low-cost and open-source design allows the device to be easily replicated and modified according to different use cases, needs and contexts. The device’s modularity adds flexibility for it to be easily adapted to use cases that are limited by the user’s creativity, interests, and needs. Making the device portable is crucial for taking these systems outside of traditional laboratory settings. Portability is introduced through a design that is easy to install, self-contained, and lightweight.
5.3 System Design

5.3.1 Hardware

Modular and Low-Cost Design

Bioreactors maintain the stability of specific variables needed for cellular growth [26]. Unfortunately, most reactors act as unified systems, tending to lack flexibility, in which variables like temperature, vibrations, growth medium, and ventilation are carefully controlled [179]. MAIA modularizes each one of a classic reactor’s functions. Modularization allows each function or control variable like temperature, ventilation, and growth mediums to be independently assembled and manipulated. Added flexibility lets researchers, students, and hackers design and attach different stimulation modules according to cell type and experimental design.

Modules are designed in a "LEGO-like" manner to be stacked and ordered according to the type of culture, stimulation, and experiment layout. The design allows for every module to share power and communication lines. Shared power and communication let users run synchronized protocols from the motherboard, as shown in figure 5-1. The motherboard runs the main incubation protocols and communicate with each peripheral module through the I2C protocol, giving it the capability to maintain control of the multiple, in-sync modules that the reactor has the potential to have.

The mechanical design of the modules is standardized so that it can be easily manufactured and replicated by using commonly available fabrication techniques and components. The current evaluations used commercially available printers and materials (Prusa i3 MK3S+ and Formlab’s Form3) to produce each of the reactors base modules. Fabrication of each component is carried out strictly with tools found in the Fab Lab Foundation’s component, material, and machine inventory to keep distributed replication viable. In addition, the designs are simplified so that modules can attach without the need for careful adjustment or alignment.
Figure 5-1: Modular "plug and play" architecture allows for flexible design of experiments and growth setups. Modules are connected to shared power and communication lines and stacked accordingly to desired function and stimulation. Power, control, and temperature modules are considered the minimum assembly. Significant changes, dependant on cells and experiments, occur in the growth and multi-functional modules.

Module Design, Manufacturing & Assembly

Base Structure Module

The base module is the main building component for the reactor system. It acts as the minimal unit for creating specialized modules. The module was designed so that it can be manufactured by 3D printing it on commercially available 3D printers. The current implementation used the Prusa i3 MK3S+ [141] and Formalsb’s Form3 [50] printers to test assembly tolerances and material properties for black and white PLA, and for Formlab’s proprietary Clear Resin V4 [49].

The design can hold two PCBs, one on the top side and one on the bottom side. It also has four cross plugs that can be used to attach accessories that enhance function for the base design such as Peltier trays, PCB fasteners, ventilation lids, etc. Figure 5-2 shows the manufactured module on three different materials and some of the proposed accessories mounted on it. Notice that the same design can be used for a
wide range of structural and functional purposes within the reactor’s construction.

Figure 5-2: Base modules printed on different materials with PCB’s, growth medium and accessories mounted on them. The image illustrates the flexibility of the module and its capability to be adapted for the different functions that the reactor is required to perform.

**Power Module**

The power module supplies power to all of the stacked modules. The design has a standard JST female battery plug. Battery capacity can be varied according to experimental designs. The board also allows for connecting a USB cable’s leads connected to a standard 5V power supply such as the ones found on commercial computers. The power supply can be changed depending on the specific reactor need or experiment being run at any given time. Modifications can be made to the module if the experimental setup contains more modules and thus requires larger amounts of power. In this case, higher current supplies can be introduced into the same form factor.
Mother Module

The mother module is the central brain of the reactor. The board is based on the ESP32 microcontroller. The device is WiFi and Bluetooth enabled so that protocols can be programmed externally and commands sent wirelessly to the board. We can program protocols that run on a server and communicate wirelessly with the reactor. This allows the device to be used for a broad range of lab experiments that require variable adjustment over long periods of time.

The motherboard carries out communication with all other modules to check sensor and actuation status of all peripheral boards. Communication with all other boards is carried out through the I2C protocol in which peripheral modules act as responder devices for the motherboard. The board’s communication capabilities can allow MAIA to be integrated into friendly user interfaces for control. Figure 5-4 shows the module mounted on top of the power module.

Temperature Module

The temperature module is based on an ATTINY 1614 microcontroller. The module controls a Peltier element that is in close contact with the growth medium to provide the right temperature for cell reproduction. The Peltier is driven by a set of transistors, and an onboard temperature sensor monitors that cultures are at their required temperatures. The control programs do not run directly on the onboard
Figure 5-4: Mother PCB module with WIFI and Bluetooth capabilities placed on PLA base module.

microcontroller but on the mother module’s ESP32 based board. External modules only send their current state to the motherboard and receive specific commands to actuate and read their peripherals.

Figure 5-5: Temperature PCB module placed on PLA base module. The illustration shows the Peltier accessory that can be attached to module to hold the Peltier material on top of the device when connected.

Growth Module

The growth module is an empty base module where bacterial cultures are physically placed. The growth module can house different types of cultures in a cavity that can hold circular containers with a maximum of $80\,mm$ of diameter and $14\,mm$ of height. The current version is placed above of the temperature module so that the
Peltier element is in direct contact with the surface of the growth module. This allows bacterial cultures to have an isolated and controlled temperature chamber.

We can control variables like ventilation, light incidence and heat through the addition of accessories. All of this is controlled depending on the desired experimental setup. Future versions of the device could have active growth modules that use micropumps and microfluidics to maintain cultures alive for longer periods of time.

![Growth module assembly](image)

Figure 5-6: Growth module assembly. The structure can house different types of cultures with different growth requirements. Future versions of the device will validate liquid and microfluidic capabilities.

**Light Stimulation Module**

The light module is considered the first stimulation module built for MAIA. The past modules are considered to be standard and essential for the majority of projects that MAIA can be used for. The light module is specifically designed to test different protein expression circuits that are dependant on the incidence of specific wavelengths on the cells.

The board is based on an ATTINY 1614 microcontroller which controls three different LED arrays. The arrays shine light in visible wavelengths. The version has capabilities to shine red, blue and green light. The board is mounted on the lower
PCB mount of the base module design so that its light shines directly on the bacterial culture. This type of design could be further improved to include other types of light such as infra red or ultraviolet. It could also be adapted to create imaging systems that can monitor cell growth over time through an embedded camera.

![Diagram of PCB module](image)

Figure 5-7: Temperature PCB module placed on PLA base module. The illustration shows the Peltier accessory that can be attached to module to hold the Peltier material on top of the device when connected.

### 5.3.2 Software

**Communication Goals and Protocols**

As outlined above, the motherboard is based on a WiFi and Bluetooth-enabled microcontroller that allows users to design and execute reactor protocols directly from a web application. Wireless connectivity allows people with low computational skills to operate the device because no embedded programming is required. In addition, peripheral modules are designed to only send their current sensor and actuation states to the motherboard, and the motherboard will return commands for changing those states according to the user-specified protocol. Future versions of the device will integrate friendly Graphical User Interfaces that can allow for the device to be used by an even broader audience.

