SSC17-I-02

Eating Your Own Big-Data Dogfood: Large Satellite Performance with Small Satellite Hardware

Nicholas McCarthy, Daniel CaJacob, Darek Kawamoto
HawkEye360
13873 Park Center Road. Suite 550N. Herndon, VA 20171; 571-203-0360
nick@he360.com

ABSTRACT

Tool sets, algorithms and technologies developed to create value from the availability of big data have potential not only to justify and reward the collection of sensor data from space but also to improve the quality of sensor data collection. The 2018 HawkEye360 Pathfinder mission will demonstrate balancing constrained, space-based compute platforms hosting sensor hardware with an approach to ground-segment data processing typical of cloud-based, Big Data analysis to maximize the performance of payload hardware on-orbit. We present specific examples related to the improvement of time- and frequency-of-arrival (TOA and FOA) estimation for AIS transmissions. Using a small corpus of raw AIS data captured from commodity hardware on planes over the Chesapeake Bay, we investigate early-prototype machine-learning models and test hypotheses as to on-orbit collection improvements. Providing a description of the compute resources available as part of the HawkEye Pathfinder payload, we discuss system design considerations and practical approaches to deploying payload sensor data collection enhancement as part of an automated system for smallsat data collection, ingestion and enhancement. Typical Big Data business models involving power-sensitive commodity hardware sensors at the periphery of a system serviced by a backbone of cloud compute resources have evolved a number of effective open-source and academic software resources amenable to the smallsat use case. We describe the HawkEye Pathfinder analytic software stack, focusing on how it leverages code and concepts developed to enable Big Data processing and how those concepts extend to facilitate improved sensor data collection as part of a mutual feedback system between space and ground processing components. Limitations facing the application of techniques derived from Big Data analytics to the problem of enhanced payload data collection via emitter characterization arise as part of the system design discussion. We posit ideas for mitigating these factors through the application of predictive analytics.

INTRODUCTION

Accurate geolocation by most traditional means comes down to measuring accurately TOA and FOA for one or more emissions at one or more receiver locations. In [17], the author characterizes Cramér-Rao lower bounds for TOA and FOA measurements in terms of a signal's integration time, noise bandwidth, root mean square (RMS) radian frequency, RMS integration time, and effective input signal to noise ratio (SNR), γ. Where TOA and FOA estimates are given by measuring correlation and ambiguity peaks from two separate digitized representations of the emission with input SNRs γ_1 and γ_2, the author offers the equation

$$\frac{1}{\gamma} = \frac{1}{2}\left[\frac{1}{\gamma_1} + \frac{1}{\gamma_2} + \frac{1}{\gamma_1\gamma_2}\right] \tag{1}$$

for γ, along with the following two observations. When $\gamma_1 \approx \gamma_2$ and $\gamma_1 \gg 1$ dB, we see $\gamma \approx \gamma_2$. On the other hand, when $\gamma_1 \gg \gamma_2$ and $\gamma_1 \gg 1$ dB, we see $\gamma \approx 2\gamma_2$. Striving for as much as possible of that factor of 2 motivates a great deal of algorithmic and logistical complexity, especially in the context of estimating TOA and FOA in the context of a low-powered, low-earth-orbit (LEO) satellite mission.

Small commercial satellites present a business opportunity because the size, weight and power (SWAP) constraints imposed mostly by launch costs correspond to countervailing trends toward better compute efficiency at the lesser-capable end of computing market, not the top. Software Defined Radio (SDR) presents an attractive mission activity because it allows for operating flexibly within a wide frequency range. Configured such that a wide frequency range of operation combines with the ability to capture radio frequency (RF) emissions from a broad footprint on the surface of the Earth in an orbit such that the effective footprint for RF collection revisits each coordinate frequently, smallsat SDR presents an effective compliment to imaging platforms featuring highly directional sensors.

In Section II, we indicate sources of loss to our link budget due to design choices concerning a balance of mission objectives and cost effectiveness.

A degradation in received SNR will irreversibly limit the theoretical lower bound for standard deviation in estimating TOAs and FOAs of RF bursts, but equation 1 shows technique can go a long way toward making up for added white Gaussian noise (AWGN) and the like. Using a standard additive model for noise, η, combined with a time series of samples, y, transmitted through a channel and received as samples, x, we write

$$x[t] = y[t] + \eta[t] \tag{2}$$

for later reference.

We can achieve as much as a 3 dB rise in effective input SNR to TOA and FOA measurement if, rather than correlating digitized signals $x_1[t]$ and $x_2[t]$ received by similarly situated receivers to find TOA estimates for the same burst, we leverage an understanding of the signal to produce a digital copy, $\hat{y}$ to approximate the original signal y and apply correlation-based techniques using $\hat{y}$ and x_i for $i \in 1, 2$. We would hope $\mathrm{SNR}(\hat{y}) \gg \mathrm{SNR}(x_i)$ and $\mathrm{SNR}(\hat{y}) \gg 1$, satisfying the desired condition. For our exemplar signal, AIS, a public specification makes estimating y to a first-order approximation tractable. We know, for instance, AIS uses GMSK which specifies a modulation index h of .5 ([11]), and time-bandwidth product BT should be between .3 and .5 ([15], Section 3). Like any AIS receiver, we should have the ability to process an AIS message and derive the underlying bits. Following the specification for a transmitter should allow us to generate an ideal digital copy of the original sampled burst, stripped of noise, η, injected by the channel. If in Section II we draw an admittedly arbitrary line between exquisite and non-exquisite systems, we draw another line here between exquisite and non-exquisite software processing techniques. Specifications for real-world communications systems include tolerances to account for imperfections in analog hardware such as amplifiers and filters. Commodity AIS transceiver providers build systems to meet a price point, and the resulting parts list can make for real-world communications, indeed. The AIS specification provides guidelines as to ramp up and ramp down times for the amplifier, and defining GMSK features such as BT and h need only remain true to the specification inasmuch as a general-purpose GMSK receiver must successfully process the output. In Section III, we investigate attempts to move beyond the AIS specification in composing $\hat{y}$ by attempting to estimate variation from the specified ideal value for h and to apply that estimate to our model for AIS transmission. Additionally, we measure improvements to realized standard deviation in TOA for both lab generated data and live captures, and we estimate the complexity our processing chain accrues in implementing estimation techniques for h.

While Section III demonstrates we can, to some extent, overcome hardware limitations through the surgical use of software, it does not address any of the logistics behind accomplishing the surgery itself. If we could take for granted an unlimited downlink channel, disinterest in latency, unlimited compute power on the space platform, or a rational market for data, no matter how raw, based on the potential for value from that data, then there might be no need to continue. Of course, none of these assumptions holds. Operating small satellites is an exercise in resource management and triage, and our case study on accuracy in AIS burst arrival statistics shows these problems need not cease once the data hits the ground. In Section IV, we situate AIS TOA and FOA processing within the definition of Big Data processing. RF processing from satellite predates Big Data by decades, but the problems Big Data imposes on data scientists motivate solutions entirely relevant to the problem at hand. We analyze a full-system implementation for processing AIS metadata using a stream-processing software architecture, mapping features designed for Big Data processing onto the problem of resource-constrained AIS arrival estimation.

Recalling design choices from Section II, HawkEye360 has skewed physical properties of its satellite sensors such that it can succeed in bringing data to bear on the problem of focusing more highly directional sensors such as imaging satellites. In Section V, we outline plans to develop predictive analytics for normative ship statistics. Furthering the theme of the paper, we show inference need not go exclusively in one direction: predictive models designed as input to products capable of tipping location-specific processes for partners, customers, and other stakeholders can just as easily tip HawkEye360 assets, making those same products more robust. Deep insight into AIS signal externals beyond that available through analyzing message content can serve as a basis for rich situational awareness and increasingly meaningful anomaly detection.

