Neptune: An Automated System For Dark Ship Detection, Targeting, And Prioritization

Adam Byerly, Will Zhang, Sesan Iwarere Ph.D., Waseem Malik Ph.D., Sheldon Bish Ph.D., Musad Haque Ph.D., Tamim Sookoor Ph.D.
Johns Hopkins Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel MD 20723

ABSTRACT

Dark ship detection at open ocean scale drives the need for enhanced space-based intelligence, surveillance, and reconnaissance capabilities. With the boom of commercial space-based sensing, an automated process is needed to meet the growing volume and velocity of data. Aggregation and fusion of multi-modal data from the variety of existing and proposed space-based sensor networks can be leveraged to produce target quality tracks on ships. These sensor modalities include SAR (Synthetic Aperture Radar), EO/IR (Electro-optical / Infrared), and AIS (Automatic Information System). In this paper we demonstrate the ability to perform automated target recognition of surface vessels from these modalities on a space-flight processor to simulate on-orbit detection. These detections are fused to form quality tracks which can then be used for anomaly detection of dark ships via pattern of life. Tracks formed over continental or global scale motivates the need for further automated analysis. A significant amount of human effort would be needed to analyze thousands or tens of thousands of tracks in detail and in real-time. We developed a suite of pattern-of-life tools that extracts features from tracks and flags tracks that deviate too far from some learned definition of normality.

Keywords: Dark Ships, Automated Target Recognition, Sensor Planning, Pattern of Life

1. Introduction

1.1 Dark Ships

Vessels engaging in illicit activity, such as selling oil to regimes in violation of international sanctions [2], smuggling drugs and arms [3], and engaging in illegal fishing [4], routinely attempt to evade detection. A critical challenge to national security is to quickly and reliably locate and identify these ships anywhere in the world. The Automated Identification System (AIS) is a radio frequency (RF) system for identifying and locating maritime vessels that relies on ships being fitted with transceivers and broadcasting information such as their identity, position, course, and speed. These messages can be received by other vessels, as well as AIS base stations on the coast and on satellites. International law requires all voyaging ships over 300 gross tons to participate in this system [1]. Yet, ships attempting to hide their identities, or locations, can spoof the information of other vessels or simply turn off their transponders. Further, any vessel under 300 gross tons is not required to participate in AIS, giving rise to the possibility of nefarious actors leveraging smaller craft for their activities while appearing

benign. In addition to ships going dark in terms of AIS transmissions, they could also turn off other sources of RF emission through emission control (EMCON) in order to go radio silent. Military vessels, for instance, engage in this activity by switching over to commercial radar or previously unseen war reserve waveforms in order to hide from sensors monitoring military waveforms. Therefore, there is a need to expand the definition of dark ships, from simply being AIS dark, to all ships attempting to evade detection or engage in nefarious activity, and develop a capability to generate target-quality data on ships of interest by establishing and maintaining accurate tracks and ship identification.

The problem of locating and identifying dark ships and developing quality tracking data on them has a number of constraints. First, to generate targeting solutions and enable blue interdiction, location data with sufficient precision and confidence will need to be processed, exploited, and disseminated rapidly. Second, to target these ships anywhere in the world, global coverage will be required. Third, to detect nefarious vessels, whether they be transmitting AIS or not, automated processing must be enhanced to “see-through” the evasion, and identify the true high-value bad-actors, the “needles in the needle stack.” Finally, any solution to this problem has to be easily fielded by various stakeholders. These include analysts and operators tracking vessels in the field, commanders planning missions at headquarters, and captains navigating blue vessels against red adversaries in the open oceans.

