LAHA: A FRAMEWORK FOR ADAPTIVE OPTIMIZATION OF DISTRIBUTED SENSOR FRAMEWORKS

A DISSERTATION PROPOSAL SUBMITTED TO MY COMMITTEE
IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

By

Anthony J. Christe

Dissertation Committee:

Philip Johnson, Chairperson
Lipyeow Lim
Dan Suthers
Peter Sadowski
Milton Garces

Keywords: distributed, sensors, management, adaptive, optimizing, predictive

October 24, 2018
Version 1.0

Copyright © 2019 by
Anthony J. Christe

ABSTRACT

Distributed Sensor Networks (DSNs) are faced with a myriad of technical challenges. This dissertation examines two important DSN challenges.

One problem that is apparent in any DSN is converting "primitive" sensor data into actionable products and insights. For example, a DSN for power quality (PQ) might gather primitive data in the form of raw voltage waveforms and produce actionable insights in the form of classified power quality events such as voltage sags or frequency swells or provide the ability to predict when PQ events are going to occur by observing cyclical data. For another example, a DSN for infrasound might gather primitive data in the form of microphone counts and produce actionable insight in the form of determining what, when, and where the signal came from. To make progress towards this problem, DSNs typically implement one or more of the following strategies: detecting signals in the primitive data (deciding if something is there), classification of signals from primitive data (deciding what is there), localization of signals (when and where did the signals come from), and by forming relationships between primitive data by finding correlations between spatial attributes, temporal attributes, and by associating metadata with primitive data to provide contextual information not collected by the DSN. These strategies can be employed recursively. As an example, the result of aggregating typed primitive data provides a new higher level of types data which contains more context than the data from which is was derived from. This new typed data can itself be aggregated into new, higher level types and also participate in relationships.

A second important challenge is managing data volume. Most DSNs produce large amounts of (increasingly multimodal) primitive data, of which only a tiny fraction (the signals) is actually interesting and useful. The DSN can either utilize one of two strategies: keep all of the information and primitive data forever, or employ some sort of strategy for systematically discarding (hopefully uninteresting and not useful) data. As sensor networks scale in size, the first strategy becomes unfeasible. Therefore, DSNs must find and implement a strategy for managing large amounts of sensor data. The difficult part is finding an effective and efficient strategy deciding what data is interesting and must be kept and what data to discard.

This dissertation investigates the design, implementation, and evaluation of the Laha framework, which is intended to address both of these problems. First, the Laha framework provides a multi-leveled representation for structuring and processing DSN data. The structure and processing at each level is designed with the explicit goal of turning low-level data into actionable insights. Second, each level in the framework implements a "time-to-live" (TTL) strategy for data within the level. This strategy states that data must either "progress" upwards through the levels towards more abstract, useful representations within a fixed time window, or be discarded and lost forever. The TTL strategy is interesting because when implemented, it allows DSN designers to calculate upper bounds on data storage at each level of the framework and supports graceful degradation of

DSN performance.

There are several smaller, but still important problems that exist within the context of these two larger problems. These larger problems are addressed by solving a series of smaller problems. Examples of the smaller problems that Laha hopes to overcome in transit to the larger goals include optimization of triggering, detection, and classification, building a model of sensing field topology, optimizing sensor energy use, optimizing bandwidth, and providing predictive analytics for DSNs.

The claim of this dissertation is that the Laha Framework provides a generally useful representation for DSNs. I will evaluate this claim in the following ways.

First, to evaluate the generality of the network, I will design, implement, and deploy two Laha-compliant reference networks in two different domains, power quality and infrasound. These reference implementations will generate evidence for the ways in which Laha supports the goals of the sensor networks and ways in which it might fall short. The implementations may also provide insights into the types of distributed sensor networks for which Laha is well-suited, and the types for which it is not.

Second, these implementations will enable me to evaluate the multi-level representation system. I claim that Laha will enable a distributed sensor network to derive actionable insights from low level data, and that each of the levels will be important to that process. The two reference implementations will provide concrete data as to the set of levels that are useful in practice, or whether different levels would be more appropriate, or if the level strategy itself has problematic features.

Third, my evaluation will assess the TTL-based approach to managing data volume. I claim that a benefit of Laha's mechanism for managing data is that it will enable the calculation of upper bounds on data storage requirements. In my evaluation, I will develop the analytical procedures required for calculating data storage requirements, and see if these procedures are valid in practice. One obvious problem with a TTL approach is the possibility of false negatives: data that is discarded before it has been recognized as important. My evaluation will include studies designed to assess the frequency of false negatives and how important the problem might be in practice.

Finally, my evaluation will assess the ability to solve the tertiary problems of optimizing triggering, detection, classification, bandwidth, sensor energy usage, predictive analytics, and the ability to build a model of the sensing field. I claim that these problems need to be addressed in some form in order to solve the larger problems of turning primitive data into actionable insights and to provide a mechanism for managing large amounts of sensor data. I will compare and contrast state of the art algorithms present in the literature to determine if they are effective in practice and useful for addressing the two larger problems. My evaluation will provide metrics on false negatives and false positives as a means of demonstrating effectiveness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Distributed sensor networks (DSNs) consist of any number of sensors that collect and sense information about the physical environment around them. The sensors that make up these networks can either be homogeneous or heterogeneous. Distributed sensor networks are dynamic in that sensors can be added or removed from the network at any time. DSNs also increasingly include mobile sensors as well. With the onset of the Internet of Things (IoT), its easier than ever to build and deploy distributed sensor networks. Further, mobile devices, such as mobile phones, are seeing increased usage as intelligent sensing agents.

Distributed sensor network (DSN) optimization is a broad topic with many different facets to consider. Much of the literature on the topic focuses on optimizing data flow between sensors as data flows from sensor to sensor and eventually to a sink.

The focus of this proposal however, is to deal with the challenges of a specific subset of DSNs. That subset is DSNs where data always flows directly from each sensor in the network to sink nodes, thus, eliminating the need to worry about intra-sensor communication, networking, and routing.

There are a broad range of technical challenges beyond the data sink. The introduction of the Internet of Things (IoT) has created an explosion of internet connected devices that sense a massive number of attributes about the physical world surrounding them. An increase in sensors has created an increase in multimodal data generation with the inverse problem of creating a decrease in the signal-to-noise ratio, making it more difficult to identify and classify signals of interest. Multimodal data provides challenges for analysis algorithms because each sensor may be streaming multiple physical features that need to be analyzed and dealt with independently and dependently. As the densities of sensors increase, analysis must be able to work with missing data, incorrect data, incomplete data, and data coming from a heterogeneous mix of hardware and sensor configurations. Further, we can no longer assume that sensors are static in time and location as mobile sensors are quickly becoming more prevalent, making analysis trickier. All of these issues require an increase in storage and computational resources. Therefore, we must find approaches to deal with sensor data to lessen these hurdles.

As DSNs scale, available storage must be balanced with the amount of data being retained. Further, once data is collected, we need strategies for turning sensor data into actionable data and insights. This generally involves detecting and classifying signals of interest. It's these last two important DSN challenges that are the focus of this dissertation.

1.1 Converting Sensor Data into Actionable Insights

Data collected from sensors is often a sampled payload of data points representing some feature in the physical world. As examples, weather stations produce sampled features relating to temperature, wind speed, and humidity, power meters produce a metric of total electricity consumed, power quality sensors produce sampled data points which include voltage, frequency, and THD, and infrasound networks produce sampled data which represent audio waveforms.

These features by themselves, while interesting, do not provide any context as to if there is a signal, what the signal is, when and where the signal came from, or what caused the signal in the first place. Detection and classification algorithms are used to attempt to extract some of these properties. Primitive data is aggregated and compared to other primitive data to find correlations in both time and space. Data is compared to historic data in an attempt to find patterns or other similarities. This type of data is more interesting in that we might learn more about a signal using these techniques, but they still don't provide actionable insights or causality information. Further, the problem of providing actionable insights is highly dependent on the sensing domain. Depending on other available sources of data, providing data fusion and context from outside of the DSN can be difficult.

1.2 Big Data Management in DSNs

Big Data is generally defined by the four V's; volume, velocity, variety, and value. These characteristics can be observed in many of the DSNs that exist and are being created today.

That is, distributed sensor networks create a large volume of data due to the abundance of IoT and mobile devices that make up DSNs. As communication infrastructures improve and hardware becomes smaller, smarter and more energy efficient, sensors are able to send and transfer larger amounts of data. The ease of building and deploying sensors in DSNs means that more sensors can be produced much more cheaply allowing for more sensors to be used within a DSN, increasing coverage, but also increasing the volume of data.

Distributed sensor networks create a variety of data with different formats and data quality issues. Distributed sensor networks can produce data at high velocity. These characteristics of data produced from distributed sensor networks create a need for efficient architectures and specific algorithms designed for working with Big Data.

Further, sensor networks are often constrained in both computing power and available energy sources. This forces us to find comprises between data collection, onboard sensor processing, sensor communication, and network coordination.

As DSNs scale, the amount of data a DSN must store and process increases. At certain scales, DSNs simply can not store and process all of the primitive data that sensors are producing or process and store aggregate data products that detection and analysis routines produce. Designers of a

DSN can either choose to collect and keep all data forever (from raw data to generated products), or they can implement strategies for systematically discarding (hopefully) non-interesting data. If the first option is chosen, then there is no risk of accidentally discarding signals of interest and data can be reanalyzed when analysis algorithms change or are tweaked. However, storage and analysis of such amounts of data can cause system degradation or even be unfeasible. If the second option is chosen, processes must be put in place that attempt to only store "interesting" data and discard sensor noise. The second option runs the risk of discarding important data and old data can not be reanalyzed under this approach. However, this approach provides the benefits of providing predictable data storage requirements that can be tuned and optimized for a particular domain and DSN.