SPACE SYSTEMS

Development of space-based systems, which are inherently resource-constrained and remote, invariably results in compromise between the ideal system and the practicable one. Smallsats necessarily cope with smaller

SWAP allowance than space-based systems generally, and this allowance limits performance to even greater degree. HawkEye's Pathfinder cluster of 3 micro satellites is no different. While we can certainly imagine the perfect RF geolocation satellite cluster, we recognize that such an ideal is not practical within the size and budget constraints allotted to a small startup company. Moreover, our mission forces us to make trade-offs between global coverage and antenna directionality.

The HE360 Pathfinder cluster has an ambitious mission. If HE360 had as its mission simply to approximate TOA and FOA for AIS collection, then the design might have looked starkly different. A focused mission might require less power or enable heavier processing since our engineers could optimize performance for one or a small number of algorithms, perhaps going as far as to design a custom ASIC for high-performing, high-cost routines. We could budget for one or more frequency-specific antennas or antennas with more directionality and gain. As it stands, The Pathfinder mission must support many diverse algorithms to address numerous signals ranging from AIS to radar. This mission set requires a flexible software framework. The SDR payload has a tunable frequency range of approximately 100 MHz to 15 GHz. The extremely wide frequency range necessitates a suite of RF front ends specialized for frequency subranges and an antenna farm including dipoles, patches, a horn and more. We build all this into a spacecraft not much larger than a microwave oven. Pathfinder antennas are, generally, not very directional. This ensures that they will see as much of the Earth as possible but comes at a cost: more susceptibility to co-channel interference and less gain. Similarly, the satellites' limited power budget bounds the processing capability that can be brought to bear in orbit. The spacecraft are not designed to be exquisite platforms. However, concessions made to accommodate such flexibility are offset by processing on-board in software and FPGA, ground processing, and a system tying both processing segments together. The ground segment ingests data processed on-orbit, and feeds conclusions back to the spacecraft to form a virtuous cycle.

CORRELATION AND AMBIGUITY PROCESSING: REDUCING VARIANCE IN PEAK ESTIMATION

Due to its varied mission, the HawkEye Pathfinder RF front end will provide less than optimal frequency and spatial selectivity for AIS processing, resulting in decreased SNR for RF presented to the payload processor. This decrease in SNR is cause for concern in geolocation processing. As given in [17], the best possible standard deviation for TOA estimation is

$$\sigma_{\mathrm{TOA}} = \frac{1}{\beta}\frac{1}{\sqrt{\gamma B\tau}}, \tag{3}$$

where, for AIS, $\beta \approx 12500$Hz, B is the channel bandwidth of the signal, τ is the integration time, and γ is the effective SNR.

In addition to SNR attenuation, modulator parameter mismatch can contribute to further losses in TOA estimation. In particular, when the modulation index h used to generate the digital copy $\hat{y}$ is incorrect, the correlation result can easily take a 10 dB penalty. The AIS standard, [3], specifies $h = 0.5$, nominally, but allows for deviation from this number, so h-mismatch is an especially important concern manifesting as a significant implementation loss term in TOA estimator performance.

Due to decreased link margin resulting from RF system compromises, it is imperative for HawkEye to mitigate any losses caused by modulation parameter mismatch in order to live up to the promise of exquisite collection.

Motivation: Quantifying Implementation Loss

Among modulation parameters we have considered and at ranges of deviation from specification we have observed, cross-correlation performance is most sensitive to the modulation index, h. While the very definition of GMSK stipulates $h = 0.5$, in practice we see AIS transmissions with h ranging from about 0.45 to 0.55. Given the sensitivity of our processing to this single parameter, we choose to process AIS as if it were a GFSK signal.

Figure 1 shows the performance of TOA estimation for simulated data when the TOA estimator uses $\hat{y}$ generated with a matching modulation index h; SNR is measured as $\frac{E_b}{N_0}$.

On the other hand, Figure 2 shows the performance of the same TOA estimator where we generate $\hat{y}$ with $h = 0.5$. When h is mismatched, the estimator loses two to three orders of magnitude in efficiency. The main reason for this loss in estimator efficiency is the introduction of accumulated phase error terms in the generated digital copy $\hat{y}$.

Solution Approach and Simulation

Figures 1 and 2 suggest improved TOA processing can be had by estimating h independently for each burst.

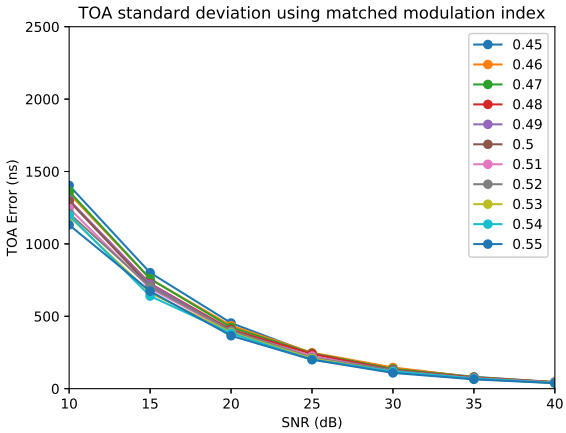

Fig. 1. TOA Estimation Performance (Matched Modulation Index).

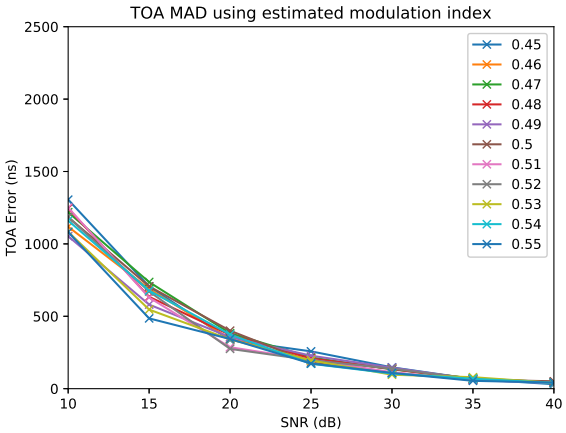

Fig. 3. TOA Estimation Performance (Estimated Modulation Index).

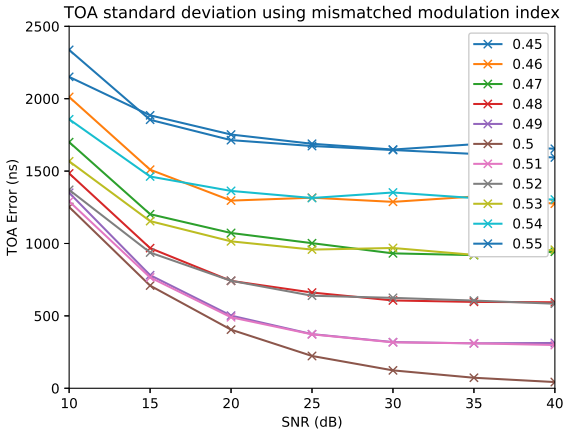

Fig. 2. TOA Estimation Performance (Mismatched Modulation Index).

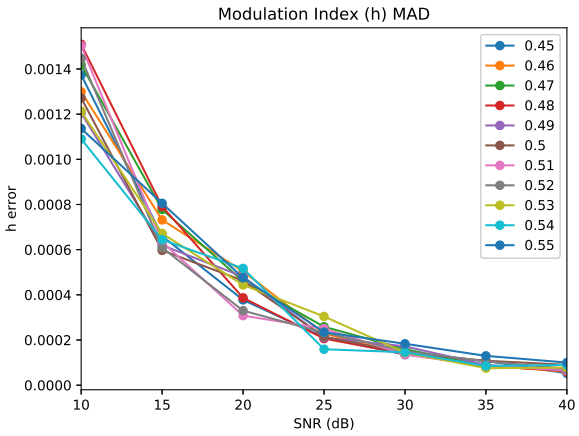

Fig. 4. Modulation Index Estimation Performance.