1.2 Existing Capabilities

There are currently a number of commercial and non-profit organizations attempting to solve the dark ship problem. For instance, Global Fishing Watch [5] is a partnership among Google, Oceana, and SkyTruth to “provide the world’s first global view of commercial fishing activities” in order to reduce “global overfishing, illegal fishing, and habitat destruction.” The goal of the project is to increase transparency on fishing activity through an online map that provides tracks of fishing vessels from January 2012 through three days prior to the present time. The system uses satellite-based optical, synthetic aperture radar (SAR), and visible infrared imaging radiometer suit (VIIRS) sensors to supplement AIS and detect dark fishing fleets. While this solution addresses a number of constraints of the problem by providing global coverage through an easy-to-use interface, shedding light on illegal fishing activity in aggregate, it does not solve the problem of enabling the real-time targeting of dark ships. Its interface also just provides the locations and tracks of all vessels and does not flag any activity as being illegal or a ship as being dark leaving it up to a human analyst to derive these conclusions.

Another solution is Hawkeye 360’s [6] approach of instrumenting satellites with RF sensors to monitor widely used communications channels which can give away the position of ships that have turned off, or do not have, AIS transponders. ICEYE [7] and Spire [8] extend this approach by augmenting the satellite-based radios with synthetic-aperture radar to detect ships engaging in EMCON activity, so that a ship that goes dark in the RF spectrum but is still visible to SAR can be detected. Planet [32], in a similar approach, provides a data feed for vessel detection that locates ships seen from their satellite optical imagery, regardless of the vessel’s AIS broadcasting status. Again, these systems work well to locate vessels not broadcasting AIS, but still fail to

adequately sift through the trove of vessel detections to identify actual bad-actors. In particular, Planet's vessel detection feed identifies users that can "monitor [ships'] patterns of life and anomalies," but they fail to develop such a capability themselves.

1.3 Neptune System

Similar to the approaches of Global Fishing Watch [5], Spire [7], ICEYE [8], and Planet [32], we attempt to detect dark ships through multi-modal sensor fusion. While the above approaches are focused on a subset of modalities which a determined adversary could attempt to evade, we are building a platform, Neptune, through which all available sensor modalities, including optical, synthetic-aperture radar, and electronic intelligence, can be fused, making it harder for a dark ship to evade. Additionally, a pattern-of-life system is developed to prioritize anomalous vessel behavior, drawing attention to high-value dark targets sooner. Finally, to optimize the quality of sensor fusion, sensor resources are directed to the highest priority ships using a game theory assignment algorithm. Figure 1 describes this closed loop activity facilitated by the CLCSim scenario simulator which includes the track fusion engine. CLCSim is a software package that is used to create high fidelity representations of sensor-target interactions and enables the integration of sensor planning as part of the closed loop collaborative situational awareness process. The track information generated in CLCSim is passed to our pattern-of-life system to generate relevancy scores for each track; these relevancy scores rank the tracks as more (or less) anomalous (described in detail in Section 4), and bring operator attention to more anomalous vessels faster. These relevancy scores are then leveraged by a game-theoretic planner (described in detail in Section 3) to optimally perform sensor-to-target assignment.