1.3 Traditional Approaches to DSN Optimization

Much of the literature focuses on the reduction of bandwidth and communication between sensors nodes and between sensor nodes and the sink. This is mainly performed as a means of sensor energy requirements allowing sensing to stay online longer or focus their energy usage for sensing or edge level computing. Anastasi et al[4] provide a really nice literature review on techniques for energy conservation in wireless sensor networks. Many of these approaches utilize optimized triggering[3] or exploitation of topology knowledge[39] to minimize sensor communications and save sensor energy. General approaches to Big Data management include compression[33] or storage systems where the goal is to have a distributed file system and move data close to where it is being processed, such as the Hadoop Distributed File System[39]. Other systems such as NiFi[14] provide a nice interface for ingestion and movement of data between Big Data tools while also providing data provenance, but do not go far enough in focusing on data reduction and graceful degradation. Carney et al.[5] discuss how monitoring applications require management and clean up of stale sensor data. Much of the literature on topology management is written to decrease sensor energy requirements by exploiting the density of sensors within a sensing field topology. For example, the ASCENT[6] framework provides adaptive self configuring sensors that exploit topology denseness to decrease sensor energy usage. Several other frameworks have been designed with the same goal of reducing energy usage by exploiting topology[31],[30].

1.4 Laha: An Abstract Framework for Adaptively Optimizing DSNs

I propose an abstract distributed sensor network framework, Laha, that adaptively optimizes data storage using a tiered TTL approach and makes strides towards providing actionable data by providing a mechanism in which typed aggregated data is continually refined to the point of being

of becoming actionable.

The Laha data model can be conceptualized as a multilevel pyramid (see fig. 1.1). Laha Actors act on the data model to move data upward through the levels and to apply optimizations downward through the levels. Many of these optimization techniques were developed independently. Laha provides a conceptual framework that enables them to work together.

The lowest level stores all recently received raw sensor data. This data expires and is automatically removed within a limited period of time (for example, 1 hour) unless the data is found to be interesting, and is thus propagated upwards to the next level of the hierarchy. Higher levels of the data hierarchy organize data in the same way, however each level adds context to the examined signal or signals. Context includes classifications, locality metrics, temporal metrics, or similarities to current or prior signals of interest. The highest level of the hierarchy, Phenomena, represents predictive capabilities of the sensor network which are then used to optimize and tune the lower levels. The Phenomena also form the basis for providing actionable insights.

A high level summary of the Laha abstract framework is provided as figure 1.1. The figure shows the levels and names of the hierarchy, a brief description of the functions of each level, and Laha's Actors and how they move data upwards (right hand side) and how they apply optimizations downwards (left hand side).

The Laha framework aims to provide two important benefits to DSNs:

1. Convert sensor data into actionable data and insights
2. Provide graceful degradation and metrics on storage requirements for voluminous sensor data

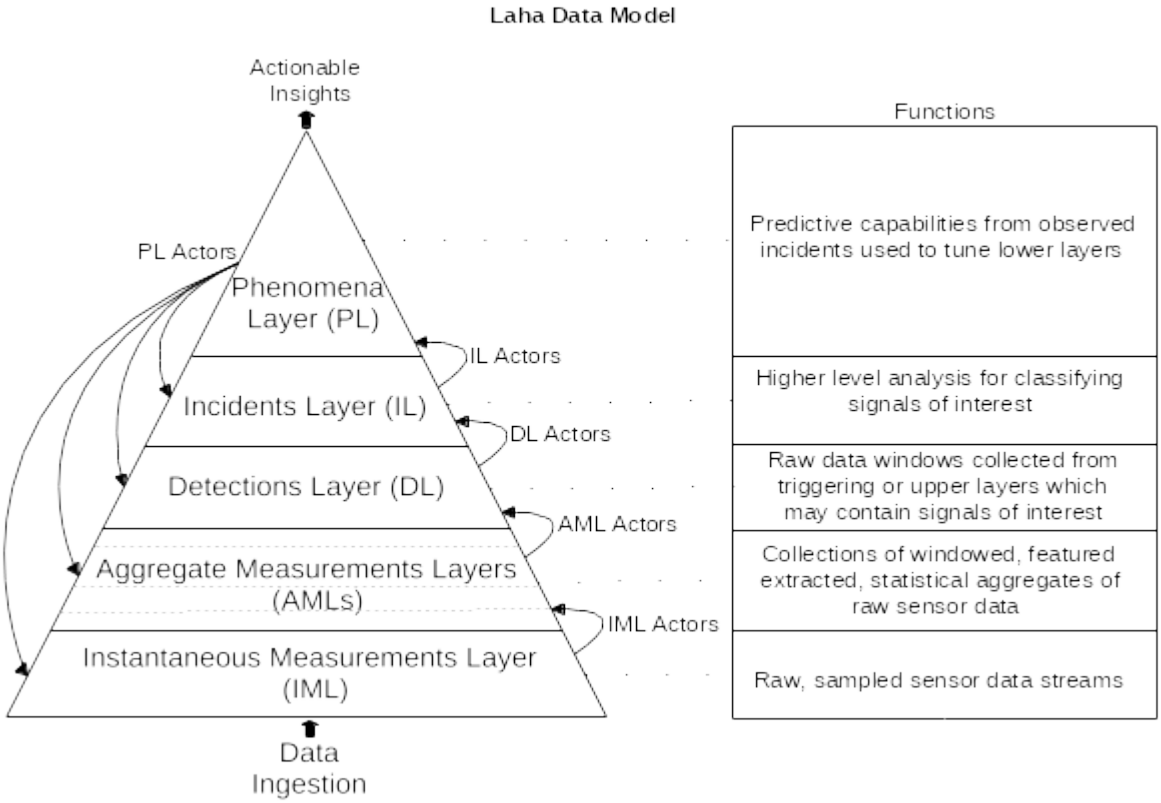
Although not the main focus of this dissertation, Laha hopes to show tangential benefits in the following DSN problem domains:

1. Triggering optimizations
2. Detection and classification optimizations
3. Topological optimizations
4. Sensor energy requirement optimizations

These tangential benefits are provided by Laha Actors that exist within each level of the Laha framework. I don't claim that these techniques are novel, but I do claim that either all or a subset of these techniques are required to enable progress towards the main goals of this framework. To that end, Laha Actors will implement several state of the art algorithms present in the literature that will address these tertiary problems.

Laha will be evaluated by designing and implementing two Laha-compliant reference implementations, OPQMauka and Lokahi. Open Power Quality (OPQ) is a power quality (PQ) network

Figure 1.1: Laha Conceptual Model Summary



consisting of custom hardware and distributed software services that detect distributed PQ signals such as voltage sags and swells, frequency sags and swells, transients, THD, and other known PQ issues. OPQMauka is a distributed, plugin based middleware component of OPQ that performs higher level analysis, data management, and optimizations of the OPQ services. Lokahi is a distributed infrasound network consisting of mobile iOS and Android devices and multiple cloud based software services whose purpose is to supplement the International Monitoring System (IMS) in detecting large infrasound signals.

The reference implementations will be designed and deployed to test sites at UH Manoa and at the Infrasound Laboratory in Kailua-Kona, Big Island during Q4 2018.

Data collected from the PQ network will be validated against calibrated reference sensors that have already been installed at the power mains of a subset of buildings on campus during Q1 of 2019. The Office of Energy Management at UH Manoa has given us full access to live and historic PQ data collected at these reference sensors. OPQBoxes will be co-located and placed in buildings with the reference sensors so that I can validate that the triggering and raw data streams I receive from the OPQBoxes are in line with what the reference sensors are observing.

Data collected from the infrasound network will also be validated against industry standard calibrated BNK infrasound sensors. Further, signals in the infrasound network are known a priori since I am able to control the signals that are generated from our calibrated infrasound source, allowing further validation of received signals.

In order to evaluate the generality of the Laha-framework, two separate Laha-compliant DSNs sensing different domains will be designed, distributed, and evaluated.

The first Laha-compliant DSN is Open Power Quality (OPQ) a distributed DSN that collects and analyze power quality (PQ) signals. PQ is a measure of the "goodness" of the power feeding your electronics. The features that this network collects includes voltage, frequency, and THD. From these features, OPQ can classify the following PQ signals: voltage dips/swells, frequency dips/swells, high levels of THD, and transients. Another goal of this network is to detect distributed PQ signals. That is, the same signal detected on multiple sensors to study how PQ signals move through a power grid. This network will provide metrics on the number of incidents classified as well as numbers of correct predictions from Phenomena. The number of classified incidents will be compared to industry standard PQ monitors co-located with OPQ sensors as a means of evaluating if Laha is capable of supporting the goals of this network.

The second Laha-compliant DSN is Lokahi, a distributed, mobile infrasound detection network. Infrasound consists of sounds waves that are less than 20 Hz. These signals are generated by large movements of the atmosphere and can be observed from large standoff distances. Examples of infrasound sources include volcano eruptions, meteors, missile launches, and large explosions. In this network, Android and iOS devices are deployed with a special app that is capable of collecting acoustic signals as they travel through the atmosphere. As part of the evaluation, in this network,

we hope to be able to collect and discriminate infrasound signals from different types of infrasound sources. Many of these signals will be correlated with industry standard infrasound sensors to show that Laha is capable of supporting the infrasound detection goals of this network.

In order to evaluate the multi-level representation of the Laha Framework in the context of providing actionable data, I will setup experiments to produce cyclical and predictable signals and test whether or not Laha is able to utilize predictive analytics to provide actionable insights. To test this, I will provide the number of false positive and false negatives for predictive analytic results. I will evaluate if the sensing domain has any effect on how well Laha is able to provide actionable insights. I also claim that each level in the Laha-hierarchy is important in the process of deriving these insights. I will provide data that either supports or opposes the usefulness of each level, whether the current number of levels is adequate, and whether the idea of using level to provide actionable insight is useful at all.

I will evaluate my claim that a tiered TTL approach to sensor data management provides the benefits of providing an configurable upper bounds on storage requirements for each Laha level, graceful degradation, and a reduction of sensor noise being stored. To test this, I will implement procedures for calculating storage bounds and see if these theoretical bounds are valid in practice. Since it's possible that the TTL approach could throw away important data, I will measure the number of false positives using the TTL approach as a means of evaluating its usefulness with a discussion of how detrimental these false positives might actually be to understanding and creating actionable data sets.

Finally, I will evaluate multiple state of the art algorithms current in the literature for optimizing triggering, detection, classification, sensor energy usage, and topological modeling and provide metrics to their usefulness for making progress towards the larger goals of providing a generally useful representation for DSNs, converting primitive sensor data into actionable insights, and providing a tiered approach to DSN data management and storage requirements. I will provide a discussion on whether these techniques are useful within the two domains that they are implemented, and if they are, how they contribute to the overall goals of this framework. We also hope to show which combinations of tertiary techniques provide the most traction in solving the overall goals of this framework.