We apply a joint-estimation approach. The prototype algorithm is a numerical optimizer on h, TOA, and FOA seeking to maximize the cross-correlation score.

AIS simulation blocks were written within the GNU Radio framework. Random bits were loaded into the payload of AIS messages, and a GFSK modulator with $BT = 0.5$ and a uniform random $h \sim \mathcal{U}(0.45, 0.55)$ were used to modulate the messages. A random amount of AWGN was added to the messages so that the SNR varied between 10 and 40 dB, in 5 dB increments.

Figure 3 shows MAD values for the TOA estimator described. MAD is a scaling of the median absolute deviation of a set meant to approximate standard deviation for the (presumably Gaussian) distribution underlying that set with robustness to outliers; see [14], [12]. If we define $\mathcal{E} \coloneqq \{e_i | e_i = t_i - \tau_i\}$ where t_i ranges over a set of TOA estimates and τ_i is the known TOA corresponding to measurement t_i, then

$$\mathrm{MAD}(\mathcal{E}) = 1.4826 \text{ median } \{\forall i \, |e_i|\} \, . \tag{4}$$

We choose this robust metric to tolerate occasional numerical instabilities in the joint estimator resulting in outlier measurements. Operationalizing this approach will require improvements to algorithmic stability.

After outlier rejection, the joint estimator has reduced the TOA estimator variance to roughly the same levels as the estimator using the correct modulation index, showing that modulation parameter estimation successfully mitigates the implementation loss associated with parameter mismatch. Figure 4 shows the MAD error in the modulation index estimates.

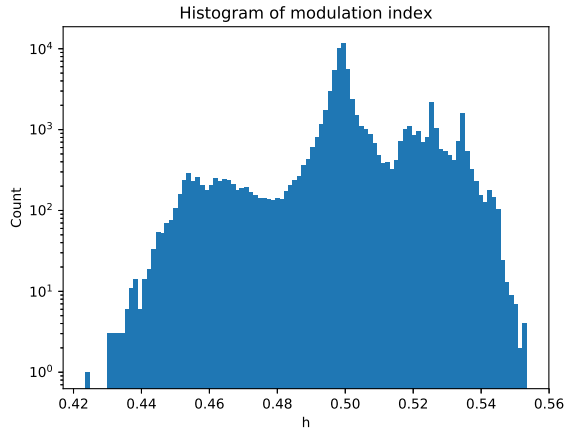

Fig. 5. Histogram of Estimated Modulation Index.

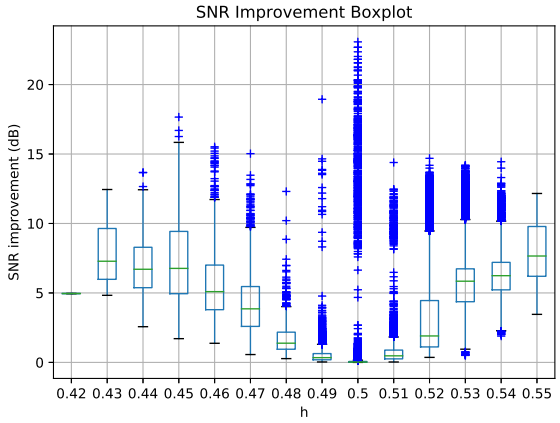

Fig. 6. SNR Improvement By h.

Real Data Results

In 2016, HE360 performed airborne collection of AIS transmitters in the Chesapeake Bay. Replaying these recordings through an AIS processing chain, we are able to compare TOA correlation scores using the nominal value for h against those from the joint estimation algorithm.

Figure 5 shows the estimated modulation index for all AIS packets with SNR of at least 10 dB. While most of the processed packets have $h \approx 0.5$, a significant number of packets do not.

Since SNR values can be mapped uniquely to expected correlation peak values, the inverse of this function can be used to estimate the recovered SNR of the signal. The correlation values (with and without the joint estimator) were converted to equivalent SNR values. SNR improvements, grouped by h, are shown in Figure 6. The boxes with their horizontal lines in the plot show the 25th, 50th, and 75th percentiles, while the whiskers show thresholds for outliers. Outliers are shown as blue crosses. Since the joint estimator optimizes against correlation peak value, correlation peak value (and hence the derived SNR measurement) will always improve as a result of the process.

For $h \approx 0.5$, the recovered SNR for a majority of packets does not improve, but recovered SNR for a non-trivial number of packets improves significantly (as much as 20 dB). Optimizing over h, TOA, and FOA jointly might allow for more precise FOA and TOA measurements even where resulting estimates for h are identical to the nominal value. As h deviates from 0.5, the median SNR improvements generally increase.

These results indicate that jointly estimating h with TOA and FOA significantly increases correlation scores and mitigates implementation loss.

Computational Considerations

Adding another dimension to a numerical optimization routine does not come without added computational burden. In particular, the cross-correlation surface is not as well-behaved across the h-dimension as it is in the TOA and FOA dimensions. In order to cope with this, multiple initial estimates need to be explored before using gradient-based methods to maximize the correlation score. Exploring the parameter space in this way requires extra computation, resulting in an estimated 5-10x increase in computation versus using the nominal value for h.

As we will discuss in Section IV, the added computation is too much to handle for the embedded payload computer on HawkEye Pathfinder satellites. This motivates us to explore different strategies to distribute computation between the space payloads and ground processing infrastructure with a limited downlink bandwidth between the two.

In particular, utilizing prior information about modulation indices corresponding to AIS transmitters could limit parameter estimation overhead on-orbit. Additionally, the algorithm developed here has room for careful computational improvement; further research and better understanding of the correlation surface could lead to some shortcuts and increase what is possible on-orbit.

BIG DATA PROCESSING FOR AIS VALIDATION

Contextualizing HawkEye Data within Big Data

The most widely accepted definition for Big Data centers on the three V's as described by Doug Laney in 2001 and revisited in 2012: velocity, variety, and volume. ([4]) Roughly speaking, a Big Data problem presents substantial challenges because of the the sheer volume of data required as input to the solution, because of the variety in data needed (often referring to grappling with unstructured or variously structured data, but also having to do with fusing differently-natured data), and because of demands for processing data for use in real-time. We consider enhanced AIS arrival processing from the standpoint of this definition.

From Section III, we see enhanced AIS arrival processing requires as input raw in-phase and quadrature (I/Q) time series data sampled at above the critical rate for the AIS signal. AIS signals have a symbol period T of $1/9600$ seconds. We assume the ability to account for Doppler on-orbit so that we can digitally downconvert each AIS message to DC. Given the spectral efficiency of GMSK (we estimate excess bandwidth to be $.4$), capturing the signal at a rate of 19.2 kilo samples per second will provide data sampled well above Nyquist. Although 12-bit in-phase and quadrature (I/Q) samples generally provide sufficient dynamic range for RF captures, we use 16-bit ints to correspond to existing instruction set types for ARM and x86 processors and then rely on data compression to make amends. Assuming every satellite can see at least one AIS message at any given time (our link margin for non-directional AIS collection allows for a viable footprint of radius about 2500km), we can set a reasonable daily estimate for data volume given continuous full-take I/Q collection per satellite:

$$
\begin{aligned}
&(19200 * (2 * 16))\,\text{bits/sec/channel} * 2\,\text{channels} \\
&* (3600 * 24)\,\text{sec/day} \\
&* (1/8)\,\text{bytes/bit} * (1/2)\,\text{compressed bits/bit} \\
&= 6.636\,\text{GB/day.}
\end{aligned}
$$

AIS signals are on the lower end in terms of bandwidth among signals of interest for HawkEye360, but physical and cost limitations at the downlink receiver put both absolute and practical limits on the extent to which our initial satellites can produce volumes of data. Using the following assumptions, we derive an anticipated budget for downlink capacity using a commercial x-band downlink transmitter offering 3 and 25 Mbps modes of operation.

$$
\begin{aligned}
&((3\,\text{Mbps} * 6\,\text{passes/day}) + (25\,\text{Mbps} * 4\,\text{passes/day})) \\
&* 6\,\text{mins/pass} * 60\,\text{s/min} * 1/8\,\text{bytes/bit} \\
&* (1 - .05\,\text{overhead}) = 5.04\,\text{GB/day.}
\end{aligned}
$$

Already, downlink capacity serves as a bottleneck for the processing pipeline, and the analysis completely ignores a host of other products HawkEye hopes to provide, be it RF survey, raw RF capture, or processing and analysis for a number of different specific RF emissions. As a point of comparison, [4] suggests Twitter facilitates sending 300,000 tweets per day, or 62.2 GB, and that number forms an extreme lower bound for Big Data volume: the same site suggests self-driving cars will generate 2 petabytes per year.