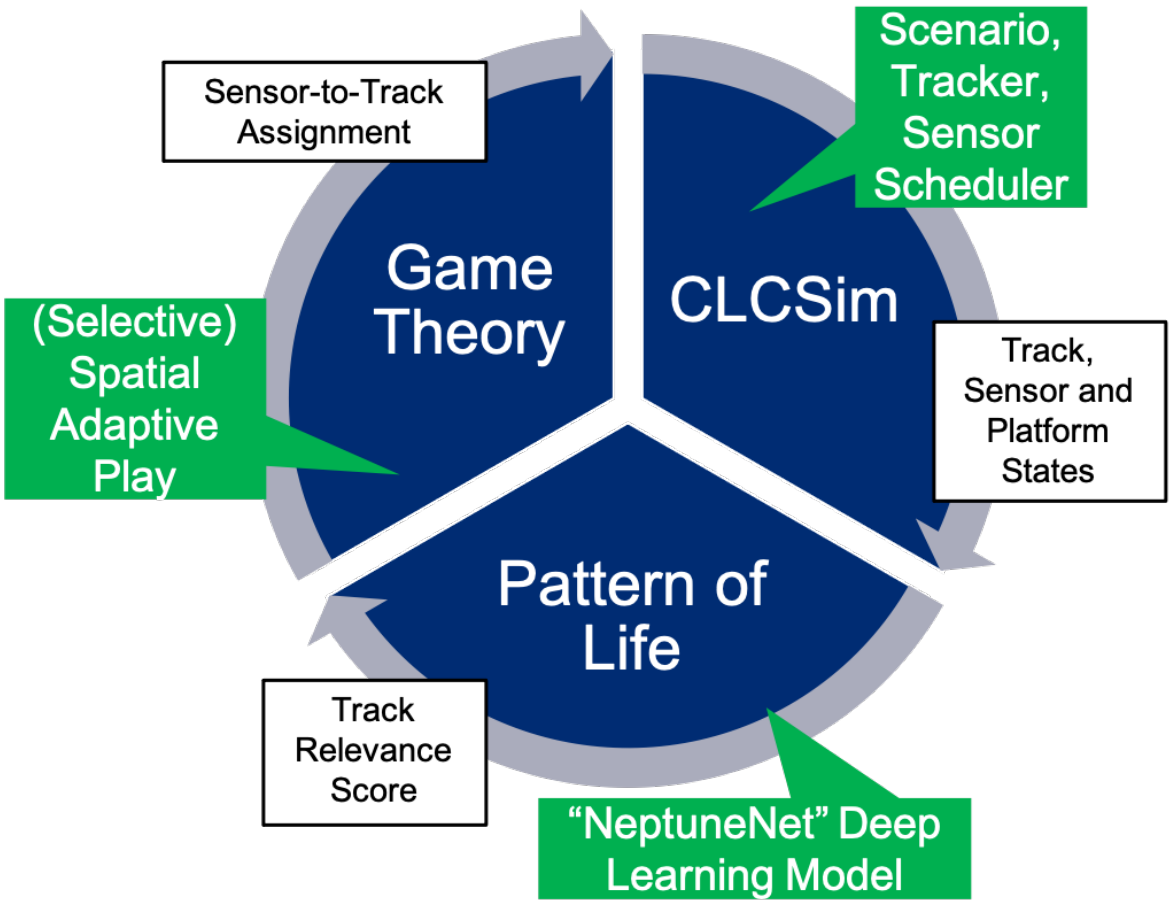


Figure 1. *Overall view of Neptune system. Vessel detections are fused, and vessel tracks are maintained, within CLCSim, which in turn passes state information to the pattern-of-life component for vessel analysis and track prioritization. This prioritization is used by our game theory planner to coordinate and assign sensor coverage.*

A main enabler of the Neptune system, is the idea that more powerful, low size, weight, and power (SWaP), processors, such as the NVIDIA Jetson TX2i, will be flown in space, opening up the possibility of pushing advanced processing to the spacecraft. Capella Space [16], a commercial space-based SAR company, has already demonstrated spaceflight of a Jetson processor in space [17], and several research universities [18, 19] have near-term plans of their own to fly Jetson boards. As more commercial space companies begin flying spacecraft with commercial off-the-shelf (COTS) accelerators, and processor manufacturers (NVIDIA, Xilinx, etc.) continue to improve processing capability, we can fly more sophisticated algorithms, allowing us to push our detection capabilities to the sensing edge. This push of detection capability to the sensing edge allows for reduced exploitation time and enhanced on-board decision making.

With the capability to detect vessels on-board a spacecraft, we can then investigate tipping and cueing between satellites to ensure that track custody of ships of interest are maintained, even when satellites go out of theater. This conjunction of on-board target recognition, with sensor fusion and planning, will allow the Neptune system to more rapidly locate vessels, and maintain custody of said vessels. We rely on the multiple hypothesis tracker (MHT) capability of the Closed-Loop Collaborative ISR

Simulation Test Bed (CLCSim) [20, 21] to manage pseudo-tracks of vessels between sensor detects.