The current version of the device employs user datagram protocols (UDP) to communicate with the motherboard’s WiFi chip. The chip sets a fixed IP address
Table 5.1: Available commands processed by mother module to control light and temperature states.

<table>
<thead>
<tr>
<th>Command</th>
<th>Structure</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>T##</td>
<td>[T][Desired Temperature]</td>
<td>Sets desired temperature.</td>
</tr>
<tr>
<td>L1</td>
<td>[L][Light State]</td>
<td>Turns on Red LED’s.</td>
</tr>
<tr>
<td>L2</td>
<td>[L][Light State]</td>
<td>Turns on Green LED’s.</td>
</tr>
<tr>
<td>L3</td>
<td>[L][Light State]</td>
<td>Turns on Blue LED’s.</td>
</tr>
<tr>
<td>L4</td>
<td>[L][Light State]</td>
<td>Turns off all LED’s.</td>
</tr>
<tr>
<td>L5</td>
<td>[L][Light State]</td>
<td>Turns on all LED’s.</td>
</tr>
</tbody>
</table>

when it connects to the internet. The address is communicated by the chip through its serial port. Once the address has been found, it is used on a remote server to send protocol commands to it.

Commands are constructed by appending the module’s first letter and a number. A "T" would indicate that the command represents a temperature command. Likewise, an "L" indicates that a command corresponds to modifications to the light status. Table 5.1 shows the current command structures and actions carried out by each command. As more modules are added, the commands would follow the same structure. For example, a microfluidic module would have the letter "M" assigned to its commands. The second part of commands is defined depending on the sensing and actuating components that each module has. Protocols are constructed within a Python script that sends timed commands to the motherboard. Scripts can be executed remotely and left running on servers, helping to automate laborious protocols or experiment designs.

**Module Programs**

**Temperature Module Program** The temperature board uses a closed-loop control sequence that allows it to maintain a desired temperature. As the module only has the capability of heating up; the minimum temperature for the device is ambient temperature. The program constantly reads the thermistor’s value and activates the Peltier element as needed. The Peltier element is activated until the desired value is reached. The last desired temperature that the device receives is stored on internal
memory so that any momentary power outage does not disrupt a protocol.

**Light Module Program** The light board only has actuation elements on it. Its three outputs switch the light arrays on and off. The board constantly listens to updates on light commands, stores new values and executes them by switching specific LED’s on and off.

### 5.4 Validation

This section validates (1) the bioreactor’s technical performance and (2) the bioreactor as a tool for enhancing creativity and experimental thinking. Validation is done through two deployments. The first deployment uses multiple bioreactors to evaluate growth and stimulation of a bacterial blue light sensing system (see Light Experiment 1). The second deployment is done within a workshop. The workshop helps to evaluate performance of the device (See Light Experiment 2) and the reactors educational qualities (See Qualitative Evaluation).

#### 5.4.1 Deployment

**Light Sensor Deployment**

We evaluate the performance of the device by growing a solid agar based e-coli culture. To do so, we run two experiments. One of the experiments was done in preparation of the workshop and the second one was carried out in the workshop with participation from students.

The experiments validate bacterial growth and gene expression correlated with specific stimulation conditions. We place the bacterial culture on agar plates and grow them within different reactors. Each of the reactors uses different settings for temperature and light stimulation.

The experiments are run with an e-coli K12 strain transformed with a blue light sensitive genetic circuit described in [62]. The plasmid used contains a circuit with a photoreceptor and a chromoprotein reporter. Blue light activates the photoreceptor,
activation leads to production of blue pigmented proteins. The plasmid uses a combination of the mUAV plasmid [9], which contains the AmiCP gene, and the pDawn plasmid [132], which contains the YF1/FixJ light system. According to [62], the YF1/FixJ is first inserted into a pET-21(+) vector via Twist Bioscience. Restriction cloning is then used to insert the amilCP gene into the new plasmid.

Cell cultures were first grown as liquid cultures in a commercial incubator. The cultures were then transferred onto 30mm diameter Petri dishes with Ampicillin and a standard LB agar mix from Fisher Scientific [18] for running all of the experimental setups. It is relevant to note, that the 30mm diameter dishes were used but the reactor’s design is made for 15mm diameter dishes.

Figure 5-8: MAIAs setup for blue light stimulation experiments.

**Workshop**

The device was also tested in the context of a hybrid workshop imparted for the How To Grow (Almost) Anything class [78]. The workshop was held at the MIT Campus in Cambridge, MA, USA. The goals of the workshop include (1) introducing students to MAIA as a platform, (2) evaluating performance characteristics for the device, and (3) qualitatively evaluating characteristics related to how students use the device, how the device allows them to frame final project ideas, and how the device helps
them to structure their scientific and experimental thinking. The workshop had eight in-person student participants and 8 online participants. All in-person students were able to directly interact with setting up reactors and experiments. Figure 5-12 shows images of students engaging with the bioreactor during the workshop.

![Students using the reactor to set up different experiments on the MAIA platform.](image)

The workshop had three different sections. The first section of the workshop introduced basic concepts of bioreactors, and specific information about how MAIA is built, how to replicate it and how to use each one of its components. The second section corresponds to the use case implementation with light sensitive bacteria. Students define experimental setup according to the bioreactor’s capabilities and collectively indicate a hypothesis for each one of the tests. Lastly, the third section was a hands-on experience setting up six MAIA reactors and ten experiments to run. Six of the experiments were setup within MAIA reactors and the four others were set up as controls. Students defined variables, set them on each reactor, prepared their samples and installed the Petri dishes inside of the units.
5.4.2 Data

Light Experiment 1

The first experiment used seven reactors to evaluate cell growth and performance of the light and temperature modules. Table 5-10 shows the experimental setup used on each reactor. The bacterial culture was incubated using the e-coli strain described above. 25μL of culture were put into each of the agar plates and left within the reactors for 24 hours.

Results showed that eight out of the nine experiments were successful. Reference experiment table 5-10 and figure 5-11 to see results. The cell cultures reacted to the growth and stimulation devices as expected. Only one of the reactors malfunctioned. The cause is thought to be a problem in the temperature module coming from a faulty temperature sensor connection. The device maintained the Peltier active all the time and burned the agar plate.

We can see that all of the plates that had the light stimulation module on top of them, had an uneven growth on the plate. This is thought to be because of the direct contact heat from the light stimulation module. The experiments were done with bigger agar plates than the reactor was designed for, making them have direct contact with the light stimulation PCB. This is a highly relevant variable to consider for future improvements and future stimulation modules for the device. The ideal Petri dish size is 15mm of diameter.

The experiment’s results allow us to conclude that the reactor is able to grow cell cultures and vary the light stimulation that it can give to evaluate the performance of a light sensitive system on e-coli. Cell cultures grew as expected and gene expression was seen on all of the experiments that had blue light stimulation. Experiments with stimulation of other wavelengths correctly showed cell growth but no gene expression.

Light Experiment 2

The second experiment was carried out within the context of the workshop. The bacterial liquid culture was prepared as outlined above. For this experiment we used
Table 5.2: Experimental setup and results for the first experiment to evaluate cell ground and gene expression on bioreactor. Light value key: B = Blue, G = Green, R = Red, N = No Light, E = Exposed (white light). Recator key: M = MAIA reactor, O = Outside (no incubator), I = Commercial incubator. Temperatures are expressed in degrees Celsius.