Downlink rates determine an upper bound for HawkEye's total self-generated data volume, and that upper bound does not meet the standard for Big Data. As regards the second V, our own data is structured: we define for ourselves our own formats. Of the three V's, only velocity stands out for this case study as an unadulterated Big Data challenge. Since base station downlink events occur irregularly, data ingestion patterns necessarily resemble a batch processing regime, but the most valuable data comes with an incredibly short shelf life. In some sense, HawkEye strips data of its original momentum between collection and downlink and has a responsibility to impart downlinked data with velocity it originally possessed.

Strictly speaking, then, processing HawkEye-only data does not constitute a Big Data challenge. Nonetheless, software and stratagems related to handling Big Data impact this use case in more ways than are related to latency constraints. In the case at hand, two factors force us to deal with AIS captures and associated data via a process of down-selection. At issue is cost per bit to downlink data, value per bit processed, and a distribution of computational resources making it impossible in most cases to consider doing the later without doing the former. Software running on the edge of the HawkEye system, at the point of RF reception, can access orders of magnitude more data than the downlink budget for the satellite hosting the sensor, but systems at the edge are incompletely connected to other sensors and under-provisioned in terms of computational resources. This situation has echoes in writing parallelized, distributed database lookups and in managing the flow of sensor data from cell phones in toward service providers and Big Data companies. Ultimately, any volumetric data problem reduces to successful data down-selection. Even if the Big Data landscape features ramping up data storage and search capabilities such that hard decisions

need not be made at the point of collection, manipulating the resulting data stores in a productive way involves reducing the data to contextually relevant, digestible products. For HawkEye, the forcing function putting a premium on analysis to determine how to down-select data is not the sheer number of records in a growing database or the sheer number of sensors in the network, but the limited number of bits we can bring down from space. Deriving a mechanism to determine comparative value among the bits available to download reveals itself to be crucial to the value the entire system can provide. To bound this problem, we notionally construct each system component required for enhanced AIS collection encompassing the entire system, space and ground. The system turns out fundamentally to rely on context derived from data external to HawkEye, putting the system in contact with data sets of diverse structure and character and forcing us to contend with Big Data variety.

System Component Descriptions and the Payload Processor

In constructing our system for enhanced AIS burst collection, we begin by analyzing the value of the final product, deducing the function and placement of the components necessary to obtain that value in the face of unavoidable engineering constraints. We consider how the enhanced look at AIS processing in Section III adds value to the data produced by a system as a whole. After all, AIS messages nearly always include a self reported location accurate to within GPS tolerances. A service purporting to provide locational information orders of magnitude worse than information already contained in the message provides very little value. In reality, the value proposition for processing based on AIS signal externals rests on the potential for corruption in the AIS message internals. Summarizing [8], AIS is designed to be open, not secure, in order to facilitate ease of use for collision avoidance. For a brief introduction to some of the many AIS spoofing scenarios available to the enterprising seafarer, we refer the reader to [6] (emphasizing the category of threats described by the author as protocol-specific). In particular, even if we can trust the entire receive chain used to process an AIS message, any aspect of the message including position, cargo, flagged country, destination, port of origin, heading, Mobile Maritime Service Identity (MMSI), and weight is subject to both intentional and unintentional falsification. Nor does this 2014 report constitute idle speculation. In the context of detecting illegal fishing, [9] introduces a reality immediately evident upon examining AIS data sets of sufficient size. Instances of MMSI numbers used simultaneously by multiple transmitters abound. AIS transmitters use operator input to determine MMSI broadcast values, and evident operator error (or operator indifference) shows up as repeated detection of MMSI numbers such as 888888888 or 000000000. “Certainly not all spoofing represents illegal or undesirable activity, but it is unquestionably one method illegal fishers use to conceal their identity and their behavior,” determines the author in [9]. According to [10], Global Fishing Watch collects more than 20 million AIS messages per day, amounting to the following volume of data:

$$20000000\,\text{messages/day} * 256\,\text{bits/message} * 1/8\,\text{bytes/bit} = 640\,\text{MB/day}.$$

so revealing AIS spoofing fails to pose a volumetric challenge by any reasonable standard. Still, the author in [9] approximates .25% of AIS messages are spoofed, and common sense suggests intentional spoofing must infect a much lower percentage. We appeal here to common sense because we know of no authoritative source for statistics: catching AIS spoofers in the act is hard. The ease with which unintentional AIS spoofing can occur presents a layer of indirection obscuring intentional spoofers and further rewarding their efforts. Nor does the seemingly small quantity of intentional AIS spoofers serve to distract attention from spoofed AIS as a valuable source of information. It may be that only a small percentage of all maritime behavior is illegal, but stopping that illegal behavior in such cases as piracy, over-fishing, human trafficking, and other offenses presents an incredible upside. AIS provides essentially no means of authentication in any sense, and increasing reliance by analytics platforms on AIS as a means of identification and situational awareness make it all the more attractive a target for spoofing and other attacks. For the purposes of this study, value in enhanced AIS arrival processing stems from an ability, ultimately, to gain insight quickly into AIS spoofing.

With respect to AIS spoofing, we begin to shape the fundamental components of the system from the ground up, starting with a description of the services providing most potential value. HawkEye requires its processing system to detect and correct for AIS spoofing such that the resulting data may be used as though AIS provided a trusted means for authenticating the message contents. In extreme cases, such as during the commission of a criminal act, the ability to discover spoofed AIS and to report it can provide immense value, as can improved geolocation accuracy derived from enhanced TOA and FOA arrival processing, where that value decays rapidly on the order of minutes following data capture on-orbit. In such instances, forwarding data to parties such as law enforcement or imaging satellite services with enough accuracy and in a timely enough

manner can mean the difference between halting a crime in progress and letting it go unpunished. Given the capabilities described in Section III, HawkEye has two interconnected means to furnish the needed data at low latency. Whether through enhanced or ordinary TOA and FOA arrival processing, HawkEye can, for every AIS message collected, perform an independent geolocation of the transmitter. In so doing, HawkEye can both validate the content of the message referring to transmitter position and provide geolocation accurate enough to track potential spoofers. The second means depends on enhanced processing involving the modulation index, h, and has as a byproduct of better geolocation results the potential to detect anomalies in AIS broadcasts going beyond transmitted location. Section I touches on BT as well as amplifier ramp-up and ramp-down, additional signal externals which combined with h can help to distinguish one transmitter from another. Put another way, in the process of sharpening geolocation results for AIS bursts, HawkEye gains a basic emitter identification capability. If HawkEye downlinks enhanced TOA and FOA arrival processing results along with the I/Q collection needed to determine signal characteristics, then HawkEye can perform an independent validation for the reported MMSI. This conclusion presupposes HawkEye has collected enough historical I/Q AIS burst data to provide a statistically accurate description of signal characteristics per transmitter. Starting from a formulation of the highest-value data we wish to produce, we have determined at least in part what data we need from the payload processor as well as a need for an accurate assessment of AIS transmitter characteristics on a per-emitter basis and a requirement for detecting and reporting AIS spoofing with low latency.