I expect to deploy reference implementations before the end of 2018 with validated data collection beginning and continuing through Q3 2019. I anticipate writing my dissertation along side the deployment and data collection process and to be finished in Q3 2019.

1.5 Anticipated contributions of Laha

Laha hopes to make the following four contributions to the areas of DSNs and specially optimization and management of DSNs. First, the Laha design, a novel abstract distributed sensor network that provides two useful properties relating to data management, converting primitive data to actionable

data and tiered management of Big Data.

Second, an evaluation of the Laha abstract framework through the deployment of two Laha-compliant reference implementations, validated data collection, and several experiments that are used to either confirm or deny the benefits touted by Laha.

Third, two Laha-compliant reference implementations, OPQ and Lokahi, which can be used to form DSNs for the collection of distributed power quality signals and the distributed collection of infrasound signals.

Fourth, an evaluation of state of the art algorithms in practice and a discussion of their usefulness as applied to two different DSN domains.

Fifth, a set of implications for modern distributed sensor networks as a result of the evaluation of Laha. That is, how does the confirmation of denial of Laha's benefits affect the field of modern DSNs moving forward?

CHAPTER 2
RELATED WORK

This chapter reviews related work looking at defining Big Data in terms of DSNs, Big Data management, self-optimizing DSNs, predictive analytics and forecasting, optimizations to triggering, detection, and classification of signals-of-interest within the context of DSNs.

2.1 Big Data and Distributed Sensor Networks

Big Data is a term that is used to define either the characteristics of collected data or the processes involved for storing and analyzing collected data. Information that is considered Big Data provides a number of challenges.

One of the best and somewhat up-to-date (2014) reviews on Big Data literature is provided by the Presidents Council of Advisors on Science and Technology (PCAST) in their report to the White House[13]. In this review, Big Data is described using several definitions.

The first definition includes "high-volume, high velocity, and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making"[2]. This definition focuses on the characteristics of the data that make it "Big". In this context high-volume refers to the total amount of data that requires processing, high-velocity refers to the speed at which data arrives, and high-variety refers to the fact that sensor data is often heterogeneous and incomplete. The second part of the definition includes the terms cost-effective, innovative forms of information processing for enhanced insight and decision processing which hints at the fact that we need technology that is able to deal with these types of data characteristics while doing so within the limits of a system with the goal of refining the data to provide insights and decision making that wouldn't have been possible without the information processing.

A second definition[38] mentioned by the PCAST report rings more true to what Laha attempts to accomplish within the context of DSNs and says that Big Data "a term describing the storage and analysis of large and/or complex data sets using a series of techniques including, but not limited to, NoSQL, MapReduce, and machine learning". This second definition defines Big Data in terms of storage and analysis techniques and is a useful definition for describing the processes by which Laha and the Laha reference DSNs deal with distributed sensor data.

2.2 Distributed Sensor Networks and Big Data Management

There are many technologies for movement, transformation, and storage of sensor data. Current state of the art technologies include distributed streaming and computation engines such as Apache Kafka[17] or Apache NiFi[1]. Although these framework provide a lot of flexibility in terms of

transformations applied and data management, they do not provide automatic mechanisms for data management. Other, less known technologies are discussed in [14], but also suffer from the fact that they are flexible in moving large amount of data, but do nothing to address storage requirements or graceful degradation.

Another approach is to use compression techniques, such as those described in [33]. However, at scale, even data compression can not keep up with the approach of storing everything all the time.

There are many distributed computation engines and techniques which provide a generic framework for distributing computational tasks across multiple CPUs and multiple machines. The two that are generally receiving the most academic attention are MapReduce[10] and Apache Spark[42]. Although these computation engines are very generic and quite powerful, they can't easily inherit any of the optimizing benefits provided by Phenomena in the Laha framework.

The data grid[7] is a framework that was designed to provide two basic services the authors believe are fundamental for distributed management and analysis of large scientific datasets, storage systems and metadata management.

The paper "Data mining with big data"[41] constructs the HACE framework which is specifically designed for mining of insightful data from varied Big Data sets. Although this framework is useful for managing multiple streams of data and mining over multiple features, it does not attempt to provide an upper bounds of storage requirements or provide graceful degradation in the face of large scaling networks.

In terms of frameworks using aggregation to facilitate data reduction, Camdoop[9] is a framework that aims to push aggregation techniques from the edge the entire way to the sink. Camdoop was able to show positive results in data reduction while still maintaining semantic meaning. However, Camdoop was designed to run over simple data streams (such as word count logs) and its not known how this system would perform with more primitive types of data. Camdoop was designed to run within CamCube simulations and its not known how this would run in practice with a real DSN.

Rehman et al. created a big data reduction framework[36] and argue that reducing data early in the analytics process can lead to efficient value creation. This framework was designed specifically for enterprise customer Big Data analysis, but I believe some of the core tenants could apply to any Big Data problem. They argue that by performing data reduction early in the process its possible to lower service utilization costs, enhance trust between users and developers, and preserve privacy of users among other benefits.

Luan et al. in their paper on Fog Computing[20] describe data reduction and aggregation techniques by performing some of a subset of computations and data reduction on the edge of the network, such as in mobile devices (cellphones) or in servers that geographically located near the data acquisition sources. Aggregated data is then send from the edge devices to data sinks for

further analysis or action. One of the major difficulties with this approach is handling scale and being able to dynamically deploy resources to the edge as data streams scale up.

In a paper by Stateczny et al.[32], researchers work to determine if artificial neural networks can be used to provide Big Data Reduction for hydrographic sonar data. The researchers found that they were able to see some reduction, but ran into issues when the data was very dense. The research presented here also appears to be very domain specific.

2.3 Distributed Sensor Networks and Predictive Analytics and Forecasting

Anastasi et al.[4] breaks data predictions algorithms for DSNs into two classes. The first class of algorithms are defined as stochastic approaches and use probabilities and statistics to provide predictions. The other class is called time series forecasting and uses historical time series data to provide future predictions. An example of a stochastic model for predicting sensor data is the Ken model[8] which was developed for energy reduction by minimizing the data sent between sensors and sink nodes. This is accomplished by using a model of sensed data and only sending data when the sensed values at the sensor do not match what was predicted by the model. The model is built during a training phase in which a probabilistic density function (PDF) is generated for the model. Ken is flexible enough to provide models for different types of sensed phenomenon and can work anywhere where there are high correlations in time and space.

Time series forecasting algorithms typically use moving average, auto regressive, or auto regressive moving average models. The authors of the PAQ framework[35] uses auto-regression techniques to build a model of sensor readings that is compared between sensor node and sink nodes while providing probably correct error bounds. The SAF architecture[34], by the same authors, improves on the PAQ framework by refining the AR models and also adds the ability to not only detect outliers, but also detect inconsistent data. These approaches provide predictions for a single feature, however Laha provides the ability for DSNs to be multi-modal. The paper presented by Le et al.[19] uses time series forecasting, but provides multiple models which are switched out when the data changes. That is, given the current state of the network, a model is selected that is most likely to provide correct predictions. This is useful if a network has multiple features that can be used for forecasting.

Han et al.[12] create an approach for efficient mining of partial periodic patterns in time series databases. Research before this could only match periodic signals if the patterns were completely full, however the authors augment this approach to being able to find partial periodic signals which are more common in practice. The authors show that the signals can be recognized after 2 passes of the database. Keogh et al.[16] take a different approach with their Tarzan algorithm and instead of mining for known periodic signals, they come up with an approach to enumerate all “surprising”

patterns of data in time series databases. They use a statistical approach that works in linear time to determine if the occurrence of a data point differs from that expected by change. They found that their approach was more sensitive and selective than other approaches described in the paper.

2.4 Determining Topology and Localization

The paper by Langendoen and Reijers[18] provides comparisons for localization techniques of large DSNs. Langendoen's requires that the approaches are self organizing and do not depend on global infrastructure (such as GPS) , are tolerant to node failures, and are energy efficient. These constraints rule out other localization approaches such as GPS. One thing that differentiates Laha networks to Langendoen's is that Langendoen assumes a random distribution of sensor nodes where sensors in Laha networks are strategically placed. If there are a fraction of nodes that do know their location (anchor nodes), then there are several techniques that meet Langendoen's criteria including Ad hoc Positioning System from Niculescu et al.[22], the N-hop Multilateration Primitive by Savvides et al.[29], and Rabaey's work on robust positioning algorithms[27]. The three approaches all use three similar phases for localization: distance between anchor nodes and other sensors, position, and refinement. Laha hopes to provide sensor distance between sensors rather than physical distance. The above algorithms use flooding of the network for evaluate distance metrics, which may not be possible in Laha deployed networks.

When timing synchronization between nodes is sufficient, that is, the synchronization between sensors provides a timing accuracy of more than the Nyquist frequency for the signals of interest trying to be captured, it's possible to use arrival time of signals to provide metrics on sensing field topology and localization. This is the premise behind sets of algorithms that look at a single signal and the arrival times of that signal at multiple sensors along with possible direction and then attempt to provide an estimate of source signal localization. This has been performed in infrasound networks using the INFERNO framework as described by Perttu[25] and in other acoustic DSNs such as those used for efficient shooter localization (finding the source of a gun shot from collected acoustic signatures) in [11] and [21]. Localization of non-acoustic signals has also been shown in the literature. For example, Parsons et al. provide a method for localizing PQ disturbances by analyzing energy flow and peak instantaneous power for both capacitor energizing and voltage sag disturbances from sampled voltage and current data[24].

Although not related to determining the topology of a PQ network, there is research that can also find the optimal placement of PQ sensors given a the topology of the network. Won, et al.[40] provide an automatic method of placing PQ sensors on a known topology to maximize signal collection while minimizing the total number of required sensors.

2.5 Optimizations for Triggering

Triggering is the act of observing a feature extracted data stream for interesting features and triggering sensors to provide raw data for a requested time window for higher level analysis. Adaptively optimizing triggering is a way to tune triggering algorithms and parameters with the aim of decreasing false positives and false negatives. In this context, a false positive is triggering on a data stream that does not contain a signal of interest and a false negative is not triggering on a data stream that does contain a signal of interest.