<table>
<thead>
<tr>
<th>Plate Number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactor</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Light</td>
<td>B</td>
<td>B</td>
<td>N</td>
<td>B</td>
<td>N</td>
<td>R</td>
<td>B</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>Hypothesis</td>
<td>Blue Growth</td>
<td>Blue Growth</td>
<td>Growth</td>
<td>Blue Growth</td>
<td>Growth</td>
<td>Blue Growth</td>
<td>No Growth</td>
<td>No Growth</td>
<td></td>
</tr>
<tr>
<td>Result</td>
<td>Blue Growth</td>
<td>Failed (Burned)</td>
<td>Growth</td>
<td>Blue Growth</td>
<td>Growth</td>
<td>Light Blue Growth</td>
<td>Blue Growth</td>
<td>No Growth</td>
<td>No Growth</td>
</tr>
<tr>
<td>Success</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Figure 5-10: Experiment 1 results. Six out of seven reactors worked properly. Cell growth and gene expression happened as hypothesized.

10μL of culture on each Petri dish and allowed cultures to grow for 48 hours. Students were tasked with designing their own experimental setups and variables. Table 5.3 outlines the experimental setups chosen by students. This batch of experiments included positive controls grown on the lab’s controlled temperature chambers.

Results show that nine out of the ten experiments worked as hypothesized. We can see that bacterial growth is less dense than than growth from experiment one. This is because we used a smaller concentration of cells for each of the plates, This is also reflected on the intensity of color for the colonies.

We see that all cultures grown on MAIA reactors have the same uneven growth that we got on the first experiment results. As the experiments used the same agar plates, uneven growth is still attributed to direct contact with the light stimulation module. Nevertheless, both results allow us to validate that the reactor’s temperature
Table 5.3: Experimental setup and results for the second experiment to evaluate cell ground and gene expression on bioreactor carried out by students. Light value key: B = Blue, G = Green, R = Red, N = No Light, E = Exposed (white light). Reactor key: M = MAIA reactor, O = Outside (no incubator), I = Commercial incubator. Temperatures are expressed in degrees Celsius.

control and growth modules, maintain an ideal temperature to allow for cells to grow. The light isolation and stimulation also had positive results on experiment two as blue protein pigments can be observed on stimulated cultures.

With this experiment we can also empirically evaluate that the reactor can be successfully operated by students as all of the cultures that they set up showed bacterial growth. In the results, we see variation in uniformity and density of cultures, this is attributed to the variability that is introduced as students set up their own pipettes, load the cells onto the dish, and insert each dish into a reactor. With this we can conclude that both of the experiments allow us to verify correct performance of the device.

Qualitative Evaluation

Student interviews were carried out after the workshop. Interviews were non structured conversations that allowed us to evaluate the perception of students on the devices that they used during the workshop. All of the conversations were recorded so that the author could extract key points that help to validate the bioreactor’s citizen science oriented goals. Students also documented their workshop experience within their class webpages. Documentation helped us to see the potential impact of the platform as students are able to process their new knowledge and integrate it with their previous and future work for the class.
Information from interviews and documentation validates that the bioreactor’s design and module functions are easy to understand. Its agnostic design also makes it easy for students to relate their device to their own projects and take module inspirations into their own work.

"I really enjoyed learning and using the bioreactor because it is highly related to my projects. It helps you to fix an environment for an organism to grow."

"The system is pretty cool! It consists of stackable modules with ESP32s (WiFi enable microcontrollers) that have different functionalities; e.g. temperature control, LEDs."

"I learnt a lot from the bioreactor. I will integrate the temperature module into my final project."

A participant went as far as proposing new modules and potential uses that he would be interested in developing for his future research.

"Potential alternative modules could include chemical modules for controlling or creating unique environments for bacteria. Maybe say a CO2
emitting module or say Methane chamber or bacterial testing. Makes me think of this company called C2Sense and how they create environmental chemical nanosensors for monitoring chemical properties of controlled environments (i.e., chicken farms)."

Students see value in the modularized design as it helps them to structure their experiments taking into account the different variables that they need to control to test their designs. It is clear that the device aids students on the development of their scientific and creative thinking processes.

"The bioreactor is interesting in the sense of compartmentalizing the different elements of an experiment that you are doing. Isolating each component and creating an environment that you are studying is super useful! That idea can be applied to many other devices. The small modules also act as an interesting framework for thinking. "

"Hearing the process of designing the modules and the inspiration from Arduino, smartphones and LEGO was helpful. It was also helpful for designing experiments, knowing the variables of interest and parsing them out onto modules helps to think through projects. "

Furthermore, student interests and critiques of the reactor show how community based development could lead to use cases of the reactor that where not previously thought of.

"One of the limitations of the bioreactor is that it makes sense in a lab but it would be interesting to imagine bioreactors that could work outside of a bio lab context. Maybe you need to culture your kombucha in a more controlled environment or perhaps you want to test different parameters for kombucha growth at home. Or say you want to create at-home “coolers” for containing your probiotic foods or say different types of fermentation chambers. "
Lastly, several students went on to incorporate the reactor into their project proposals and research. Students created their own bioreactor designs based on MAIA along with modules and extensions that will be useful for their projects. One of the students created modules and kits that would allow her to carry out biocementation experiments. Another student used modules to help her develop temperature sensitive textiles. A third student sketched out the design of a bioreactor for creating bio-tinted silk. Figure 5-12 shows some of the designs, adaptations and sketches that the students made.

Figure 5-12: A) Sketches of modular bioreactor modules for production of bio-tinted silk. B) Biocementation module prototypes. The designs include improved ventilation modules and a tool carrying module for portable experimentation. C) Bioreactor adaptation for testing temperature sensing bacteria grown on textiles.

5.5 Discussion

5.5.1 Limitations & Future Work

Technical Limitations

The device shows limitations on limits for temperature. The current program allows for temperatures of up to 99°C to be used. Higher temperatures can be achieved by changing the printing material for the base module. Current used materials deform when exposed to high temperatures for prolonged periods of time.

Another limitation of the device is that connections between modules are currently done through female and male JST battery plugs. The plugs need to pass and run
cables through each module vertically. This decreases the ease of assembly for the reactor. While it remains easy to assemble, the plugs add a layer of complexity that can be tackled by improving the connection mechanism used. Magnetic connectors like the ones used for charging modern portable devices, such as smart watches, can be introduced onto the base module’s design to make each module easier to connect and change.

One of the biggest contributions of MAIA’s design are its modular characteristics. The present work explores the use of a light stimulation module. Future iterations of the device will use a wider variety of stimulation modules. The stimulation modules can be specifically designed depending on different experimental needs that researchers or students might have. As mentioned in the qualitative evaluation section, students were able to propose some possible uses that they would give to the platform.

Currently, the device only allows for use with agar based growth. Expanding its capabilities to liquid and microfluidic growth could greatly increase its utility. These different growth mediums could greatly extend the span and reach of experiments that can be tested on the bioreactor. Additionally, an improved graphical user interface can have great impact on the technical barriers that the device has for its use.