Without a means to share TOA and FOA between receivers in a cost-effective way, the payload processors cannot use the arrival information as input to calculating geolocation independent of AIS signal internals, and Section III shows analysis needed to determine benchmark values for such signal externals as h imposes too great a strain on the power and computing budget to perform routinely on-orbit. In whatever way we might expect to use AIS processing to shed light on spoofing, the payload processor alone cannot plausibly detect and downlink only the relevant AIS bursts without tying into a larger system involving processing on the ground.

What can we afford to downlink? If we derive a nominal TOA and FOA for each AIS burst via DSP performed on the payload computer, save only the cyclic redundancy check value, an estimate for the burst's SNR and the demodulated message using an ordinary serialization protocol (for this example, we use Protocol Buffers), then we can downlink metadata associated with a single message using (on average) 60.42 bytes. According to [11], AIS divides each minute into 2250 slots of duration 26.7ms. Our estimate for a day's metadata collection follows, assuming we can process one and only one message per slot.

$$60.42\,\text{bytes/message/channel} \\ * 2250\,\text{messages/min/channel} * 2\,\text{channels} \\ * (60 * 24)\,\text{mins/day} = 391.52\,\text{MB/day}.$$

Provided we can build and maintain a database of AIS signal externals mapped to emitter MMSI (where the mapping is not necessarily unique), we propose taking the following course. We maintain a copy of the resulting table on each payload processor, uplinking deltas to the table during base station passes. During AIS burst processing, but prior to TOA and FOA estimation, we read the entries, indexed $\{1, 2, \ldots, n\}$, in the table corresponding to the processed MMSI. We derive arrival estimates $\text{TOA}\,(h_i)$ and $\text{FOA}\,(h_i)$ separately for each h_i, $i \in \mathcal{I} \coloneqq \{1, 2, \ldots, n\} \bigcup \{s\}$, where $h_s = .5$ as prescribed by the specification. For each index i in $\mathcal{I}$, we derive a confidence indicator c_i by measuring the correlation peak of the received signal against the matched filter generated assuming h_i. Along with the metadata associated with normal processing (including $\text{TOA}\,(h_j)$ and $\text{FOA}\,(h_j)$ for $j = \arg\max_{i \in \mathcal{I}}(c_i)$, we downlink j and c_j, providing not only a guess discriminating among emitters with the same MMSI but a probability associated with that guess as well. For very common MMSI values such as 000000000, we forgo enhancements, settling for h_s. The payload processor may encounter a message from an emitter as yet unprocessed by the ground segment and for which, even if the table has an entry mapped to the message's MMSI, that value is not derived from the emitter in question. In such a case, the processor will estimate FOA and TOA using a non-causal value h_j for some $j \in \mathcal{I}$, but the resulting estimates $\text{TOA}\,(h_j)$ and $\text{FOA}\,(h_j)$ should prove at least as good as for h_s the majority of the time. (It is, to be precise, possible $\text{TOA}\,(h_j)$ is less accurate $\text{TOA}\,(h_s)$, but the expected value for the accuracy of $\text{TOA}\,(h_j)$ is better than the expected value for the accuracy of $\text{TOA}\,(h_s)$ because $c_j \geq c_s$.) In such cases, we expect the value c_j to reflect the fact h_j did not result from analysis performed by RF data emitted by the emitter in question, and weakness in a downlinked c_j can serve as justification either to prioritize future RF capture for the MMSI in question (to better constitute the table) or as evidence (perhaps not strong evidence) the emitter in question is currently using a spoofed MMSI. In corner cases, different receivers processing the same burst may select

TABLE 1
PROCESSOR LOAD BY BURST PROCESSOR COMPONENT.

demodulation	100%
AIS burst detection	37%
TOA/FOA estimation	22.5%
resampling	22%
matched filter generation	4 %
total compute resources occupied	66.4%

different values for j. Again, TOA and FOA should not suffer, on average, versus using the process for normal metadata processing since the matched filter used to generate TOA and FOA correlates at least as strongly with the received burst as does the matched filter derived from h_s.

To determine the the feasibility of such a mechanism, we start with the total number of emitters possible. According to [5] there existed in 2012 a total of 104,304 propelled merchant ships weighing 100 gross tons or more, whereas [1] cites the existence of more than 650,000 total AIS emitters (including fixed position emitters such as light houses and port entries). Settling on an estimated number of 500,000 total RF emitting vessels, the cost in memory allocation to store a unique putative float value for h corresponding to each emitter is only 2 MB. Doubling that number to account for MMSI multiplicity, the entire table should fit easily into the 1 GB of RAM available to the payload processor (a Zynq-7000 series SoC) alongside all other memory requirements. Nor do repeated lookups into a 500,000-size table pose any real challenge, even to an aging dual-core ARM clocked at under 1GHz. The downlink budget expands to 423.92 MB/day assuming we represent each c_j as a float and each j as an unsigned char. The processing overhead for AIS messages increases on-orbit.

Demodulation occupies the largest share of CPU resources, and we express all components as a percentage of demodulation resource occupation in Table 1. (Note: total compute resources occupied assumes processing for both AIS channels under a projected model for reception of AIS messages on-orbit in Table 1.) In order to operate in enhanced metadata processing mode, we repeat matched filter generation and TOA/FOA estimation $|\mathcal{I}|$ times for each detected burst. Carrying forward our assumption as to the multiplicity of MMSIs, $|\mathcal{I}| = 3$, so we see an increase in compute resources of 28.6% versus the normal metadata processing mode. Budgeting for the increased computational load for enhanced processing, we would anticipate 85% occupation. We caveat this number with several observations. Our estimates for overall performance rely on several implementation details for which a handful of candidate solutions exist. The most conservative implementation brings the CPU resource occupation for AIS burst detection relative to demodulation up to 75%, and operating in normal processing mode occupies 80% of total compute resources. Operating in enhanced processing mode would demand 99.0%. The payload processor requires some overhead in order to remain responsive to incoming commands and to allow for variation in computational load without risk of dropping data at the point of ingest. In flight, the payload processor will accommodate a number of low-level processes not reflected in these CPU occupation estimates. Whereas initial estimates on the order of 85% CPU occupation suggest we could operate full-time in the proposed manner, the thin margin for error evident in our estimates leads us to consider a further wrinkle. If we have underestimated $|\mathcal{I}|$ or overestimated our eventual algorithmic performance, our processing will prove unsustainable. As such, we consider a hybrid approach. Enhanced metadata processing would be standard operating procedure, but background processes would monitor compute overhead. Above a threshold percentage, the process would revert to normal metadata processing. Accepting this modification, building and maintaining a database for AIS externals appears to present the greatest barrier to bootstrapping from a mode allowing collection using only normal metadata processing to a mode combining normal and enhanced metadata processing, harnessing gains in TOA and FOA estimation described by Section III.

Already, we have estimated the downlink cost associated with full-take I/Q AIS burst capture versus metadata collection, just as we have described our primary motivation to incur the added cost. Namely, full-take I/Q collection allows us to estimate AIS modulation parameters specific to the emitter of each burst serving a dual purpose: parameters found to be incongruous with parameters associated to the reported identity of the emitter provide evidence of AIS spoofing, and all collected estimated parameters help to establish a baseline against which we can compare future collections for incongruity. As our downlink budget does not allow for sustained full-take RF collection, emphasis falls on targeted RF collection: down-selecting from among all bursts which to downlink as full-take RF. We focus here on down-selecting based on MMSI and statistics related to modulation index h. Targeted collection within the HawkEye system presupposes a ground-based targeting algorithm, but selection occurs on orbit such that limitations pertaining to the payload processor apply generally to the process of deriving selection criteria. Presented with an AIS burst, the payload processor must extract features from the data using processes from

among those fitting within the budget for the processor's computational load. We recycle numbers corresponding to the processing subcomponents in Table 1, as the results of those processes can be used as features for burst selection.