Neptune goes a step beyond current solutions by attempting to perform automated analysis and identify nefarious dark ships based on their behavior. While existing projects, such as Global Fishing Watch, rely on machine learning algorithms that have been hand-tuned by teams of analysts to identify certain dark ship behavior such as Chinese fishing fleets illegally fishing in North Korea's Exclusive Economic Zone (EEZ), we are attempting to train algorithms that can discriminate anomalous activity without any predetermined patterns. This will result in a more automated solution that is applicable to the needs of a number of sponsors. Our approach for behavior-based dark ship detection relies on the development of algorithms to identify Patterns of Life (PoL) which capture normal vessel behavior in various regions across the world. From this background of normal behavior, anomaly detectors can flag abnormal behavior for further scrutiny by an end user providing an easy interface through which to interact with the system without the overwhelm that is possible through interfaces presenting all available data, such as the one provided by Global Fishing Watch.

1.4 Remainder of Paper

The remainder of this paper is organized as follows: Section 2 further develops the motivation and algorithms Neptune uses for on-board target recognition. Section 3 describes the sensor and data fusion work, as well as an example demonstration description. Section 4 covers the pattern-of-life analysis for prioritizing dark ship behavior and providing contextual information to a human operator. Section 5 concludes this paper with a summary and future path to adoption.

2. On-Orbit Automatic Vessel Detection

2.1 Space-Based Observation

Earth orbiting missions are increasingly designed around large-scale constellations of small satellites with greater heterogenous sensor capability, data volume, and complexity of joint operations. In parallel, sensors continue to be constructed to consume less size, weight, and power (SWaP) while generating larger volumes of multifaceted data, and the cost to orbit has decreased drastically with the advent of the commercial space revolution [31]. In addition to this increase in on-orbit data generation, bandwidth limitations and ground station contact schedules often restricts analysis until several hours after an observation has occurred, driving a need for smarter on-orbit processing in the hunt for dark ships.

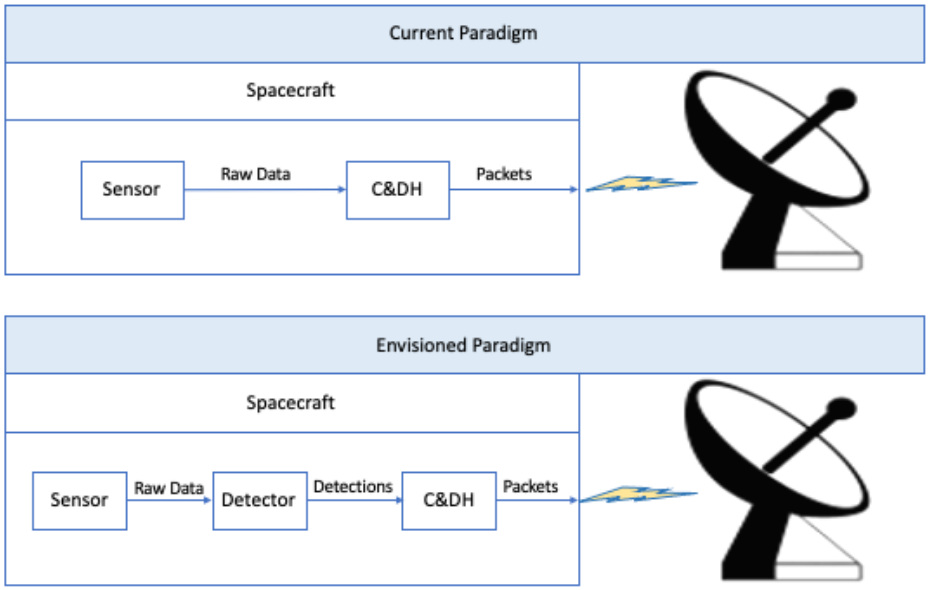


Figure 2. *High-Level Downlink Data Flow. In a normal system, observations are captured via an on-board sensor which typically passes raw data to the spacecraft Command & Data Handling (C&DH) system. The C&DH system formats these measurements into packets for downlinking to the ground. With on-board detection, these raw measurements would instead first pass through the detector to generate detections, and those detections would be passed to the C&DH system for packaging and transmission.*