**Evaluation Limitations**

Future work will give the reactor a more robust quantitative evaluation. Using the small Petri dishes and diluted samples, we can carry out colony growth count so as to compare the reactor’s growth capabilities to those of commercial incubators. Stimulation modules can also be evaluated with fluorescence gene expression. This would be able to indicate the amount of gene expression impact that each experimental condition and stimulation module has. For example, we would be able to more clearly quantify the impact of temperature or stimuli exposure on up-regulating or down-regulating gene expression.
Social Limitations

The current version of the device was evaluated with graduate students with almost no previous synthetic biology experience at highly recognized academic institutions. While this is a good step towards evaluating the potential of the device, it still falls short of fully democratizing its potential.

Future evaluations and designs for the device will aim to climb down the educational ladder. This would make it possible for even kids at an elementary educational level to interact with the device. This can be achieved with improved module designs and with the help of an intuitive graphical user interface, as outlined above.

Furthermore, the device must also climb down the resource ladder, it is still hard to imagine such devices in communities with low resources. While the device is able to carry parts of the experimental process, there are still high costs associated to reagents, cells, agar, pipetting equipment, cell storage and incubators to prepare the cultures. Addressing these costs can be possible through the design of modules that use local materials, low energy production processes and cultures that are not necessarily reliant on complex lab infrastructure. In this context, the reactor could be used for workshops that help students to understand microbial cultures around them, this would eliminate the need for a big part of the heavy machinery needed for transforming cells while still stimulating creative and scientific thinking muscles.

5.5.2 Conclusion

Creating technological solutions that can be easily scaled and used by communities that have very contrasting needs, skills and resources is critical within the frame of community sensing. This means that we should shift our focus from only creating smart devices to enabling smart communities. With this shift, deploying technologies becomes less about finding centralized ways to impose technological solutions and more about giving communities skills and tools for them to develop their own unique solutions that respond to their specific needs. In the future, for example, we can envision communities that could use this infrastructure to actively monitor for undesired
metals in their water lines or to collectively monitor the spread of a disease. These
types of devices could also enable decentralized bioproduction of compounds that
could be used to combat certain health issues or improve distributed food production
processes.

Exponential technologies hold great promises for addressing some of the world’s
hardest challenges. Efforts to allow distributed deployment of these technologies will
be crucial for addressing the variability and scale of issues that will arise in the coming
decades. The present chapter outlines the design and validation of a low-cost, modular
and portable bioreactor. The bioreactor’s design proves to be a unique and valuable
addition to the current efforts of foundations, competitions and devices that aim to
democratize synthetic biology. This growing ecosystem of low-cost, open-source and
community oriented devices allows us to envision a future where communities are able
to respond to challenges with highly sophisticated technologies built in-situ.

While decentralized citizen science technologies will be essential for addressing the
future’s challenges, not all problems can be addressed through bottom-up processes.
The next chapter demonstrates how to integrate bottom-up sensing technologies and
frameworks to improve upon top-down city transformation and infrastructure.
Chapter 6

JettSen: Consensus Building

Infrastructure


6.1 Overture

Urban development requires a merge of bottom-up and top-down processes. As explored on the last chapters, some problems are best addressed distributively but some others require centralization, heavy infrastructure and large amounts of resources. For these types of problems, it is crucial to have mechanisms that can take into account communities’ points of view and concerns. It has been shown that the most successful top-down interventions are strongly aligned with the environment’s and communities’ needs. Sensor systems combined with machine intelligence can play an important role in high level community engagement and consensus processes.

Integrating sensing and data into consensus mechanisms, even within top-down
processes, can allow us to better inform communities and stakeholders about a broad range of issues that can be impacting them. Communities can then negotiate what issues are most relevant to them and escalate their concerns to the stakeholders who can solve it, such as government officials. The form of participatory democracy might greatly improve top-down policy and infrastructure decisions.

This chapter presents JettSen, a sensor fusion system, developed in collaboration with the Panasonic Corporation, that is mounted on shared electric bicycles. The system uses sensor fusion and unsupervised learning algorithms to create knowledge layers that describe different aspects of a bike ride. Clusters are placed in a knowledge abstraction framework that helps stakeholders to migrate their focus from providing or seeking solutions to asking the right questions and better defining the issues that are affecting communities.

The system is able to operate at the level of the rider, infrastructure, and fleet. This gives the system the ability to identify a broad array of issues that bikers can face. The chapter demonstrates the development of the open-source hardware, shows how K-means clustering can be used to characterize different types of trips, riders, and use contexts. Lastly, we discuss how the clusters can be used as inputs and visual aids for consensus building processes.

6.2 Background

6.2.1 Context

In recent years, we have seen a clear shift in mobility patterns worldwide. Cities are becoming ever more interested in deploying systems that serve as a better mobility alternative than privately owned cars. Shared mobility systems like Uber, Lyft, Bird, Lime, and multiple bicycle-sharing companies are attracting users at unprecedented rates [121]. These systems can sometimes complement widespread mass transit alternatives. Studies also show that utilization of bike-sharing services also increases mass transit use [47], which will push cities to improve services for these public transporta-
tion options. In addition, ongoing research points to lightweight, electric, shared, and autonomous as the key characteristics for the mobility of the future [27, 107, 192]. This means that bicycles and e-bikes, both shared and privately owned, will increase in relevance in the upcoming years.

The increasing demand for these types of systems is a great opportunity for cities to develop decentralized mobile sensor networks. These networks can tap into intelligent decision making that can positively impact the development of infrastructure, communities, and improve upon user experience in a way that was not possible until only a few years ago. City governments will be able to make commuting much more safe, healthier, and more sustainable while balancing privacy concerns [64].

Another important characteristic of bike-related infrastructure is that it is reversible and easier to iterate compared to heavy infrastructure [156]. This iterative qualities make bikes highly attractive for rapid testing of smart infrastructure within cities. Sensor systems that are unable to be quickly adapted for different needs are limited due to the inherent complex dynamics that abound within cities.

Despite mobile sensor networks being an attractive alternative for smart infrastructure, current deployments on bikes are based on systems designed with a specifically constrained problem in mind, and therefore lack flexibility for implementation on a wider range of problems than the ones they were designed to solve. This limits them greatly when it comes to using them for consensus building processes. Flexibility in sensor systems enables a more humane exploration of improvements for cities, migrating focus from providing solutions to asking the right questions to satisfy citizens’ needs.

6.2.2 Related Work

In the realm of cycling and bicycle-sharing systems, specifically, there are many ways in which mobile sensor devices have been used to improve a city’s and driver’s performance [127]. Accelerometer, ultrasound, and wind flow sensor data from bicycles has been used to improve user safety while driving [66, 170]. Other approaches have added extra sensors and interfaces using smart helmets [171]. Similarly, GPS location
coordinates for stations and bicycles have been used to identify user behavior patterns [89]. These systems have also been used to inform bike sharing companies about areas where there is a lack of stations, stations that need maintenance, or stations that are underutilized [16].

Although the idea of sensorizing bicycles has been extensively explored and implemented [196], there does not yet exist a system implementation that acts as a platform that is capable of enabling the fusion of multiple sensors and acting as a consensus building mechanism. A platform with such characteristics would allow for faster and cheaper exploration of concepts and implementations that could bring continuous change for cities and users.