At a computational expense over and above AIS burst detection, the processor can determine the MMSI associated with a burst via demodulation, and our discussion of enhanced metadata processing shows TOA and FOA estimation can provide statistics reflecting transmitter-specific characteristics such as h. We assume the processing load associated with deriving features related to h it is exactly the load associated with enhanced processing. Referring again to Table 1, if we can get away with extracting the MMSI only, we project having to occupy 56.91% of compute resources in order to perform targeted RF collection. Nor does this processing go to waste for deselected bursts. While performing targeted RF collection, the processor can and should perform TOA and FOA estimation as per our combination of normal and enhanced metadata processing. The payload processor need not repeat execution of burst processing components needed for TOA and FOA estimation already performed as part of feature extraction.

In terms of the value proposition outlined above, operating via targeted RF collection succeeds or fails based on the algorithm used to target certain bursts among all of those detected. Aligning the workings of the targeting algorithm with the dual goals of full RF collection, we begin to define how the algorithm must operate. First and foremost, the targeting algorithm should attempt never to omit collection for a burst likely to have resulted from spoofing. Second, burst downlink should take place in such a way that prioritizes bursts most likely to strengthen the algorithm used to achieve the first and cardinal goal. To remain sensitive to payload processing constraints, we design a system with two layers, one corresponding exclusively to MMSI such that burst selection may take place subject to MMSI fitness alone. At this juncture, historical AIS data comes fully into play, as do data and analysis from a number of sources with various interests in the shipping industry and other aspects of maritime behavior. To every MMSI $m \in \mathcal{M} \coloneqq \{0, 1, \ldots, 999999999\}$, we assign a conditional $P(S|m)$ in the following (simple) manner, where $P(S)$ is the prior probability for any burst that the transmission is spoofed. First, a number of organizations maintain watch lists associating certain vessels (easily associated with MMSIs via sites such as MarineTraffic) with having demonstrated prior bad behavior. Trygg Matt tracking ([2]), a searchable site dedicated to tracking illegal fishing behavior is one example. This site provides context for each listing including organizations responsible for flagging the vessel and historical listing and de-listing status such that a data ingestion tool could easily instrument a mechanism for varying priority among MMSIs on the list. We combine externally-sourced information with observations such as the fact that certain MMSIs, simply because of their structure, indicate spoofing (often of an unintentional nature). Essentially, we build a black list for in-depth RF collection, assigning probabilities to each entry using domain knowledge. Casting the black list in terms of probabilities serves to limit collection based on reason rather than random deselection in cases where downlink or computational processing resources cannot keep up with demand.

Bringing historical AIS data to bear on MMSI prioritization requires more sophistication. A recent article, [16], captures basic principles by which historical (and constantly updating) AIS data provided by a number of terrestrial and satellite processors can serve to prioritize full RF collection through pattern recognition and anomaly detection. In [16], the author references a search for gaps in vessel movement records indicating with some likelihood a period of "going dark," or disabling AIS transmission, and combining this behavior with evidence of another established fishing vessel pausing in the area for long periods of time. Global Fishing Watch engineers posit such behavior can indicate a likelihood (not a certainty) of transshipping occurring. Again, behavior recognition techniques depend to large extent on domain knowledge. Organizations have succeeded in identifying illegal fishing and other bad behavior as documented in [16], and industry insiders can describe identifiable behavior, but deep learning and other methods leveraging automated feature extraction remain out of reach owing to a paucity of training data. Anomaly detection, used in conjunction with pattern recognition or independently, provides still more input to the burst targeting algorithm. Most behavior patterns indicate bad behavior only in certain contexts. Within the context of the success story in [16], ships may go dark regularly to avoid detection by pirates. Anomaly detection can provide powerful leverage to discern between contexts for a single recognized pattern without appeal to expert knowledge (assuming the targeted behavior pattern lies outside the normal range of all behavior). Applying regression analysis to behavior pattern detection using samples of geospatially similar historical data can help to determine outlier status. If ships commonly go dark in waters typically plagued by pirate activity, then behavior pattern detection within that geospatial context should happen regularly. Extending this line of reasoning, any

anomalous behavior might warrant further investigation via full-take RF analysis. Moving in this direction, HawkEye has experimented with historically anomalous vessel-type reporting in AIS messages. A near-limitless supply of labeled data exists for this problem, making it possible to deploy machine learning and even machine learning with automated feature selection. Essentially, these methods train a model to recognize vessel type independent of the reported vessel type value using time-contiguous collections of AIS position data as input. The resulting model, a classifier, can come to a probabilistically weighted assessment of vessel type and provide a posteriori evidence of vessel-type spoofing conditioned on the reported type. Each behavior pattern recognition and anomaly detection scheme runs within the ground architecture using a continuously updated feed of AIS data, periodically finding events based on messages from one or a handful of MMSIs m, and updating $P(S|m)$ accordingly. Finally, we take into account the health of our own data archive. For any m for which we lack any or recent RF data, we prioritize collection for that MMSI by manipulating a second probability $P(F|m)$ conditioned on m. Here $P(F)$ is the prior probability associated to HawkEye providing a false identification for any burst given the basic emitter identification capability described above. Weighting $P(S|m)$ and $P(F|m)$ according to relative values w_S and w_F associated with detecting AIS spoofing and HawkEye database enrichment respectively, we derive a priority $\mathrm{Pri}\,(m) \coloneqq w_S * P(S|m) + w_F * P(F|m)$.

The second layer for targeting burst collection requires on-orbit enhanced metadata processing. Recalling notation from Equation 2, we have already outlined much of the basic premise: for each burst y, we demodulate to find MMSI m and estimate $\mathrm{SNR}\,(y)$ by finding

$$C_m(y) = \max_{\{\mathrm{h,BT,ru,rd}\}} \left(\hat{y}_{\{h,BT,ru,rd\}} \star y\right). \tag{5}$$

Here ru is a single real variable parameterizing AIS burst ramp-up, rd is a single real variable parameterizing AIS burst ramp-down, $\hat{y}_{\{h,BT,ru,rd\}}$ is the matched filter generated from y using the stated signal parameters, and $\star$ denotes cross-correlation. Theoretically, SNR is a function of $\mathrm{E}\,(x \star y)$, so we apply that function to derive our estimate $\widehat{\Gamma(y)}$. We assume each physical transmitter with MMSI m produces a distinct distribution, and we analyze the resulting mixed Gaussian distribution by sampling:

$$H_m(y) = \arg\max_h \left(\max_{\{\mathrm{h,BT,ru,rd}\}} \left(\hat{y}_{\{h,BT,ru,rd\}} \star y\right)\right). \tag{6}$$

Using standard statistical methods, we estimate for each m the expected value and variance for each component of the mixture. For any m, it is possible the corresponding mixture is unimodal or has fewer modes than the cardinality of the mixture. In [9], the author gestures towards methods making it possible to determine from historical AIS data the cardinality of the mixture model, and these same techniques combined with RF burst collection can help to indicate precisely which submixtures contribute to single modes in the resulting distribution. Appealing to Figure 5, we project it may be difficult to separate many of the underlying distributions, but we see plenty of outliers such that for some values of m the distribution here described may provide quite a lot of leverage. The resulting expected values become $\{h_i | i \in \mathcal{I} \backslash s\}$. For each y, we then determine $i(y) \in \mathcal{I} \backslash s$ such that y is most likely to have come from the distribution with expected value $h_{i(y)}$. We can then determine

$$C_m(y, i(y)) = \max_{\{\mathrm{BT,ru,rd}\}} \left(\hat{y}_{\{h_{i(y)},BT,ru,rd\}} \star y\right). \tag{7}$$

Finally, we normalize this value to some reference SNR to obtain $\overline{C_m(y, i(y))}$, exploiting the relationship between $\mathrm{E}\,(x \star y)$ and $\mathrm{SNR}\,(y)$. For each $i \in \mathcal{I} \setminus s$, we calculate $\mathrm{var}\left(\overline{\mathcal{C}_{m,i}}\right)$ where

$$\overline{\mathcal{C}_{m,i}} \coloneqq \left\{\overline{C_m(y, i(y))} | i(y) = i\right\}, \tag{8}$$

and send these values to the spacecraft.