An increase in number of data sources, e.g., via commercial space, also leads to a large influx of data, most of which will be empty ocean. Moving automated detection to the tactical edge, e.g., on-board the spacecraft, enables autonomous processing of multiple data streams, and decreases needed bandwidth. Such processing can aid in the faster exploitation and dissemination of time-critical information needed to execute real-time targeting. We envision a paradigm where the spacecraft "tells" you when it sees something of interest, as seen in Figure 2.

2.2 Deep-learning Based Detection

Deep learning has been demonstrated to address the object detection and classification problem [9, 10]. In particular, much interest has been given to applying deep learning to detecting surface ships [11]. In Wang et. al. [12], a dataset of calibrated SAR images from two space platforms (Sentinel and Gaofen) was collated and annotated with bounding boxes around surface vessels. This dataset consisted of images from multiple operating modes / resolutions, included coastal and open-ocean regions, and contains scenes with single and multiple vessels. Wang et. al. also demonstrated the capabilities of several deep learning detectors, achieving state-of-the-art performance (mean-average precision) as compared to alternative techniques, e.g., constant false-alarm rate. While the goal of [12] was to consider the detection accuracy of these models, the question still remains as to the timeliness of processing and exploitation.

To address the operational challenges of pushing detection models to the edge, namely low SWaP constraints and limited processing resources (as compared to ground-based processing), we must consider the trade-off of detection accuracy with inference time. That is, if a model performs well in detecting all vessels in an

observation, but fails to generate these detections in a timely manner, those detections may no longer be informative. Likewise, if a model can generate detections instantaneously but fails to identify most vessels in an observation, there may be too many missed targets for operational relevance.

To address this trade-off, we consider the ResNet family [13] of convolutional neural networks (CNN). The ResNet family of CNNs has been shown to be performant in image processing tasks [13], and includes a variety of models with increasing depth. Deeper models in the family, e.g., ResNet101, achieve greater accuracy than shallower models, e.g., ResNet34, at the trade-off of inference time; deeper models take longer to execute. By using the ResNet family, we can evaluate several models of varying depth, but identical architecture, on our target hardware. With these ResNet models as our feature extraction backbone, we then consider the RetinaNet detector [10], a state-of-the-art deep learning detector. The RetinaNet detector is both fast and accurate at identifying and localizing objects of interest in input images, giving us the best of both worlds: fast detection speed and accurate detections.

2.3 Training Details and Experimental Results

We evaluated the ResNet18, ResNet 34, and ResNet50 models on an Ubuntu 18.04 machine with a 4GB Nvidia GTX 970 for training, and an Nvidia Jetson Nano and Xavier for inference benchmarking. A PyTorch implementation [15] of RetinaNet was trained on the Wang et. al. dataset as described in [12], with a batch size of 8, using each of the ResNet backbones. The results of this training are shown in Table 1.