6.2.3 Contribution

This paper proposes a system architecture for an open source mobile sensor fusion system, specifically implemented on a prototype electric assist bicycle \(^1\) manufactured by Panasonic Corporation. The system offers a flexible platform that can enable citizens, companies and governments to extract information that can be later used within an iterative, citizen-centric, decision making process.

The platform’s design incorporates an array of custom made and commercially available sensors that are capable of continuously saving data containing a varied set of descriptors for a bike ride. The diversity of continuously recorded variables is what enables the system to act as a flexible platform for the development of numerous projects with varied goals.

The mobile sensor platform is accompanied by a knowledge abstraction framework that helps to exploit the flexible nature of the platform. The knowledge abstraction framework is a crucial pillar to the proposal because it gives a solid structure to understand the ways in which the platform can have an impact in different scales and dimensions; community level, driver level or infrastructure level. When used together, the sensor system and the abstraction framework can empower citizens, researchers,

\(^1\)The bikes used in this study provides assist torque in addition to human pedalling. This bicycle does not drive solely from the motor unit.
The system is able to help in identifying distinct driving patterns that users can have on a bike with the use of unsupervised machine learning techniques. In addition, it demonstrates the value of having a large set of sensors in helping to address questions that could not be well addressed or explored by only one sensor. The goal of the implementation is to understand different interventions that can be made to improve the experience of the driver as well as the performance of the city through the use of sensor fusion, localization, and machine learning guided by the knowledge abstraction framework.

6.3 System Design

6.3.1 Hardware

System Architecture

The mobile sensor fusion system was built as a platform with a modular nature to allow for multiple use case scenarios. The architecture was designed in a way that each type of sensor value can be accessed individually to make the system adaptable to various research projects or questions. The system is capable of continuously recording a total of 27 different variables. Data acquisition is carried out on a central computer (Raspberry Pi 3 B+) that also acts as a local storage unit for the data.

A central unit is used to directly communicate with four external sensor modules (1) environmental sensor module, (2) camera module, (3) GPS module and (4) internal bicycle sensor. All modules use a wired serial communication to transmit data over to the central unit. The central unit accesses data of all four sensor modules, assigning timestamps, merging and saving it on to three independent files. Figure 6-1 illustrates the different modules of the system as well as their data input and output interactions.

To collect data, the central unit starts running all sensor programs from start-up. For activating data logging, a built in button is programmed for manual activation.
When the button is pressed, three simultaneous data collection processes are activated. The first process collects values from the environmental sensor and the motor unit. The second process stores GPS values and the third process saves raw camera data.

Each data process creates an individual file to store data, identifiable by a universally unique identifier (UUID). Each data instance is tagged with a timestamp to facilitate proper post processing of the data. The files are stored separately within the internal memory of the central unit and carry the same UUID as name to match the different data sets that belong to a single trip. This allows for future fusion of data sets containing distinct sensor values. Each process runs individually to maintain a collection frequency that is favorable for each type of sensor module.

Figure 6-1: Mobile Sensor Fusion System Architecture. Four sensor modules feed data into central unit. The central unit processes data, adds timestamps and generates three separate files labeled by a UUID. Data is later extracted and analyzed on an external computer.
Variable Description

Variables collected by the system are grouped into three main description categories (1) bicycle state variables, (2) environmental state variables and (3) geo-spatial state variables. The chosen classification gives a holistic description of a given bike ride, taking into account variables that may be of interest for distinct purposes and allowing for exploration of the relationship of a user, with the bike, the environment and the city itself.

Bicycle State Variables

Bicycle state variables are variables that represent the current state of the ride. This variables describe the mechanical and electrical state of the bicycle and are directly related to the use of the bike itself. Most notably, the motor drive provides the ‘input torque’ which is a direct measure of the interaction of the user with the bike. Table 6.3.1 gives a description of each variable that is categorized as a Bicycle state variable.

Environmental State Variables

Environmental state variables are variables collected by a research oriented environmental sensor [130, 166]. The variables describe the environment that surrounds the bicycle along the bike trip. This variables describe external factors that have to do, mostly, with the atmosphere surrounding the bicycle. Table 6.2 enlists variables classified as environmental state variables along with their units.

Geo-spatial State Variables

Geo-spatial state variables are the variables that describe the geo-location of the sensor system as well as the visual state (through a camera) of the bicycle. These variables are useful for locating and giving context to the bicycle state and environmental state variables as well as for mapping infrastructure that may be surrounding the bicycle at a given time within a trip. Table 6.3 enlists variables classified as geo-spatial state variables along with their units.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery Temperature</td>
<td>°C</td>
</tr>
<tr>
<td>Battery Voltage</td>
<td>mV</td>
</tr>
<tr>
<td>Output Current</td>
<td>mA</td>
</tr>
<tr>
<td>Remaining Battery Percentage</td>
<td>%</td>
</tr>
<tr>
<td>Remaining Distance</td>
<td>0.1 km</td>
</tr>
<tr>
<td>Remaining Time</td>
<td>0.1 min</td>
</tr>
<tr>
<td>Input Torque</td>
<td>kgf</td>
</tr>
<tr>
<td>Rotation State</td>
<td>Discrete</td>
</tr>
<tr>
<td>Crank RPM</td>
<td>rpm</td>
</tr>
<tr>
<td>Power Voltage</td>
<td>mV</td>
</tr>
<tr>
<td>Motor Temperature</td>
<td>°C</td>
</tr>
<tr>
<td>Motor Duty</td>
<td>Discrete</td>
</tr>
<tr>
<td>Motor RPM</td>
<td>rpm</td>
</tr>
<tr>
<td>Motor Speed</td>
<td>km/h</td>
</tr>
<tr>
<td>Encoder Count</td>
<td>inc-dec</td>
</tr>
<tr>
<td>Bicycle Speed</td>
<td>km/h</td>
</tr>
<tr>
<td>Drive Mode</td>
<td>Discrete</td>
</tr>
<tr>
<td>X Axis Acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>Y Axis Acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>Z Axis Acceleration</td>
<td>m/s²</td>
</tr>
</tbody>
</table>

Table 6.1: Bicycle State Variable Description and Units.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>lux</td>
</tr>
<tr>
<td>Temperature</td>
<td>°C</td>
</tr>
<tr>
<td>Humidity</td>
<td>Relative %</td>
</tr>
<tr>
<td>Pressure</td>
<td>% Pa</td>
</tr>
</tbody>
</table>

Table 6.2: Environmental State Variable Description and Units.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude</td>
<td>Decimal – Decimal</td>
</tr>
<tr>
<td>Longitude</td>
<td>Decimal – Decimal</td>
</tr>
<tr>
<td>Camera Output</td>
<td>RGB Values (0-255)</td>
</tr>
</tbody>
</table>

Table 6.3: Geo-spatial Variable Description and Units.

Component Description and Connections

Figure 6-2 shows the physical implementation of the complete system. Sensors are distributed at different locations of the bike depending on mechanical, operation and
electrical requirements. Figure 6-3 shows the electric connections, highlighting power lines and data transmission lines. Its important to note that the entire system is powered by the battery pack that is originally used to power the motor unit on the bike. The next subsections give detailed information about the five main components of the system’s architecture.