Many of the optimizing triggering algorithms present in the literature exist to minimize sensor energy requirements and bandwidth requirements. This is addressed in great detail in the literature review by Anastasi et al. [4]. This is accomplished by reducing communications between sensor nodes and the sink. It's argued in [26] that the cost of transmitting a single bit of information from a sensor cost approximately the same as running 1000 operations on that sensor now. However, there is some contention on this topic as [3] argues that in some modern sensors computational requirements can equal or eclipse those of sensor communication.

One of the main drivers of optimization of triggering is to take advantage of the known sensing field topology of a DSN. This is often referred to in the literature as "topology control"[28]. When the topology of the sensing field is known and when there is an adequate density of sensors, Vuran et al. show that sampled data display strong spatial and temporal correlations[37]. This fact can be used to reduce the amount of duplicate sensor data that is transmitted, stored, and processed. Topology control is generally split into two categories, "location driven" where the location of the sensor is known and "connectivity driven" which aims to dynamically activate or deactivate sensors to provide complete coverage of a sensing field. Many of the location based approaches in the literature attempt to maximize the ability for sensors to communicate with each other, however Laha takes the approach that all sensors communicate directly with sink nodes eliminating the need for optimizing intra-sensor communications. One downside to location based approaches is that GPS sensors can be energy hogs and only work with directly line of site to the atmosphere. In these cases, a subset of sensors can be supplied with a GPS and the other sensor use additional techniques such as NTP or statistical analysis to determine location[18].

More details on topology control can be gathered in the reviews by Karl et al.[15] and Vuran et al.[37].

CHAPTER 3
SYSTEM DESIGN

Laha, which means, to spread or distribute in Hawaiian, is an abstract framework for distributed sensor networks that provides a means for turning primitive data into actionable insights, tiered management of voluminous amounts of sensor data. This major goals are in part accomplished by augmenting a DSN with the ability to adaptively optimize its bandwidth, detection, classification, and sensor device power requirements.

The Laha framework is made up of five levels that can be viewed conceptually as a pyramid (see 3.1). Primitive data entering the Laha framework is located at the bottom the bottom of the pyramid. As data moves upward through the levels, noise is discarded, less interesting events are discarded or aggregated into upper levels, events are given more meaning and context, and associations and predictions are made.

3.1 Big Data Management in Laha

The Laha framework acts as an adaptive sieve for filtering noise and uninteresting data collected from a DSN. In this way, each level only passes what it considers interesting to the level above it. All data at a particular level is garbage collected at specific intervals relating to its important to the DSN.

Each level only keeps data for a specified amount of time before it is garbage collected. As data moves up the pyramid, it is generally considered more useful and therefore has a longer Time to Live (TTL), the amount of the time the data lives before it is garbage collected. When a higher level detects "something interesting", the data contained in the time window of "something interesting" is copied into the levels above it and will still persist even though the original data is garbage collected. In this way, Laha preserves data from all levels if they are associated with interesting data. This also provides graceful degradation of services. The TTL is managed by the overall memory management of the system. Laha Actors are designed to work within the constraints of the TTL at different levels. If a constraint is broken, the Actor logs this issue. TTL can be optimized by Phenomena at each level to either tune for system performance or tune for decreasing of false positives and false negatives at different levels within the Laha hierarchy.

A summary of how data management in Laha is provided in table 3.1. Note that the TTL is configurable for each implementing network and the table provides default values.

3.1.1 Instantaneous Measurements Level

The Instantaneous Measurements Level (IML) receives raw, sampled data from the DSN. The amount of data received is determined by the sample rate of each device multiplied by the number

Figure 3.1: Laha Conceptual Model

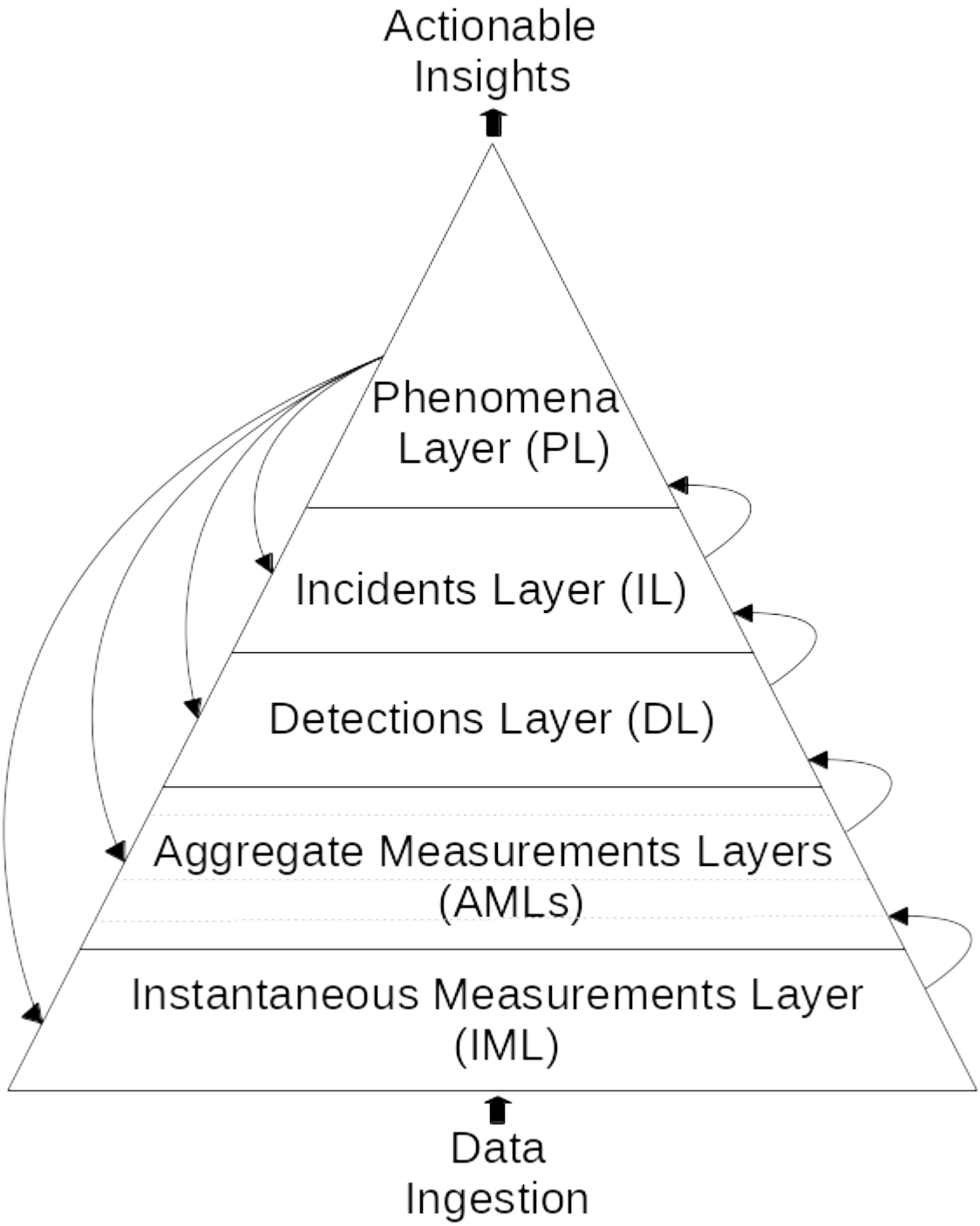


Table 3.1: Summary of data management and context addition in Laha

Level	Description	Time-to-Live (TTL)
Phenomena Level (PL)	Contextual & predictive analytics	
Incidents Level (IL)	Classified signals	1 year
Detections Level (DL)	Triggered windowed raw data	1 week
Aggregate Measurements Level (AML)	Statistical aggregates of raw data	1 day
Instantaneous Measurements Level (IML)	Raw sensor data	1 hour

of fields per sample. Most of the time samples, from devices in the network are mainly sampling noise. A large percentage of the data in this level is destined for garbage collection and data is assigned a Time to Live (TTL) of one hour.

3.1.2 Aggregate Measurements Level

The Aggregate Measurements Level (AML) is responsible for rolling up IMs from the IML. In general, this level only works with feature extracted data, rather than working with the raw samples. Each measurement in the AML provides summary statistics over a configurable time window. For example, these can include min, max, mean, median, mode, and variance statistics.

It's possible to breakup the AML into several sub levels, each with different window sizes. For example, Laha might roll IMs into one minute AMs, then roll one minute AMs into hour AMs, then days, and so on. Each sublevel within the AML can have its own configurable TTL, ensuring long term summary statistics stick around for as long as needed. This provides us a high level view of the network and can provide insights into long term trends which wouldn't be visible (or available) in the IM data stream.

Similar to IMs, AMs can be saved and copied to the levels above it when interesting data is observed. This ability allows for AMs during these time periods to be stored and saved from the garbage collection process.

At this point in the hierarchy, we are still not providing any context to the data that we are receiving. Context is provided by levels above the AML.

3.1.3 Detections Level

The Detections Level (DL) is the first level that provides some context to the data that the sink is receiving. This level is responsible watching the feature extracted data streams, and requesting IMs from the IM level. In general, the detection level is meant to be trigger happy[1] and be overly aggressive when determining if a feature extracted data stream looks interesting.

When a data stream looks interesting, the DL marks a timestamp N seconds before the interesting features and M seconds after the interesting features, where both N and M are configurable

[1] Pun intended.

Table 3.2: Summary of Laha Phenomena

Phenomena	Description
Annotations	Provide context about an Incident or set of incidents
Locality	Provides context on how incidents are related in time and space
Periodicity	Designation for incidents that exhibit repetitive or periodic behavior
Similarity	Subset of incidents found using grouping and community detection algorithms
Predictive	Subset of incidents characterized by predictive or forecasting models
Future	Incidents scheduled to occur in the future

within the framework. The goal is to use a time window that catches signals of interest within it. Since these data ranges will be further processed and refined higher in the hierarchy, there is no issue with collecting larges amounts of data in this level.

The actual methods of detection is dependent on the characteristics of every individual sensor network. This framework assumes that the detection algorithms are provided by the implementing frameworks.

Similar to other levels, the DL level will have its IMs and AMs copied into levels above it when upper levels observe something interesting in the DL. The detections level is set to have a TTL of a week.