Table 1. *Vessel detection mean average precision (mAP) [14], and inference time.*

Model	mAP	Inference Time (Jetson Nano)	Inference Time (Jetson Xavier)
RetinaNet-ResNet18	0.778	(3620 $pixel)^2$ / sec	(12541 $pixel)^2$ / sec
RetinaNet-ResNet34	0.803	(2560 $pixel)^2$ / sec	(8868 $pixel)^2$ / sec
RetinaNet-ResNet50	0.807	(2486 $pixel)^2$ / sec	(8613 $pixel)^2$ / sec
RetinaNet-VGG16 (from Wang et. al. [12])	0.914	(1252 $pixel)^2$ / sec	(4337 $pixel)^2$ / sec

All three models achieve comparable validation mAP scores, with the deeper ResNet50 model scoring the best of the three. However, all three models do fall short of the reported mAP score for the RetinaNet-VGG16 model evaluated in Wang et. Al. [12]. While these models underperform the VGG16-based model in terms of detection capability, we must also consider the inference time, i.e., the time needed to execute these models on representative low-SWaP processors. In the third and fourth column of Table 1, we report the image area capable of being processed by each model on the Jetson Nano and Jetson Xavier boards; to account for differences in image sample distance and swatch size, we normalize to square pixel area. It is clear that the ResNet family base of models significantly outperform the VGG16 based models, with the

ResNet18 based detector capable of processing nearly nine times the area per second of the VGG16 model. For context, a representative Capella SAR spotlight image covers 5 km by 5 km with a resolution of 0.5 m, resulting in an image with 10000 by 10000 pixels. From our analysis, the VGG16 model would require 5.3 seconds to process a single image on a Jetson Xavier, while the ResNet18 model would require only 0.64 seconds and the ResNet50 model only 1.35 seconds.

3. Sensor Fusion

The dark ship detection problem contains the challenges of 1) persistent localization and 2) identification of anomalous behavior. To obtain the best chances of persistently localizing a dark ship is to maintain track custody on all detected ships, whether AIS broadcasting or not based on sensor modalities that do not rely on such broadcasting, like an imaging sensor. For dark ships, the best way to localize is to persistently maintain track custody on such ships from the time that they are either 1) in close proximity to a broadcasting ship (and thus detected via an imaging sensor), or 2) gave their final AIS broadcast. Fusion of different sensor modalities has long been exploited, particularly to produce quality tracks of maritime vessels. These sensor modalities may include SAR (Synthetic Aperture Radar), EO (Electro-optical) and IR (Infrared), and AIS (Automatic Information System) and TDOA (Time Difference of Arrival). For Neptune, the use of EO/IR and SAR imaging modalities bring high resolution and all-weather ship imaging capabilities to bear for the dark ship detection problem. Here, we fuse multi-sensor data over time to establish track histories and maintain track custodies of a large number of maritime vessels, particularly those that have gone dark. To accomplish this goal, we leverage CLCSim (Closed Loop Collaborative ISR (Intelligence, Surveillance, and Reconnaissance) Simulation) as a high-fidelity scenario simulation testbed on which the Neptune dark ship detection and tracking system is being developed.

CLCSim is a C++ software simulation platform for developing, testing and analyzing closed-loop collaborative ISR, sensor data fusion, estimation, control, and optimization algorithms across maritime, sea, air and space domains. Embedded within CLCSim is the Precision Engagement of Moving Targets (PEMT) tracker which oversees the filtering and multiple hypothesis tracking (MHT) processes that manage track states. PEMT is an unclassified flexible MHT software package that is capable of handling measurements from multiple sources and source configurations. For more information on CLCSim, please refer to [20] and [21].

CLCSim provides representations of sensors, targets, and the track fusion engine as well as facilities for sensor resource management. The sensor resource management facility of CLCSim enables us to ‘plug-in’ an arbitrary sensor planner that meets specific needs. Here, CLCSim passes track data to Neptune’s pattern-of-life module for anomaly detection of dark ships, then we integrate a Game Theoretic Framework to establish tipping and cueing between satellite sensors, based on the designation of suspected ships which will allow for enhanced situational awareness of dark ships. We will also be using the existing tipping and cueing algorithms that are built in to CLCSim, in particular a greedy algorithm (see the CLCSim Sensor Planner section below).