Central Unit and Power Supply

Central collection and processing is done on a Raspberry Pi Model 3B+. The raspberry Pi is connected to the rest of the modules through USB, UART and Software Serial Communication. The system’s power is drawn directly from the e-bike’s battery pack which operates at 26V. The battery is connected to a DROK Buck Converter, which is a commercial voltage regulator, that drops down the 26V from the motor battery to the 5V required by the Raspberry Pi and all sensor modules.

GPS Module

The GPS board is a commercially available GPS breakout board, Gowoops GPS Breakout Module, based on the GPS NEO-6M module. The board allows for direct communication through the use of software serial on the Raspberry Pi. The device is connected to power directly through the break out pins and uses a regular GPIO pin on the Raspberry Pi to transmit unparsed NMEA strings into the central unit. The software process collecting the GPS data verifies that GPS has proper fix to a satellite, parses the NMEA strings and extracts the latitude and longitude variables expressed in Decimal-Decimal units. The variables can later be converted to Decimal-Degrees to make mapping and visualization easier on commonly used visualization programs.

Camera Module

A commercial Raspberry Pi Camera Module V2 based on Sony’s IMX219 8-megapixel is connected directly to the Raspberry Pi’s camera connector. The module was used to avoid adding multiple soldered connections to the system. The camera is placed
on the bicycle’s front handle bar to capture the front facing view of the system ².

Environmental Sensor Module

The environmental sensor is a custom research board based on the ESP WiFi modules (MIT terMITe) that is capable of measuring 3 axis accelerations, ambient temperature, relative humidity, light intensity, atmospheric pressure and infrared proximity [130,166]. The board is connected using a micro-USB cable. Communication is carried out by using the hardware USB serial communications on the Raspberry Pi.

Internal Bicycle Sensor Module and Activation Button

The internal bicycle sensor module was custom developed by Panasonic Corporation and Panasonic Cycle Technology Corporation as part of an ongoing research collaboration with the authors of this publication. It is a unit that has control over all the bicycle state variables, excluding $x$, $y$ and $z$ accelerations, which are recorded by the environmental sensor module. The motor unit is connected directly to the Raspberry Pi and uses UART communication to send data to the central unit. It is relevant to note that there is a constant two way communication between the central unit and the internal bicycle sensor module due to the fact that the module is capable of receiving commands from the central unit that can later be passed directly to the motor to change and modulate electric assist.

This two way communication is used by the activation button mounted on the handlebar to signal the beginning and ending of data logging. The system uses the data indicator (led indicator) to signal to the user that data collection has begun or ended³.

---

²Camera module is not used on the use case implementation seen on section 4 but has been installed for future work on the platform.

³While the two way communication with the motor unit is only being used by the activation button, as the platform is implemented on different uses, the communication can become relevant to enhance the user’s interaction with the bicycle.
6.3.2 Software

Data collection, processing and storage

The system allows for simultaneous collection of data points from a large set of modules. To deal with the challenge of storing distinct data units and accessing modules operating at different frequencies, data collection was divided into three independent processes. Timestamps are added to each data point in each process to make fusion of the data sets possible.

The first process accesses and saves data from the internal bicycle sensor and from the environmental sensor board. This process creates two new data points every second $f = 2Hz$, making it the fastest process of the three. This sampling frequency was chosen because the variables collected by this process have variations that are relevant in very short time spans. We note that collection frequency for this process
Figure 6-3: Power Supply and Data Transmission Connections. Diagram shows electric connections and flow of information for the system's four sensor modules and the central unit. Note that all devices have a common ground provided by the bicycle’s battery pack, the diagram does not include ground connections to make it more simple to read. Red lines represent power (V+) connections and blue lines are all simplified data transmission lines.

can be accelerated in case the research question being addressed requires it to be. The second parallel process collects data from the GPS unit. Data is collected at the average update rate shown by most commercial GPS boards which is $f = 1Hz$.

The third process saves the data that comes from the camera module. The camera collects data at 24 fps and stores the raw RGB values of each captured frame.

Three separate CSV files are created every time the system is activated, each one of the files has the same name but is stored in a different directory. The names of the files are created using python’s standard UUID library [52] to create unique file ID’s that will never have conflicting name problems in storage. The file name also includes a timestamp from the moment of activation and the specific name of the bike that is recording the data. The file naming system was built to enable the scalability of the sensor system by using a format that allows to easily combine multiple sensor files built from multiple sensor sources coming from multiple bicycles.
Clustering Driving Patterns

It can be highly beneficial to understand different driving patterns for users to make bike rides and cities more comfortable, safe, and goal-adaptive. With this assumption in mind, a use case for the platform was developed with the aim of clustering distinct driving patterns shown on a bike trip. The goal of the implementation is to demonstrate the value of (1) sensor fusion within the context of a mobile sensor system for urban planning and (2) to show how the system, data and algorithm can yield results that are relevant at different level of abstraction and at different trip layers, as illustrated in figure 6-5.

Most implementations for identifying driving patterns and states rely on the use of supervised machine learning techniques and commonly use accelerometer data as input for their models. In contrast, our implementation leverages on a diverse set of sensors and on unsupervised clustering algorithms that allow for a reduction of bias in the created models.

Clustering algorithms, when used for anomaly and pattern recognition, leverage the fact that some patterns will not be regular and in many ways, unknown and unpredictable [126]. For our implementation, we assume that the complexity surrounding cities is best addressed by algorithms that are not restricted to predefined classes like the ones commonly used in supervised machine learning techniques. We therefore chose to base our implementation, specifically, on the well know K-means clustering algorithm.

Clustering Algorithm Details

The K-means algorithm is one of the most accepted algorithms for creating unsupervised clusters on a multidimensional space [105]. For our use case we decided to use an unsupervised clustering algorithm because it allows us to reduce classification bias by keeping class label identification out of human error.

For our K-means implementation, we took into account variables that can be found on the csv file that mixes bicycle state data and environmental data. This
allows us to create a data set that combines nine different variables. We first access individual sensor arrays and create a matrix that only contains the data arrays for the variables. Used variables are specified in Table 6.4 along with their units.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Axis Acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>Y Axis Acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>Z Axis Acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>Input Torque</td>
<td>kgf</td>
</tr>
<tr>
<td>Bicycle Speed</td>
<td>km/h</td>
</tr>
<tr>
<td>Light</td>
<td>lux</td>
</tr>
<tr>
<td>Temperature</td>
<td>°C</td>
</tr>
<tr>
<td>Humidity</td>
<td>Relative %</td>
</tr>
<tr>
<td>Pressure</td>
<td>% Pa</td>
</tr>
</tbody>
</table>

Table 6.4: Selected Variables for Clustering Analysis

After our variables of interest are isolated, we normalize their values so that all of them only have values that fall between 0 and 1. Normalization helps to balance the square distances in distinct dimensions, which is important for minimizing the cost function of the K-means algorithm [136].

Once the data has been arranged and normalized, the algorithm is implemented using the scikit-learn python library [101]. The library implements the algorithm with normal random initialization of centroids and looks to minimize squared distance between points.

It is well known that K-means can continuously reduce the sum of squared distances between points as more clusters are added. Nevertheless, if too many clusters are added, the probability of overfitting is increased. To avoid this, the number of optimal clusters was chosen through the use of an elbow curve analysis. As can be seen in figure 6-4, iterating from one to 50 clusters gives an optimal number of clusters, k, of seven for the three matrix variations that were used.
Figure 6-4: Normalized elbow method analysis for determination of optimal clustering parameters for each one of the matrix variations (x acceleration, fusion and multiple trip). Results indicate that seven is the optimal number of clusters for each one of the matrix variations. The point on each curve was chosen by analyzing the moment at which the error rate change decreases for the first time.