3.1.4 Incidents Level

Incidents represent classified signals. Incidents are individual classifications for signals of interest and are created by analyzing waveforms from Events. Waveforms from Events may contain multiple incidents. Individual signals may be classified as multiple incidents (for example a transient being classified as both a transient and frequency incidents).

Incidents are further analyzed to produce phenomena.

Incidents are expired after one year of storage.

3.1.5 Phenomena Level

Phenomena are defined as a grouping of incidents that provide one or more of annotations, locality, periodicity, predictiveness, similarity, and future phenomena.

Not only do Phenomena provide interesting insight and analytics into the underlying data, but they also provide a means for adaptively tuning the underlying collection, triggering, detection, and analysis of a distributed sensor network.

Phenomena are summarized in table 3.2 and discussed in great detail in section 3.2.

3.2 Phenomena: Providing Adaptive Optimizations in Laha

3.2.1 Annotations Phenomena

Annotations provide context about an Incident or a set of Incidents. Annotations are generally user provided or sourced from other data sources to provide supporting context to Incidents. For example, Annotations might include Cloud Cover, Hurricane Hector, Dryer Turns On, etc.

In some sense, annotations allow us to label our data sets beyond a simple classification and start looking at causal classifications. Once enough annotations have been assigned to classified incidents, Laha can used Annotations to attempt to label unknown incidents with similar characteristics.

Annotations can be used to tune Laha's detection and classification algorithms by allowing Laha to filter on incidents with known causes.

3.2.2 Locality Phenomena

Locality provides context on how incidents are related to each other in both space in time. Laha is able to determine if classified incidents are local to a single sensor, to a group of co-located sensors, or global across an entire network. Sensors can be co-located in both the physical sense and also co-located within a sensing field. For example, sensors in a power quality network may be separated by large distance geographically, but co-located through the electrical grid and the grid's topology.

Over time, Locality Phenomena is used to build a model of sensors in relation to each other and to provide a statistical likelihood that co-located sensors will observe the same signal. Locality phenomena can be used to drive network triggering, detection, and classification thresholds within a distributed sensor network by using this probabilistic model for determining the likelihood that a sensor or sensors will observe a signal of interest.

3.2.3 Periodicity Phenomena

Periodic phenomena consists of incidents that exhibit repetitive behavior, that is, the same types of incidents appearing in cycles from single or multiple devices. Periodicity allows for the easy creation of Predictive phenomena.

Periodic phenomena can come from a single incident or from multiple incidents. Periodic phenomena allow us to either tune our network to find periodic incidents or tune the network to ignore periodic incidents depending on if the incidents are of interest. Periodic phenomena are especially useful in conjunction with Annotation phenomena as Laha can assign causality to the periodic signal.

3.2.4 Similarity Phenomena

Similarity phenomena utilize grouping and community detection algorithms to group incidents together by their features. Common features used for grouping include time, location, incident type, or incident features.

3.2.5 Predictive Phenomena

Predictive phenomena consists of incidents that are characterized by a predictive or forecasting models. Predictiveness is a behavior that can be inferred from all other phenomena types.

Predictiveness is the main driver behind optimizing the control of a distributed sensor network. If Laha can predict the types of signals that will arrive at sensors, then Laha can tune those sensors and the sink to either filter the signals (if the user is not interested in signals) or tune the sensors and the sink to be extra sensitive to those signals, possible detecting them even if they previously would not have been detected.

3.2.6 Future Phenomena

Predictive phenomena can be used to create future phenomena. Future phenomena are a statistical model of the likelihood of seeing an incident or incidents at future points in time. Knowing that a signal may occur with some probability allows Actor affected by those signals time to prepare for the signals.

3.3 Laha Actors: Acting on the Laha Data Model

Laha Actors act on the Laha hierarchy and provide one of two functions. Actors can move data from one level of the hierarchy upwards through the hierarchy when interesting data is requested by an upper level. Actors can also apply adaptive optimizations downwards through the hierarchy. The Laha framework can support multiple actors at each level. For example, the Incidents Level in our reference power quality network contains actors for each of the following functions: IEEE classified voltage events, frequency variations, power outages, excessive THD, and many others. The Incident Level Actors move data from these incidents upwards to the Phenomena Level.

Table 3.3 summarizes the actors that exist within the Laha framework and their purposes.

3.3.1 Actor Constraints

Actors at each level in the hierarchy are governed by a set of constraints. These constraints include the set of possible inputs, A_i, the set of possible outputs, A_o, the set of Actors it can receive data from, A_{ai}, the set of Actors it can transmit data to, A_{ao}, and a set of performance metrics that each

Table 3.3: Summary of Laha Actors

Actor	Purpose
IML Actors	Perform feature extraction and move aggregate data to AML
AML Actors	Perform triggering on data from IML, copy data to DL if interesting
DL Actors	Perform high fidelity feature extraction on possible detections
IL Actors	Perform classification and contextualization on possible detections
PL Actors	Generate predictive analytics and optimize the lower levels of the hierarchy

Table 3.4: Summary of Laha Actor Constraints at Each Level

Level	A_i	Ao	A_{ai}	A_{ao}	A_p
IML	Raw samples	Aggregate trends	n/a	AML	Data ranges available
AML	Aggregate trends	Detections	IML	DL	Data ranges available
DL	Windowed waveforms	Hi-fi extracted features	AML	IL	Data & features available
IL	Hi-fi extracted features	Contextualized incidents	DL	PL	Incident types available
PL	Contextualized incidents	Optimizations	IL	All levels	Optimizations available

actor must maintain, A_p. The constraints assigned to each Actor are determined by the hierarchy level in which the Actor resides.

Actors are responsible for reporting constraint violations and in this way, Actors are the primary provider of health, performance, and status metrics about the Laha framework.

The constraints for each level of Laha hierarchy is summarized in table 3.4.

3.4 OPQ: A Laha-compliant Power Quality DSN

OPQMauka is a middleware component of the Open Power Quality (OPQ) framework. The OPQ project provides a hardware and software solution for monitoring distributed power quality (PQ). The OPQ project was founded with the goal of studying how intermittent distributed renewable energy sources affect PQ not just at a user's home, but also within a user's neighborhood, between neighborhoods, and globally across the grid.

The OPQ ecosystem is made up of networked hardware sensors (OPQBoxes) and various software services (OPQMakai, OPQMauka OPQHealth, OPQView). Each of these software components are made up of individual services and plugins.

The OPQ system design is laid out in figure 3.2.

3.5 Lokahi: A Laha-compliant Infrasound DSN

Lokahi is a dynamic DSN that originally evolved as a distributed infrasound detection network. Infrasound is characterized as sound waves that are less than 20 Hz. Infrasound generally can not be deciphered by the human ear, but it can be detected using microphone and barometric

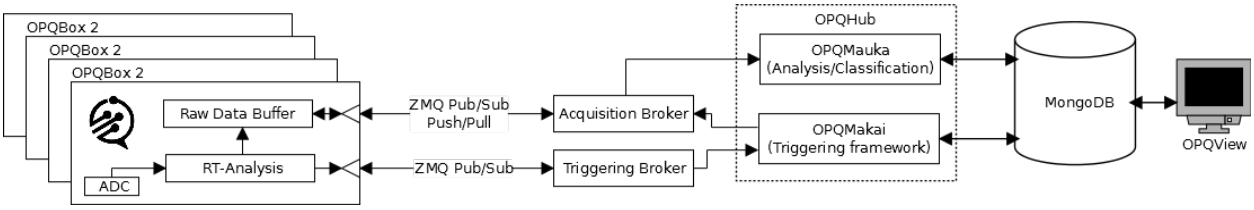


Figure 3.2: OPQ System Diagram

pressure sensors. Any large movements of the atmosphere can produce infrasound. The Lokahi network was designed to supplement the International Monitoring System (IMS) for the capture of undeclared and declared nuclear explosions. Lokahi has been successfully used to capture signals from volcanoes, hurricanes, aircraft, meteors, and other large atmospheric events.

Sensors in Lokahi are any mobile device that can run iOS or Android. We have sensors distributed world wide. The software stack for Lokahi consists of a distributed actor system for data acquisition, MongoDB for metadata persistence, Apache Kafka for data queues and interprocess communication, Python and related scientific libraries for analysis, and a distributed key-value store for long term storage or sensor data.

Recent development and improvements to the data API have allowed Lokahi to begin accepting data from any of the available onboard sensors on iOS and Android devices. Even though the main focus is still infrasound, having access to all of the available sensors provides the ability to sense other sensor fields and to perform interesting data fusion techniques.

A diagram of the Lokahi framework is provided in figure 3.3.