We have demonstrated the Neptune sensor planning strategy in a CLCSim simulation. . Our input to CLCSim were 57 modified AIS ship trajectories in the South China Sea. These ship targets were then tracked by EO/IR satellite sensors whose parameters were modeled after typical commercial satellite specifications (Table 2) from companies such as Planet or Capella. These parameters dictate the nature of the measurements that are processed by the tracker and influence track quality.

Table 2. *Demo input parameters*

Field	Value
Update Rate	1 Hz
Orientation	Yaw = 0, pitch = -90, roll = 0
FOR (Field of Regard) angle	130 degrees
Probability of Detection (P_D)	0.999
False Alarm Rate (P_{fa})	1e-8 per unit of detection volume (m^3)
Angular Resolution	0.00014 degrees

The output of our demo was then target quality tracks on these ship targets. The track information required for the relevant dark ship analysis is shown in Table 3 below.

Table 3. *Demo output parameters*

Field	Description
Name	Full Ship Name (i.e. Ship_1)
LineageID	The trackers name for a TrackID
Time	Timestamp in seconds
PositionX	ECEF (Earth Centered Earth-Fixed) X coordinate
PositionY	ECEF Y coordinate
PositionZ	ECEF Z coordinate
PositionLAT	Latitude in degrees
PositionLON	Longitude in degrees
PositionALT	Altitude in meters
Course	Degrees of heading from north
VelocityMAG	Magnitude of velocity

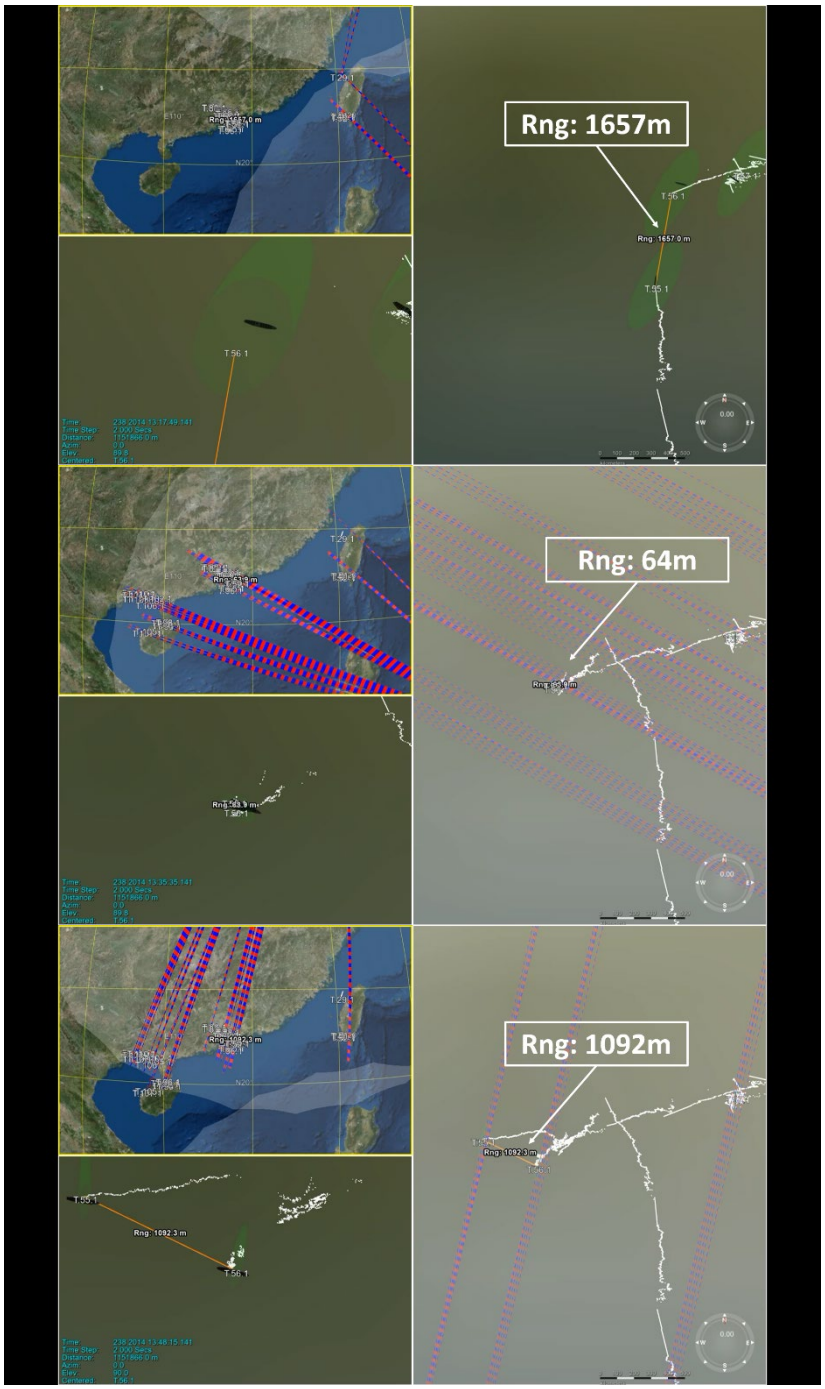
A more detailed description of our demonstration is now described. In the simulation demonstration, the baseline performance of track initialization, maintenance of track custody and track volume were assessed for a constellation mimicking that of Capella, so that hypothetical performance could be assessed with high fidelity. Using CLCSim, a constellation of 36 satellite sensor platforms spanning 3 LEO (low Earth orbit) orbits of 12 satellites each were used to track 57 ships on the South China Sea. Ship measurements used to feed the simulation came from available reported AIS data. One of the primary goals of the simulation was to determine whether the CLCISR (CLCSim-PEMT) software tools could maintain track custody among ships that come into close encounters with each other, as may happen with suspicious activity. The sensor platforms utilized an ‘AngleSensorModel’ measurement type where the measurement state was composed of only two values, horizontal angle and a vertical angle. The scan update rate was set to 1Hz, to represent a hypothetical EO/IR sensor,