6.4 Validation

6.4.1 Deployment

The use case scenario was implemented in the following order. First the mobile sensor fusion platform was installed on an electric bicycle prototype, Jetter e-bicycle, provided by a large consumer electronics company. After the system installation, we collected data from 16 different bike trips. The data was stored locally on the bicycle and later extracted. Once extracted, experimentation of different algorithmic parameters was done for the clustering algorithm.

To show the different results along the abstraction gradient, we ran our clustering algorithm on three different matrix variations built from the same data source. The variations allow us obtain different information from running the same algorithm on matrices that use different fusion variations.

The first matrix only contained data describing the x axis’ acceleration of a single trip. The second matrix was built using data from the same single trip as the first matrix, but arrays describing speed, torque, ambient temperature, light intensity, humidity and atmospheric pressure were added, yielding a matrix with 9 dimensions. Lastly, the third matrix, contained the same variables as the second matrix but was
sequentially concatenated with the trip data of 15 other trips, in this way creating a matrix that contained 9 different dimensions and 16 different bike trips.

Knowledge Abstraction and Consensus Framework

This section presents a framework for understanding the value of a mobile platform that enables sensor fusion for the exploration of different research questions. Sensor fusion represents the combination of sensors to obtain a data source that is more complete than individual sensor data collection. A variety of methods have been proposed from existing sensor fusion studies and how to classify those systems [41]. For this publication, we would like to consider what kind of system should be targeted in the effort of data acquisition in urban planning specifically using e-bikes. In the case of a mobile fusion sensor system for bicycles, sensor fusion allows for more holistic understanding of a specific bike trip and, therefore, enables the creation of models that can take into account a larger amount of variables. A holistic data collection process is more likely to yield models that can be more accurate and that can approach reality in a less reductionist manner. Figure 6-1 shows the three data file outputs of the system. Processing and fusion of the data files is not done during the collection process to keep collection frequencies stable. Rather, it is done during the analysis and post processing of the data files.

The framework outlines the way in which it can be possible to take raw sensor data and turn it into valuable fusion knowledge, specifically in the context of bicycle trips and cities. To describe the knowledge that can be obtained from the system we use two different dimensions.

The first dimension describes the level of abstraction. The distinction between abstract and rigid knowledge is crucial in the context of cities because it can help to define projects or goals that can have impact in the short-term from those that can have impact in the long run as well as those projects that address technical questions from the ones that address societal ones. An example of rigid knowledge generation would be to map the exact spots of potholes in a given area in order for governments to have a detailed map of city infrastructure that needs maintenance. In
the other hand, knowledge with higher abstractions are needed for building consensus such as master plans composed with community values. The fused sensor data may illustrate the behaviour of social demographics which lead to value propositions that the community wants to push. This may involve discussions around data ownership and privacy. High levels of abstraction are an important factor when considering the use of urban planning adjunct with citizen participation.

The second dimension deals with the layer of the trip at which the knowledge is relevant for. The three proposed layers are (1) the drive (2) the community and (3) the infrastructure. As examples, a project that uses the system’s sensors to predict the drivers likely next action would be on the drive layer. In contrast a project that uses camera and GPS to map the types of lanes would be classified within the infrastructure layer. Figure 6-5 shows the relationship that different layers of abstraction can have with the different layers of the city along with example uses placed on each one of the different dimensions.

If we look at the existing methods of data utilization in urban planning, it is a Waterfall System [114] which has a drawback in the fact that a feedback system is not designed into the system itself. In the next section, we will consider how to create a model that progressively asks questions of urban planning issues.

6.4.2 Data

Single Trip $x$ Acceleration Clustering

As previously explained, the first matrix variation only contains sequential data of the $x$ axis acceleration for a single trip. The input matrix has a shape of $[n, 1]$ where $n$ represents the number of recorded samples. Figure 6-6 is the plot of the $x$ axis acceleration data against time, the color code on the image represents the different clusters that were generated by the algorithm.

Empirically, we can conclude that the algorithm manages to identify and cluster different acceleration intensities across time. We can see that high accelerations are less common and that the strongest group is centered around the average acceleration.
Figure 6-5: Sensor variables Knowledge Impact. The diagram allows us to visualize the way in which different trip layers are related to the different levels of abstraction. The different layers of abstraction allow for exploration of highly specific and technical issues and their relationship with higher level values that the community might hold. Focusing on a specific area within the diagram can yield entirely different knowledge, even if the same data source and algorithm is used, as can be seen in section 4.

of the trip. Through the generation of definable clusters that detect intensities of acceleration, we demonstrate that using individual sensor data for a single trip can help to generate models that are able to respond to rigid issues. Examples of use for such model would be to address issues of locating potholes, intersections or accidents in a city by combining the high value acceleration clusters with the GPS coordinates were they were saved.

**Single Trip Fusion Data Clustering**

For the next analysis, we turn to the second matrix variation. This matrix contains information for the same trip as the past uni-dimensional analysis alongside a wider range of variables, giving it a shape of \([n, 9]\) where \(n\) is the number of recorded samples. The variables that are added to this matrix are accelerations for the \(y\) and \(z\) axis, torque, bike speed, temperature, humidity, light intensity and atmospheric
Figure 6-6: Single Trip X-Acceleration Clustering. The graph plots normalized acceleration data on the $y$ axis and relative time (each unit represents 0.5 ms) on the $x$ axis. The K-means algorithm is capable of effectively clustering different acceleration points.

From figure 6-7, we can identify that the clustering algorithm behaves in a more sequential way, meaning that clusters appear to be spread out horizontally over time as opposed to figure 6-6 were clusters are spread out vertically. This sequentially demonstrates the possibility of extracting information relevant to different states in a single bike trip. Clusters tend to describe different moments in the bike ride meaning that questions revolving around driving patterns and driving states can be addressed.

**Multiple Trip Fusion Data Clustering**

The last matrix variation includes the same sensor data as the second variation. In addition to this data, 15 other trips were sequentially concatenated meaning that the matrix includes information about 16 different trips and 9 different variables for each one of the trips. The matrix has a shape of $[m, 9]$ where $m$ is the sum of the vector $[n_1, n_2, n_3, ..., n_{16}]$ and $n_i$ is the number of recorded samples for each one of the trips.
Figure 6-7: Single Trip Fusion Data Clustering. The graph plots normalized data for 3 axis acceleration, speed, torque, temperature, light, humidity and pressure on the $y$ axis and relative time (each unit represents 0.5 ms) on the $x$ axis. The K-means algorithm is capable of effectively clustering different instances in a single trip in a sequential manner.

Figure 6-8, shows the results of running the clustering algorithm on the third variation matrix. We can observe that it maintains the sequential characteristics that the results on figure 6-7 show. Nevertheless, the results show a less granular or more abstract clustering pattern. Instead of clustering different states of a single trip, clusters seem to be dominated by entire trips.