On orbit, for each burst y, we calculate an new estimate $\Gamma(y)$ for $\mathrm{SNR}\,(y)$ by analyzing cluster variance during the demodulation stage. To review, $\mathrm{SNR}\,(y)$ is the SNR of received signal y, and $\widehat{\Gamma(y)}$ is the estimate for $\mathrm{SNR}\,(y)$ derived from the correlation value between y and a matched filter for y. The estimate $\Gamma(y)$ is distinct from $\widehat{\Gamma(y)}$ and is independent of matched-filter correlation. Given $\Gamma(y)$, we can determine a normalized expected value for correlation: $\overline{C(y)}$ Using enhanced metadata processing, we calculate c_i for each $i \in \mathcal{I} \setminus s$, and we normalize that value with respect to $\Gamma(y)$ to derive $\overline{c_i}$. We then find the deviation: $d_i(y) = \overline{C(y)} - \overline{c_i}$. Since $\Gamma(y)$ is independent of matched-filter correlation, we may use $d_i(y)$ for hypothesis-testing against the distributions described by measurements $\mathrm{var}\left(\overline{\mathcal{C}_{m,i}}\right)$. Setting a threshold in terms of variance, we trigger if $d_i(y)$ is above the threshold for each $i \in \mathcal{I} \setminus s$.

Of course, compute overhead for the second layer of targeted burst collection is essentially identical to enhanced metadata processing. In order to guard against insufficient processing overhead, we change the position of the process monitoring compute overhead within our system: rather than serving as an arbiter between normal and enhanced metadata processing, we position

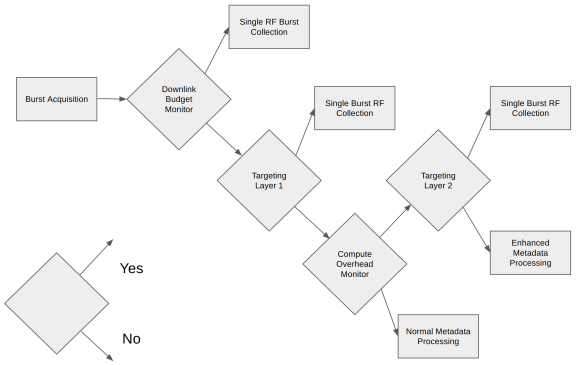

Fig. 7. Payload Processor Logical Flow

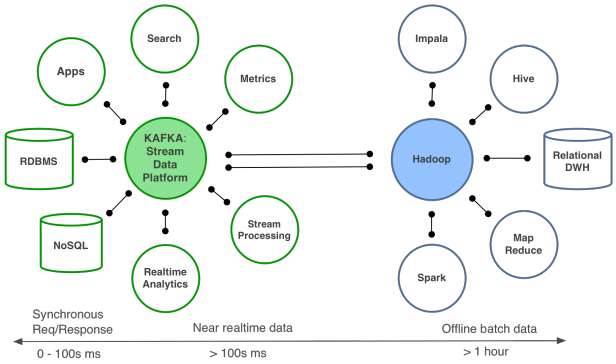

Fig. 8. Stream Processing Versus Data Lake Processing. ([13])

it to serve as an arbiter between the second layer of targeting for processing between targeted RF collection and normal metadata processing. In final summation, logical flow for operation on the payload processor should follow the logic in Figure 7.

Ground Software Deployment

The system described in Subsection IV-B places three general demands on the ground processing segment. First, it must use data downlinked from the payload processor to detect AIS spoofing with as little latency as possible. Improved geolocation accuracy resulting from estimating TOA and FOA with tighter than ordinary variance makes it possible to determine a spoofer's location of origin and to project with higher likelihood future locations, but advantages gained through painstaking effort throughout the processing chain perish quickly. Those same projections grow increasingly less accurate with the passage of time. Second, the system must host a variety of persistent processes requiring differently sourced and differently structured data. We refer here to externally sourced data feeding MMSI blacklisting, behavior recognition, and anomaly detection processes running to support a first layer of targeting for burst RF collection. These same processes or analogous ones help to provide context and amplify value for AIS spoofing detection on the ground. Some processes related to training for machine learning and DSP demand appreciable, sometimes distributed and heterogeneous compute resources, and managing those resources in production becomes a serious issue. Third, we expect to have to ingest some of these data sources in real time, continuously.

To meet these demands, HawkEye searches for value in software built or supported by organizations facing challenges of a similar nature, often at scales many orders of magnitude greater than those we anticipate HawkEye to face. This strategy depends in large part on those organizations releasing their products to open source communities as part of a larger strategy to share the costs of development and maintenance, and the past decade has seen a trend among high-valuation companies, employing vast numbers of capable developers, providing business- and Big-Data-oriented software under the Apache License, the GNU Public License, the MIT License, and other, similar terms. We highlight a handful of these packages indicating their applicability to our processing situation.

In order to support low-latency analysis of our own data at the point of ingestion, we use a stream processing approach built from the MongoDB and the Apache projects Kafka and Storm. In Figure 8, we reproduce a graphic from [13] to summarize the role of stream processing within the larger context of data analysis, focusing on expectations in terms of latency. Kafka manages data movement using a publish/subscribe model optimized for throughput and inherently capable of distribution, fault tolerance and parallelism. Each data ingestor in the HawkEye architecture, including lightweight processes to handle data downlinked in batches from base station passes, inherits from a Kafka producer class and publishes to a topic, making its data stream generally available to any other networked process in the HawkEye architecture. Every networked process may subscribe to one or many Kafka topics for real-time data input or perform lookups in MongoDB for access to static data. Storm serves as the platform for stream processing, coordinating compute resources across servers to exploit parallelism in the set of processing we know we must perform on every atom of data collected. Storm spouts source data by inheriting from a Kafka consumer class and subscribing to the topic(s) providing data to processing units (bolts) acting downstream of the spout.

Spouts and bolts have parallelism such that a parallelized spout can become part of a consumer group working in unison to consume data from partitions maintained for a single topic in the Kafka broker, enabling faster data ingestion. MongoDB is a NoSQL store capable of grouping dissimilarly structured data representable as JSON objects into searchable collections. Thus far, data in the HawkEye universe usually is structured, but similar data from different sources may adhere to different structures. MongoDB sacrifices some structured efficiency of SQL databases for geospatial search such as PostGIS but relaxes some of the difficulty associated with data normalization.