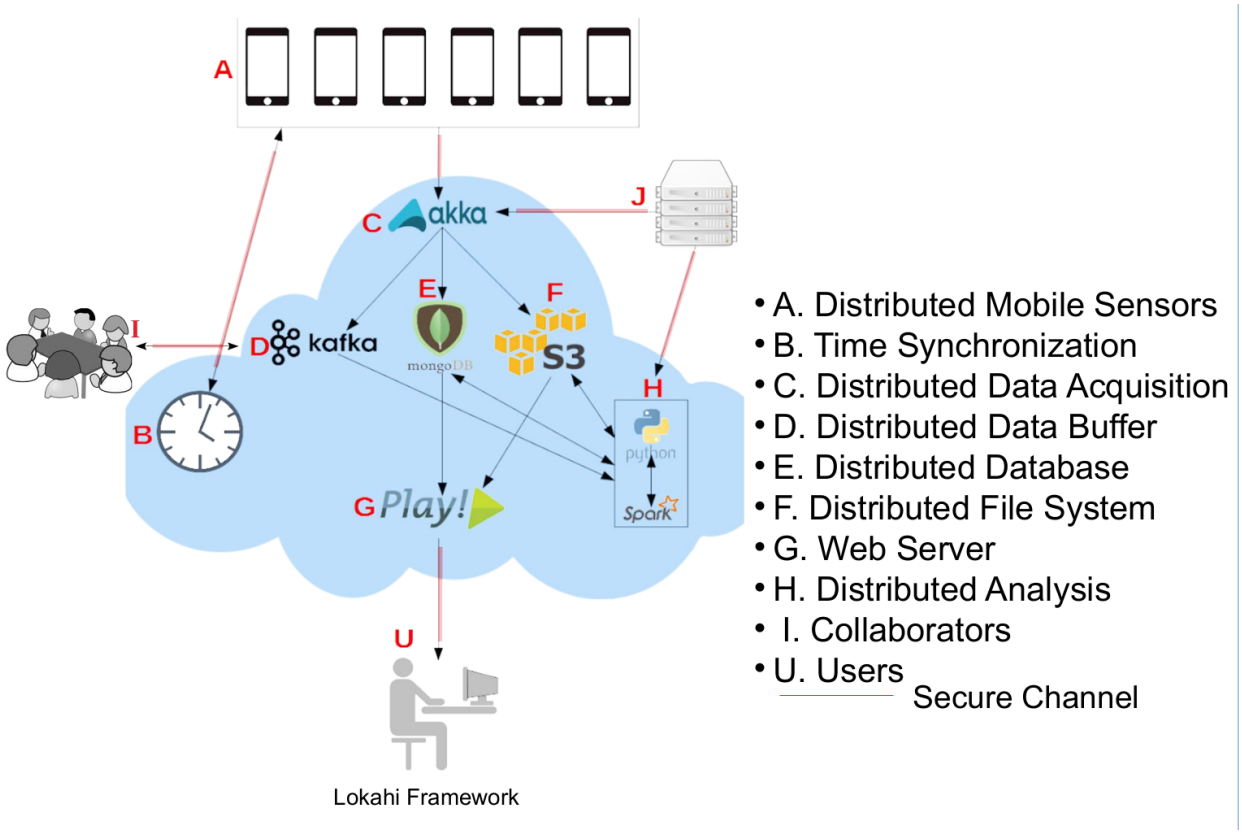


Figure 3.3: Lokahi Design

CHAPTER 4
EVALUATION

Evaluation of the Laha framework involves deploying reference Laha-compliant DSNs, validating the data collected from the reference implementations, and then comparing and contrasting various metrics for each of the proposed goals. Metrics will be collected during a set of experiments for each of the Laha reference implementations in early 2019.

The following sections describe my plans for deployment of reference implementations, data validation, evaluating the main goals of the Laha framework, and evaluating the tertiary goals of the Laha framework.

4.1 Deploy Laha reference implementations on test sites

In Q4 2018, 10 to 20 Laha-compliant OPQBoxes will be deployed on the University of Hawaii at Manoa's power microgrid. Using a provided blueprint of the microgrid as a guide and collaborating with the Office of Energy Management, these sensors will be placed strategically with the hopes of observing PQ signals on the same line, PQ signals generated from intermittent renewables, local PQ signals, global PQ signals, and PQ signals near sensitive lab electronics. Many of these sensors will be co-located with industry standard PQ monitoring systems. The industry standard sensors provide both ground truth and a means of comparison between a Laha designed network and a non-Laha designed network..

In Q4 2018, 20 to 30 Laha-compliant Lokahi sensors will be deployed near and around the Infrasound Laboratory in Kailua-Kona on the Big Island of Hawaii. These sensors will be placed strategically around a calibrated infrasound source. The sensors will be placed with the assistance of Dr. Milton Garces to ensure that I can target sensors at different distances by tuning the amplitude and frequencies of the infrasound signal. In this way, I know which devices should or should not have received the signal.

4.2 Validate data collected by Laha deployment

Beginning in Q1 2019, I will begin validated data collection from both the OPQ network and the Lokahi network.

Data will be validated in the OPQ network be comparing detected and classified signals against industry standard meters that are co-located with our sensors. Data validation will be an autonomous process that validates signals and trends seen in both the industry sensor and the OPQ sensors. Data validation will provide metrics for signals and trends that the reference sensors observed but OPQ sensors did not (false negatives) as well as signals that the OPQ sensors observed

and the reference sensors did not (false negatives). Specifically, I will be looking to compare long term trends (voltage, frequency, and THD readings over a time period of days) as well as more transient signals of interest (i.e. voltage sags/swells, frequency variations, excessive THD, and outages).

Data from the Lokahi network will be validated against industry standard infrasound sensors. We also control the amplitude and frequency of the signals generated from the calibrated infrasound source and can use geophysical equations to predict which sensors should have seen or not seen an infrasonic signal. Data validation is autonomous for this network as well. Similar to the OPQ network, I will be collecting metrics on false positive and false negatives as compared to the reference sensors.

Data validation for both networks will continue for all data collection until the end of the project.

4.3 Use Laha deployments to evaluate the main goals of the framework

The Laha deployments for both OPQ and Lokahi will be used to evaluate each of the main goals this framework claims to provide. Namely that Laha is a generally useful framework representation for DSNs. Second, Laha provides the ability to turn primitive sensor data into actionable data and insights. Third, Laha's tiered management of sensor data provides metrics on maximum bounds for storage requirements and graceful degradation of DSN performance.

Each deployment requires different techniques for performing evaluation.

In the OPQ deployment, OPQBoxes are deployed and co-located with industry standard, calibrated, reference sensors. Each of these sensors cost thousands to obtain and install, collect all the data all the time, and can only be connected to the power main as it enters a building. These sensors provide a means for verifying signals received or not received by OPQ, as well as confirming long term trend data. I have been provided access to these sensors and stored data via the Office of Energy Management at UH Manoa. The data is accessible via an HTTP API. The Office of Energy Management at UH Manoa has also provided the full schematics for the UH power grid. This will be used as a ground truth for topology estimates and distributed signal analysis. OPQBoxes are placed in strategic locations on the UH Manoa campus specifically in order to evaluate the distributed nature of PQ signals. For example, OPQBoxes are placed on the same electrical lines as well as separate electrical lines to observe how PQ signals travel through an electrical grid.

In the Lokahi deployment, I have the opportunity to generate infrasound signals using a calibrated infrasound source [23].. The source can be tuned to produce infrasound at configurable frequencies and amplitudes. The source works by attaching a variable pitch propeller to an electric motor that can be driven by a waveform generator. The source can generate signals that can be

observed at large stand off distances, over tens of kilometers. Similar to the OPQ deployment, sensors within the Lokahi deployment will be co-located with industry standard, calibrated, infrasound sensors. These sensors can provide a metric of signals that were correctly observed, incorrectly observed, or not observed at all by the Lokahi deployment. Further, infrasound itself is characterized quite well by various geophysical equations. These equations can be used to predict if sensors deployed in the Lokahi deployment are likely to observe generated infrasound signals.

Evaluation of the main goals of this network are provided in the following sections.

4.3.1 Evaluation of the Generality of this Framework

I claim that the Laha framework is useful and general enough to be applied to DSNs in different domains. To test this, I will design, develop, and deploy two DSNs. The first OPQ, measures distributed PQ signals on the electrical grid. The second, Lokahi, observes infrasound signals traveling through the atmosphere.

To evaluate the generality of the Laha design, I will provide metrics for whether or not each deployment is able to fulfill the goals of the given network.

I expect the PQ network, OPQ, to be able to detect and classify common PQ issues. I expect OPQ to observe voltage dips, voltages swells, frequency dips, frequency swells, transients, and high levels of THD. A count of these signals will be kept and compared against industry standard PQ meters co-located with each sensor. By comparing these signals to the ground truth, we will be able to tabulate a number of false positives and false negatives. In order to be considered effective, I would expect to be able to classify each of these common PQ signals, collect a set of each of the PQ signals while maintaining a low number of false positives and false negatives as compared to the industry standard sensors. In general, a negative result here would be not being able to detect PQ signals of a specific type or having a high number of false positives or false negatives.

Further, another stated goal of OPQ is to detect and classify distributed PQ incidents. That is, PQ signals that are observed by more than one sensor in situations where OPQ sensors are not co-located. First, I will evaluate if OPQ is capable of detecting distributed PQ signals. I expect OPQ to at least observe one distributed signal during the test deployment, but would not be surprised to see many. By working with the Office for Energy Management at UH Manoa, I will use a list of known PQ source events along with signals collected by OPQ and the industry standard sensors to provide a list of false positives and false negatives for the number of distributed PQ incidents observed by OPQ.

I expect the infrasound network, Lokahi, to be able to securely detect and report on infrasound incidents from a large collection of heterogeneous smartphone based infrasound sensors. This network prioritizes availability and security even in the face of network issues or no network at all. I claim that Laha is a useful framework for a DSN such as this and will evaluate if Laha is able to meet the goals of this network.

To evaluate the effectiveness of Laha as implemented by Lokahi, I will deploy 50 heterogeneous Lokahi smartphone sensors at predetermined distances from a calibrated infrasound source. I will then use the calibrated infrasound source to generate infrasound signals of different amplitudes and frequencies. While signals are being generated, I will disable network access for the sensors to simulate real life network drop outs of sensors. I will disable the networks for time periods of 1 minute, 30 minutes, and 1 hour.

I will then, for each sensor, calculate the number of false positives and false negatives for detections of infrasound signals. In order for Laha to be a useful framework for Lokahi, Lokahi must demonstrate that not only can it detect infrasound signals at different frequencies and amplitudes, but it must also do this while maintaining a low number of false positives or false negatives.

Further, as availability is a major priority of this network, network outages must be handled without signal loss. To evaluate this goal, I will measure the amount of false negatives (or missed signals) due to Laha's data management and the interplay with network outages. I would expect that if Lokahi implements it correctly, we should not see a rise in false negatives. A less great result would be an increase in false negatives.

Finally, backed by the metrics for both deployments, I will provide a critical discussion on what types of DSNs Laha is well suited for and what types of DSNs Laha is not well suited for. This will include a discussion on which parts of the Laha design are useful or a detriment to a given goal of the DSN.

The following sections continue to discuss the evaluation strategies required to show that Laha is a generally useful representation for a DSN.

4.3.2 Evaluation of Converting Primitive Data into Actionable Insights

An important goal of any DSN is to convert primitive sensor data into actionable insights. This is generally accomplished by adding some kind of context associated with the data such as classifications of a signal or linking the data with other data by comparing similarities in time, space, or other physical features.

I claim that Laha's use of Actors acting on and moving data between levels in the Laha hierarchy provides a useful and generic approach to systematically adding context to data as it moves through the framework. Laha is designed with a specific number of levels where data within each level shares the same type. In each deployment, I will evaluate the usefulness of each level with regards to adding context to the data.