Comparing the results from the second matrix and third matrix variations, we can show the value of sensor fusion for enabling an abstraction of different types of knowledge from the same data source and algorithm. Results from the second matrix variation speak about characteristics of different times within a bike trip while results from the third matrix speak about characteristics and differences between trips as a whole.
Figure 6-8: Multiple Trip Fusion Data Clustering. The graph plots normalized data for 3 axis acceleration, speed, torque, temperature, light, humidity and pressure for 16 different trips on the $y$ axis and relative time (fitted to start from zero) on the $x$ axis. The K-means algorithm is capable of effectively clustering different trips in a sequential manner.

6.5 Discussion

6.5.1 Limitations & Future Work

Future work on the platform will be centered around making the system scalable to a larger amount of bicycles, this will require improvements on the mechanical enclosures of the system as well as the electrical connection layouts. Scaling the system to more bikes will allow us to incorporate multi user classification (classifying multiple trips for multiple users) and to further explore how different fusions and algorithms could help address key questions for enabling a more humane development of our cities.

Integrating the system’s output and clusters into tangible consensus building platforms such as the CityScope [7]. This integration would help to further see the value of it as a consensus building technology as the information can be integrated into more complex models of urbanization.
6.5.2 Conclusion

The combination of a mobile sensor fusion platform with a knowledge abstraction framework that embraces citizen and community input, can become a powerful tool for the development of better cities. Our platform and framework give the flexibility that is needed to make cities more capable of managing their inherent complexity. Its situation within a knowledge abstraction framework makes it easier to integrate sensing and machine intelligence into today's consensus building systems.

At its simplest form, a project detecting potholes may only consider accelerometer readings. Yet for a city to maintain infrastructure, prioritizing which road to improve from a limited budget is non trivial. By providing a platform with multiple sensors we provide a way to incrementally accumulate knowledge for each step presented in the use case.

Extracting insight from data has long been studied in the field in data mining, more formally under the terms of Knowledge Discovery in Databases [46]. For our use case, we chose clustering which is an approach that is widely adopted to detect anomalies [40,54,125]. We have built up different clusters by using different portions of the same data source illustrating different aspects of data.

The first step may show anomalies which could be tied to dangerous road conditions like potholes. This first type of knowledge can be used to define danger. By associating GPS data with this data, we will be able to know where these sensor readings occur. Yet, it is early to decide which occasions has more impact. The second step gives context to the incidents and how pothole-like data influences the drive relative to the whole trip. We can now start to ask questions: Do we want to allocate maintenance efforts where people drive at high speed or near car traffic? The third stage gives indications of which trip has relative risk, leading discussions on community preferences. The focus of the inquiries could shift to the time of the trip or the differences in the environment: Does the community weight more to improve bike riding experience in leisure time or when citizens ride their bikes aggressively? At this point, the questions lean on community values to choose between improving
recreation time or making it safe for rushed riding behavior.

In addition, for applications like urban planning where interventions are interrelated to other socio-environmental issues [72], a platform that incrementally drives more questions is of great value. By questions we point to value judgements or moral perspectives that are addressed in studies such as [12]. This research examines a platform that not only acquires data, but also leads to stacking up knowledge that leads to these conversations.

The platform itself does not cover all the aspects that a city needs to take in account. For example, the city may need to use income levels to balance equity. By designing a platform that bridges quantitative to qualitative data, we aim for data oriented urban planning to avoid a reductionist approach and impose a single technical solution, but to provide options to explore possibilities connecting other aspects that constrains human behavior such as law enforcement and normative values [102]. Based on the discussion of results from the use case presented above, we point out the value of using the platform along with machine intelligence as tools for creating citizen consensus.
Chapter 7

Conclusions

This thesis describes a series of case studies of community-scale sensor technologies that can be used to collect granular information about socio-environmental and behavioral aspects of urbanism. This work is based on the idea that socio-environmental community-based sensing infrastructure could help us to create consensus and understanding about ways in which behavior, urbanization, and environment are closely related and co-influenced.

Complementing the strengths of current "smart city" infrastructure, this thesis expands the definition of socio-environmental community sensing so that it encompasses bottom-up data pipelines that organically stem from community challenges, needs, infrastructure, and desires. This view of sensing could unlock possibilities that are not currently possible because it may allow us to tap into community values and collective power as drivers for urban transformation. By embedding data collection into pre-existing and ongoing activities, these technologies intrinsically target conditions relevant to the community.

Whispers of The Mountain showed a sensing system that is embedded on snow skis. With this project, we learn how we can instrument equipment and activities right at the interface where human activity meets the environment. Due to their modularity and flexibility, this type of system can help us to extract holistic profiles that are valuable to multiple stakeholders. Systems that are opportunistic, distributed, low-cost, and activity-driven help us to gain unique information that more adequately
describes the complex relationships within communities and environments.

Building on the idea of low-cost, distributed, multipurpose and opportunistic networks, the thesis also shows us how to take their impact a step further through a network topology and use case to monitor coastal ocean waters. The work envisions that as more people in the community embrace the sensor nodes, the community could potentially grow its information base, helping it to decide which behavioral changes would be the most beneficial to the collective and the environment.

The next phase of investigation explored software architectures that can allow such systems to adapt to different use cases and scenarios. Through implementing $CO_2$ sensors in classrooms and offices, this study shows how algorithms can be designed so as to become predictive, adaptive, and easy to scale. Such algorithms can help us address the differences in user patterns, motivations, use cases, and spatial layouts that make deployments challenging, by enabling a system to adapt automatically to different environmental and social contexts. This reduces the cost and complexity of tailoring sensing infrastructures to each deployment site.

Community sensing places technologies at the heart of the social fabric of human collectives. As technology is increasingly embedded into everyday activities, creating tools and infrastructure for technological literacy becomes vital. Literacy can allow communities to control and develop their own technological interventions making them more effective and democratizing the benefits that they can have. MAIA, as a case study, is a device that contributes to the growing ecosystem of foundations, companies and apparatuses that are seeking to democratize benefits and understanding of SynBio. Using the device within student workshops gives valuable insights into the possibilities that community-based creativity and scientific skill enhancing devices could bring to society.

Not all problems can be addressed through grass-roots bottom-up approaches, however. Sometimes top-down interventions are necessary, and these work best when they incorporate community values in the decision making processes. JettSen’s technology and knowledge abstraction framework helps us to see how community sensing systems can be used to reach consensus on the problems that a community might
prioritize.

Taken together, the studies described in this thesis lay out instances of bottom-up sensing infrastructure that could be used within larger development efforts in the future. This vision of Socio-Environmental Community Sensing features data systems that are under control of citizens, and represent promising complements to current ways in which we develop our data infrastructure. New systems should aim to enable communities to access granular information that can precisely characterize how they engage with complex socio-environmental systems. This, in turn, could allow communities to use this information within higher-level transformation, prediction, and consensus processes. These bottom-up sensing capabilities could ultimately lead to collective behavior, policies, and infrastructure that is more sustainable and effective at addressing the great challenges that future cities will face in the future ahead of us.
Bibliography


[38] e Gizmo. e-gizmo e-201, 2019.


[98] Amino Labs. Discover your passion in stem biotechnology with kits, books, amp; more.


[122] minipcr minipcr. Making science accessible to everyone, everywhere.


Opentrons. Open-source lab automation.


167