In terms of latency demands, the software pipeline addresses needs in terms of data movement (both between the data ingestors and the Storm spouts through Kafka and then within Storm itself) and scalable compute resources delivered to meet that data. Downlinked data, in this case data related to individual AIS bursts, breaks apart easily into individually digestible units. For the most part, even expensive DSP and geolocation processes can happen in parallel such as to take advantage of easy-to-parameterize parallelism in a Storm topology. Time to process an entire downlink batch becomes a simple function of the number of actors we choose to deploy within Storm and (in extreme cases) the number of Kafka partitions used for the topic. Whether within Storm or not, the assortment of processes running on the ground have persistent, centralized access both to real-time ingested data and to the output of other processes through Kafka, so what might otherwise become a fully-connected web of directed network connections between the deployed processes, each one requiring care and feeding, reduces to thoughtful deployment of a single communications hub with bidirectional spokes reaching out to services as necessary. NoSQL data storage smooths wrinkles caused by differently structured data or data that changes structure over time such as by updates to REST APIs from which we stream data such as MMSI watch lists. Finally, fault tolerance among distributed deployments for each of Kafka, Storm and MongoDB allow for guaranteed persistence and reliable real-time streaming. It's important to note infrastructure does not, in and of itself, provide data analytics. On the other hand, among the problems we face doing data analytics (and evidently among the problems large Big Data companies face), infrastructure and deployment make up enough of the practical challenges to warrant care and scrutiny in the process of tool selection and integration.

FUTURE DIRECTIONS: PREDICTIVE ANALYTICS AND RF DATA PROCESSING

At a shallow level, we discuss three directions for processing downlinked data likely to change the effectiveness of the system discussed within this paper and other aspects of engineering at HawkEye. With our discussion of h, we broach the topic of emitter identification for AIS collection, but even for AIS processing in isolation, the problem can become more complicated, interesting and rewarding with the addition of more data and more computation. Here we bring to light some of these complications. As a separate topic, we largely ignore in previous sections the plasticity of identifying parameters such as h. It may be true that the statistical distribution for h estimated from a single transmitter has everything to do with details of production runs for physical components used to build that transmitter, or it may be that the parameters also depend on the situation. What is the temperature at the time of transmission? What is the humidity? Finally, we have largely avoided a topic of obvious interest to us, maritime positional predictive analytics, in part to better focus on the more scoped topic of TOA and FOA estimation. We mention the problem here through the lens of the paper's more specific areas of interest.

In Subsection III, we describe estimating h from I/Q data as a joint estimation process involving the variables h, TOA, and FOA, and we mention the addition of the parameter h influences the solution space in ill-behaved ways. In Equation 5, we cheat a bit further toward the same destination: here we see a joint estimation process involving h, BT, ru, rd, TOA, and FOA. Generally, we would expect the addition of more variables to the estimation to require increasingly informed approaches to maximizations and other optimizations related to those variables, and we would expect the compute requirements to rise sharply. We face these problems as we push our emitter identification capabilities to be more and more reliable. We posit differentiating transmitters by h will separate transmitters statistically, and Figure 5 suggests we are both correct and incorrect. The parameter h provides valuable information, but a richer space of higher dimensions may provide greater desirable separability. Although the eyes water somewhat at the data requirements, it is possible to imagine building an RF database complete enough to investigate unsupervised learning techniques such that we relieve ourselves of having to select features providing maximal separability among transmitters and rely instead on neural nets and clustering algorithms to process raw RF bursts and learn better sets of features than the ones we propose.

We should have the capability to investigate emitter identification parameter plasticity much sooner. Once on-orbit, a relatively modest RF database in combination with live geospatial weather feeds will allow us to examine situational signal parameter estimation with some precision. With this information, we could produce feedback to on-orbit collection making TOA and FOA estimation still more accurate. The table of descriptors for the mixture of distributions of values for h among different MMSIs could grow to include functions making those descriptors a function of predictable weather phenomena. By providing weather projections to the satellite during base station passes, we would provide the payload processor with tools to leverage weather-specific values for putative signal parameters as input to the matched-filter generation process. At the risk of wild speculation, there is room for feedback to flow in the other direction. If we can positively identify an emitter and estimate signal parameters such as h, then we can match our estimates with expectations given certain weather conditions, essentially reading subtle deviations in h from expectations like a thermometer.

Like the rest of the world, HawkEye sits at the receiving-end of nearly-live-updating information about vessel locations derived from both satellite and terrestrial collectors. Unlike the rest of the world, once in orbit, HawkEye will have access to shared views of RF emissions captured from vessels at multiple receivers in space with different ephemera. With this exposure to data, HawkEye should have a good vantage point on predicting near-term vessel movements in considerable detail. Leveraging such a capability to perform anomaly detection stands out as an obvious application, but we can use the same capability to inform payload processing as it undertakes to target RF burst collection and expensive on-orbit estimation processes. At the very least, ground processing could provide time-windowed white lists of MMSIs expected in the RF footprint for AIS collection between base station passes providing statistically-backed evidence to targeting mechanisms in space concerned with, say, avoidance of collection for vessels behaving ordinarily.

Creating a high-quality data feed of unique information given a program of smallsat engineering can be rewarding and even game-changing, though rarely is it easy. It's much easier to take such a data feed and apply analysis before passing the results to the next processor down the line, never having to consider the data's final destination. Focusing on instances where HawkEye is its own data analytics customer forces a kind of economy in considering how derived data may be used to make processes better and serves as a healthy reminder to employ empathy when delivering data to others.

References

[1] https://www.marinetraffic.com.

[2] http://iuu-vessels.org/iuu.

[3] Recommendation ITU-R M.1371-4 technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band.

[4] Why the 3v's are not sufficient to describe big data. https://datafloq.com/read/3vs-sufficient-describe-big-data/166. Accessed: 2017-06-05.

[5] International shipping facts and figures: Information resources on trade, safety, security, envrironment. http://www.imo.org/en/KnowledgeCentre/ShipsAndShippingFactsAndFigures/TheRoleandImportanceofInternationalShipping/Documents/International%20Shipping%20-%20Facts%20and%20Figures.pdf, March 2012. Publisher: IMO, Maritime Knowledge Center.

[6] Dr. Marco Balduzzi. Ais exposed: Understanding vulnerabilities & attacks 2.0. https://www.blackhat.com/docs/asia-14/materials/Balduzzi/Asia-14-Balduzzi-AIS-Exposed-Understanding-Vulnerabilities-And-Attacks.pdf, 2014. Accessed: 2017-06-06.

[7] J. Block. A self-deploying and self-stabilizing helical antenna for small satellites. may 2013.

[8] Peggy Browning. Spoofing ais: The debate continues. http://blog.exactearth.com/blog/bid/339822/Spoofing-AIS-The-Debate-Continues, 2014. Accessed: 2017-06-06.

[9] Kimbra Cutlip. Spoofing: One identity shared by multiple vessels. http://blog.globalfishingwatch.org/2016/07/spoofing-one-identity-shared-by-multiple-vessels, 2016. Accessed: 2017-06-06.

[10] Kimbra Cutlip. What does an ais message look like anyway. http://blog.globalfishingwatch.org/2016/11/what-does-an-ais-message-look-like-anyway, 2016. Accessed: 2017-06-07.

[11] Maria Angeles Jurado Gallardo and Ghislain Ruy. *FM Discriminator for AIS Satellite Detection*, pages 19–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[12] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics: The Approach Based on Influence Functions (Wiley Series in Probability and Statistics)*. Wiley, 1986.

[13] Derrick Harris. Why linkedin is important to the future of the internet of things. http://fortune.com/2015/09/02/linkedin-kafka-internet-of-things/, September 2015.

[14] Peter J. Huber. *Robust Statistics (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 1981.

[15] Kgabo Frans Mathapo. A software defined radio ais for the za-002 satellite. In *Proceedings Small Satellite Conference*, number V-3 in SSC06, 2006.

[16] Annalee Newitz. It took less than a minute of satellite time to catch these thieves red-handed. https://arstechnica.com/tech-policy/2017/02/to-catch-a-thief-with-satellite-data/, 2017. Accessed: 2017-0.

[17] S. Stein. Algorithms for ambiguity function processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(3):588–599, jun 1981.