An early approach to organizing data for contextualization is the Data Grid project[7] which proposed needing two services for building higher level extractions, storage systems and metadata management. This framework provided the context on top of data needed to easily build replication services for the data, which was important since one of the major goals of this framework was data availability and policy management. Data Grid also maintains data uniformity and does not allow

complex schemas. Data Grid does not provide a mechanism for discarding noisy data. Laha differs from Data Grid by providing support for complex metadata schemas, focuses on data reduction strategies, and provides more support for driving context. A more recent paper from Wu et al.[41] presents the HACE framework which is a framework designed for applying context to Big Data by making integration with other data sources and performing data fusion a first class member of the framework. This paper also examines algorithms for mining of complex and dynamic data, such as those generated from sensor networks. Laha differs from HACE by using a tiered approach to manage data volume while still hopefully generating actionable insights.

In both deployments, I will evaluate the number of false negatives for incident classification. Each level in the framework is responsible for not only adding context, but deciding if data should be moved upward through the levels, adding more context along the way, or discarding data because a level does not think the data is "interesting". I will keep track of the number of false negatives and which level was responsible for discarding the data with the signal. Using this approach, I will evaluate the effectiveness of each level to determine which levels correctly identify signals and which levels do not correctly identify signals, thus discarding the data.

In order to be useful, I expect each level to add context to the data while maintaining a low level of false negatives.

Using these metrics, I will provide a discussion on which domains a leveled approach may work well for versus which domains a leveled approach might not provide useful benefits.

I claim that Laha is able to provide even more context and actionable insights by implementing a level called Phenomena. Phenomena utilize predictive analytics to provide context and actionable insights over the sensor domain. First, I will evaluate if Phenomena take place in practice for both of the Laha deployments.

To evaluate Phenomena in the OPQ network, OPQ must observe a cyclical incident such as voltage swells occurring every afternoon due to solar output or an electric motor turning on at the same time every day. Once a cyclical incident is observed, OPQ must correctly create predictive Phenomena that predict the same incident happening in the future. Assuming predictive Phenomena are created, I will measure the amount of false positives and false negatives on whether the predictions were correct or not. A positive result would show that now only is OPQ capable of making predictive Phenomena, but also that a high percentage (¿ 50%) of the predictions are correct.

Evaluation of predictive Phenomena in the Lokahi infrasound network will follow a similar strategy. However, since I can control the infrasound source, I can actually run an experiment that creates cyclical and non-cyclical signals. I will then test Lokahi's ability to not only create predictive Phenomena, but also show that the predictions are accurate, that is, greater than 50% of them are correct.

A negative result would be that if either of the networks are not able to create predictive

Phenomena or a large number of false positives or false negatives (combining for ¡50% prediction accuracy).

Adding context to classified Incidents is the act of providing a statistical likelihood of the underlying cause of the Incident. These include things like showing that a voltage sag is caused by turning on the dryer every day at 2PM or an identifying as infrasound signal as a repetitive flight pattern near an airport. Context is provided by external sources to the DSN (such as users or by performing data fusion with other correlating data sets).

Evaluating contextualized events consists of setting up experiments where I assign context for a specific set of signals and resulting Incidents. Then testing to see if Phenomena are able to correctly apply context to Incidents when the same signals are generated again. I will record the number of false positives and false negatives for assigning context to Incidents.

A positive result would be to see the correct context applied to incidents more than half of the time. That is, I expect context to be applied correctly to at more than 50% of Incidents for which context has been previously defined.

I expect to see contextualization work better in DSNs where signals provide more measures for discrimination. For example, PQ networks contain many different types of classified PQ signals, however there is a small subset of causes attributed to each type of PQ signal classification.This decreases Laha's search space and in theory should make it easier to provide context.

4.3.3 Evaluation of Tiered Management of Big Data

The goal of tiered management of Big Data is to add a mechanism that provides a maximum bounds on storage requirements of sensor data at each level in the Laha hierarchy while simultaneously reducing sensor noise as Laha Actors move "interesting" data upwards. This in turn should decrease the amount of false positives since forwarded data is more likely to include signals of interest and less likely to be sensor noise.

Other approaches to Big Data management include compression[33] or storage systems where the goal is to have a distributed file system and move data close to where it is being processed, such as the Hadoop Distributed File System[39]. Other systems such as NiFi[14] provide a nice interface for ingestion and movement of data between Big Data tools while also providing data provenance, but do not go far enough in focusing on data reduction and graceful degradation. Carney et al.[5] discuss how monitoring applications require management and clean up of stale sensor data.

It's not yet known if I will see a decrease in false negatives. On the one hand, it's possible that Laha will throw away data that did contain signals of interest. In this case, detection or classification Actors will not observe the signals because the data has been discarded leading to increased false negatives. On the other hand, by reducing false positives and increasing the signal-to-noise ratio as data moves upward, Phenomena has a better chance of optimizing triggering, detection, and classification which may in turn inform Laha to save data that would have been

previously thrown away. In this way, it's possible that Laha will reduce false negatives.

We will evaluate the number of false positives and false negatives in detections, classifications, and Phenomena compared against industry standard reference sensors. A positive outcome for this metric would be a reduction in both false positives and false negatives compared to an approach that does not use tiered data management. A negative result would be an increase in either false positives or false negatives.

During the acquisition and curating of data, metrics will be collected and stored about how much data is saved (in bytes) versus how much data is discarded at each level within the Laha data hierarchy. These numbers will be compared against data storage as if the OPQ and Lokahi frameworks were to take a "store everything" approach. Evaluation metrics provided will include percentage of data storage saved per data hierarchy level as well as an estimate of overall decrease in data storage requirements for the entire DSN. A positive result from these metrics would show significant reduction in storage requirements for each level in the framework compared against a "store everything approach" and other state-of-the-art data storage solutions.

I will also provide metrics on "continuous storage pressure" which is a measure of the average amount of data storage required at each level given the current state of the network. That is, since data at all lower levels of the framework assigns a TTL to the data within the collection, the collection will exhibit a constant data pressure during sensor data collection. For example, at the lowest level, the IML collects raw data from all sensors all the time. Given the sample rate per sensor, the size per sample, the number of sensors, and a known TTL for this level, I can estimate the maximum bounds of data management requirements that the IML requires. We can develop similar estimation strategies with higher levels of the framework. I will compute the statistical error between the predicted storage pressure and the actual storage pressure recorded during the experiments. A positive outcome would show strong correlation between the predicted storage pressure and the actual storage pressure. A negative outcome would show weak correlation between the predicted and actual values.

Finally, I will provide an evaluation that weighs the results of all three metrics against each other. For example, if I see positive results for data storage reduction and negative results for false positives, do the benefits of the data storage reduction outweigh the negatives of increased false positives?

I would expect that DSNs that have a lower signal-to-noise ratio will see greater benefits from tiered data management than DSNs that already have a decent signal-to-noise ratio.

4.4 Evaluation of Tertiary Goals

In order to achieve the main goals of this framework, I claim that either all or a subset of the following tertiary goals must be fulfilled. Optimization of triggering, detection, classification, sensor energy usage, bandwidth, predictive analytics, and the ability to derive models of the underlying

sensing field topology.

To evaluate these tertiary goals, I will select and implement DSN optimization techniques from current literature. I will then compare and contrast the usefulness of different techniques and discuss how each of these techniques perform in the different sensor domains.

Finally, I will discuss how each of these tertiary goals make progress towards overall goals of this sensor network.

4.4.1 Evaluation of Adaptive Optimizations for Triggering

Triggering is the act of observing a feature extracted data stream for interesting features and triggering sensors to provide raw data for a requested time window for higher level analysis. Adaptively optimizing triggering is a way to tune triggering algorithms and parameters with the aim of decreasing false positives and false negatives. In this context, a false positive is triggering on a data stream that does not contain a signal of interest and a false negative is not triggering on a data stream that does contain a signal of interest.

Adaptive triggering is only useful in networks that utilize triggering. Specifically, this technique can not be applied to DSNs that take a collect everything all the time approach.

Triggering can also have significant impacts on overall sensor power requirements and DSN bandwidth requirements. Many of the optimizing triggering algorithms present in the literature exist to minimize sensor energy requirements and bandwidth requirements. This is addressed in great detail in the literature review by Anastasi et al. [4]. This is accomplished by reducing communications between sensor nodes and the sink. It's argued in [26] that the cost of transmitting a single bit of information from a sensor cost approximately the same as running 1000 operations on that sensor now. However, there is some contention on this topic as [3] argues that in some modern sensors computational requirements can equal or eclipse those of sensor communication.

Even if a DSN utilizes triggering, it's not clear that adaptive triggering even takes place. The first question I will evaluate is, does adaptive optimization of triggering take place at all given the domain of the DSN? That is, does the nature of the underlying sensor field contribute to optimization of triggering? I will compare if and how optimizations take place in the two reference networks for the domains of PQ and infrasound.

In order to evaluate triggering efficiency within our Laha deployments, Laha will only adaptively modify triggering for half of the devices in the OPQ deployment. In the Lokahi deployment, I will run the same experiment twice. The first run will not optimize triggering and the second run will optimize triggering.

Once the experiments have been run, I will first determine if optimization of triggering has occurred, and if it did, compare the number of false negatives and false positives against the runs that did not use optimized triggering or where optimization did not occur.

I hope to show that a side effect of Laha's optimized triggering is reduced bandwidth and sensor

energy requirements. To this end, I will calculate metrics for total data sent and received at the sink node of each network for each device in the network. A positive result would show decreased bandwidth usage for devices that utilize optimized triggering. A negative result would show similar or more bandwidth usage for devices that utilize optimized triggering.

I further hope to show that another benefit of Laha's optimized triggering is reduced sensor energy requirements. The evaluation for this metric will occur with the Lokahi network where sensors can be dependent on batteries. I will run two experiments. For each experiment, all sensors will be charged to battery level of 100%. In the first experiment, I will not utilize optimized triggering. In the second experiment I will utilize optimized triggering. In both experiments, I will measure the final battery level after the experiment and also measure how quickly the battery depletes for each sensor. This is possible because data in the Lokahi network contains timestamped entries with battery levels.

4.4.2 Evaluation of Adaptive Optimizations for Detection and Classifications

Detections occur when triggering observes something "interesting" in the feature extracted data stream. A Detection is a contiguous window of raw sensor data that was requested by triggering that may or may not contain signals of interest. Optimizing detections involves optimized the window sizes to increase the signal-to-noise ratio of the window. Fine grained features are then computed by Detection Actors and moved to the Incidents Level where classification of signals takes place. Optimizing Detections involves trimming detection windows to increase signal-to-noise. Optimizing of classifications for Incidents involves tuning parameter sets for the underlying classification algorithms.

Predictive and Locality Phenomena as well as topology optimizations will be used to provide optimizations to the Detections and Incidents levels.

Evaluation of adaptive optimizations for detection and classification within the Laha network will be conducted differently for each Laha deployment.

In the Lokahi deployment, I will control the production of infrasound signals using the available infrasound source. I will run two experiments, where the amplitudes and frequencies of the signals are the same and the locations of the devices remain invariant. In the first experiment, Laha will not use optimized detection or classification provided by Phenomena. In the second experiment, Laha will use optimized detection and classification techniques provided by Phenomena.

With known frequencies and amplitudes of the infrasound signals, I can compare the rate of detections and classifications between the optimized and unoptimized experimental runs. I expect to see a greater number of and more accurate detections and classifications from the optimized experiment.

In the OPQ deployment, I will compare the same metrics as the Lokahi deployment, but instead of controlling the source signal, I will co-locate OPQBoxes. In each pair of co-located OPQBoxes,

one will be analyzed using Phenomena optimized detection and classification algorithms and the other will be analyzed using unoptimized detection and classification algorithms.

I will collect and evaluate the number of false positives and false negatives for Incidents generated with optimization and without optimization. A positive outcome would include a decrease in either false positives, false negatives, or both. A negative result would be an increase in either or both false positives or false negatives.

I will also calculate the signal-to-noise ration in Detections to determine if optimization of detections is working. A positive outcome would be an increase in the signal-to-noise ration and a negative outcome would be similar or a decrease in signal-to-noise ratio.

4.4.3 Evaluation of Model of Underlying Sensor Field Topology

Laha should be able to build a model of the underlying sensing field topology. This is not the topology of the physical layout of the sensors (this is generally already known a priori or by collecting location information), but rather the topology by which signals travel. For example, in a PQ network the topology is the physical power grid and switches that PQ signals travel through. In an infrasound network, the topology is the atmosphere through which sound waves travel. Laha aims to build a statistical model of the distances between sensors according to the topology of the sensing field by observing recurrent incidents over time. This can perhaps shed some light on understanding the topology of a sensing field without knowing anything about it before hand.

Much of the literature on topology management is written to decrease sensor energy requirements by exploiting the density of sensors within a sensing field topology. For example, the ASCENT[6] framework provides adaptive self configuring sensors that exploit topology denseness to decrease sensor energy usage. Several other frameworks have been designed with the same goal of reducing energy usage by exploiting topology[31],[30].

To evaluate the model of the sensing field topology, I will take two different approaches for each Laha deployment. In both deployment, the sensing field topology is known beforehand to provide a ground truth. I will then compare Laha's computed signal distance between sensors to the actual signal distance between sensors as provided by the ground truths.

In the Lokahi deployment, sensors will be strategically placed at different distances from an infrasound source. Some sensors will be close to each other geographically, but separated by terrain that infrasound signals will not easily travel through. By moving the infrasound source, I can expect to see infrasound signals arriving or not arriving at the sensors depending on the source and direction of the signal along with the physical features of the land. By performing multiple experiments, I hope to provide a model of the physical environment topology that Laha has built. I will compare Laha's model to the known topology and provide a statistical error analysis.

In the OPQ deployment, sensors will be strategically placed on like and unlike electrical lines to observe how distributed PQ signals move through a power grid. In this deployment, Laha will build

a topology model that doesn't show physical geographic distance between sensors, but instead will build a model of the electrical distance between sensors. This data will be evaluated by comparing the electrical distances found by the Laha model to the actual UH power grid as referenced by the schematic provided by the Office of Energy Management at UH Manoa. A statistical error analysis of the differences between electrical distances between the model and the schematic will be provided as an evaluation metric.

A positive outcome would be to show that there is high correlation between the Laha signal distances and the ground truth distances. A negative outcome would show low correlation.

Assuming high correlation and a statistical model of the sensing field, I would like to evaluate if Laha is able to use this information to optimize triggering, classification, or predictive analytics. In order to evaluate this, I will collect the number of false positives and false negatives at all levels in the Laha hierarchy while optimizing from topology and without optimizing from topology. I expect to see less false positives and less false negatives when utilizing topology optimizations. A negative result would be a larger number of false positives or false negatives.

I expect to only see results in networks where signals travel fast enough to create a statistical difference between arrival times at the various sensors. In sensing fields where signals travel slowly and uniformly (i.e. a temperature collection DSN), it may be more difficult or impossible to actually determine the sensing field topology.

BIBLIOGRAPHY

[1] Apache NiFi.

[2] What is big data? - gartner it glossary - big data, Dec 2016.

[3] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri. An Adaptive Sampling Algorithm for Effective Energy Management in Wireless Sensor Networks With Energy-Hungry Sensors. *IEEE Transactions on Instrumentation and Measurement*, 59(2):335–344, February 2010.

[4] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, May 2009.

[5] Don Carney, Uur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Monitoring streams: a new class of data management applications. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 215–226. VLDB Endowment, 2002.

[6] Alberto Cerpa and Deborah Estrin. Ascent: Adaptive self-configuring sensor networks topologies. *IEEE transactions on mobile computing*, 3(3):272–285, 2004.

[7] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of network and computer applications*, 23(3):187–200, 2000.

[8] David Chu, Amol Deshpande, Joseph M Hellerstein, and Wei Hong. Approximate data collection in sensor networks using probabilistic models. In *null*, page 48. IEEE, 2006.

[9] Paolo Costa, Austin Donnelly, Antony Rowstron, and Greg O'Shea. Camdoop: Exploiting in-network aggregation for big data applications. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 3–3. USENIX Association, 2012.

[10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[11] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84, 2005.

[12] Jiawei Han, Guozhu Dong, and Yiwen Yin. Efficient mining of partial periodic patterns in time series database. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 106–115. IEEE, 1999.

[13] White House. Big data and privacy: a technological perspective. *Washington, DC: Executive Office of the President, Presidents Council of Advisors on Science and Technology Google Scholar*, 2014.

[14] James N Hughes, Matthew D Zimmerman, Christopher N Eichelberger, and Anthony D Fox. A survey of techniques and open-source tools for processing streams of spatio-temporal events. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, page 6. ACM, 2016.

[15] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.

[16] Eamonn Keogh, Stefano Lonardi, and Bill'Yuan-chi' Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556. ACM, 2002.

[17] Jay Kreps, Neha Narkhede, Jun Rao, et al. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, pages 1–7, 2011.

[18] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer networks*, 43(4):499–518, 2003.

[19] Yann-Aël Le Borgne, Silvia Santini, and Gianluca Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing*, 87(12):3010–3020, 2007.

[20] Tom H Luan, Longxiang Gao, Zhi Li, Yang Xiang, Guiyi Wei, and Limin Sun. Fog computing: Focusing on mobile users at the edge. *arXiv preprint arXiv:1502.01815*, 2015.

[21] Miklos Maroti, Gyula Simon, Akos Ledeczi, and Janos Sztipanovits. Shooter localization in urban terrain. *Computer*, 37(8):60–61, 2004.

[22] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1734–1743. Ieee, 2003.

[23] Joseph Park, Milton Garcés, and Bruce Thigpen. The rotary subwoofer: A controllable infrasound source. *The Journal of the Acoustical Society of America*, 125(4):2006–2012, 2009.

[24] Anthony C Parsons, W Mack Grady, Edward J Powers, and John C Soward. A direction finder for power quality disturbances based upon disturbance power and energy. In *Harmonics and Quality of Power Proceedings, 1998. Proceedings. 8th International Conference On*, volume 2, pages 693–699. IEEE, 1998.

[25] AB Perttu, MA Garces, and WA Thelen. Regional localization with the hawaii island infrasound network. In *AGU Fall Meeting Abstracts*, 2013.

[26] Gregory J Pottie and William J Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[27] C Savarese J Rabaey, Koen Langendoen, et al. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX technical annual conference*, pages 317–327, 2002.

[28] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM computing surveys (CSUR)*, 37(2):164–194, 2005.

[29] Andreas Savvides, Heemin Park, and Mani B Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 112–121. ACM, 2002.

[30] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Topology management for sensor networks: Exploiting latency and density. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 135–145. ACM, 2002.

[31] Curt Schurgers, Vlasios Tsiatsis, and Mani B Srivastava. Stem: Topology management for energy efficient sensor networks. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–3. IEEE, 2002.

[32] Andrzej Stateczny and Marta Wlodarczyk-Sielicka. Self-organizing artificial neural networks into hydrographic big data reduction process. In *International Conference on Rough Sets and Intelligent Systems Paradigms*, pages 335–342. Springer, 2014.

[33] Caimu Tang and Cauligi S Raghavendra. Compression techniques for wireless sensor networks. In *Wireless sensor networks*, pages 207–231. Springer, 2004.

[34] Daniela Tulone and Samuel Madden. An energy-efficient querying framework in sensor networks for detecting node similarities. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 191–300. ACM, 2006.

[35] Daniela Tulone and Samuel Madden. Paq: Time series forecasting for approximate query answering in sensor networks. In *European Workshop on Wireless Sensor Networks*, pages 21–37. Springer, 2006.

[36] Muhammad Habib ur Rehman, Victor Chang, Aisha Batool, and Teh Ying Wah. Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management*, 36(6):917–928, 2016.

[37] Mehmet C Vuran, Özgür B Akan, and Ian F Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245–259, 2004.

[38] Jonathan Stuart Ward and Adam Barker. Undefined by data: a survey of big data definitions. *arXiv preprint arXiv:1309.5821*, 2013.

[39] Ajit Warrier, Sangjoon Park, Jeongki Min, and Injong Rhee. How much energy saving does topology control offer for wireless sensor networks?–a practical study. *Computer Communications*, 30(14-15):2867–2879, 2007.

[40] Dong-Jun Won and Seung-Il Moon. Optimal number and locations of power quality monitors considering system topology. *IEEE Transactions on power delivery*, 23(1):288–295, 2008.

[41] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2014.

[42] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.