and all ships within the sensor's field of regard (FOR) were accepted into the detection gate for purposes of assessing tracker performance and generating track histories for pattern-of-life algorithm development. In the next simulation update a more realistic, narrow field of view will be asserted for detection gating paired with the PoL and Game Theory modules providing sensor planning (aimpoint scheduling) functionality.

A demo video was recorded of the tracker's performance via SIMDIS, a simulation display toolset [33]. This video illustrated successful maintenance of track custody between two ships that come into a close encounter, slow to a possible stop, then depart from one another as a possible example of suspicious activity. The video collage (see Figure 3) consists of a macro view of the overall South China sea (Figure 3a), which enables the visualization of the sensor field of view (FOV) as it sweeps across the scene and its relation to the location of tracks on the surface. A closeup view of a ship bound to Track 56 can be seen (Figure 3b), and a static, slightly zoomed out view of the local scene where the interaction of ships bound to tracks 56 and 55 can also be seen (Figure 3c). The green ellipsoids represent the track estimates, centered on the track position and sized by the semimajor/minor axes of the covariance ellipsoid. The white traces represent the history of track estimates (where the tracks have been) and make it easier to see how separate track paths relate to each other.

Initially, track 56 comes in from the east, while track 55 can be seen entering from the south. They cross paths at one point after which sensor coverage briefly drops. Upon the arrival of the next sensor pass, the tracks re-associate correctly and continue updating. The tracks slowly converge as they decrease in speed, up until they seem to stop completely at a spacing that is barely discernable (less than 50 meters – see Figure 3b). After this stop (for approximately 1 or 2 minutes), they proceed away from each other with 56 continuing west and 55 changing course to the south/south east, potentially returning to its original location (see Figure 3c).

This demo shows the capability of the PEMT tracker to maintain track custody on targets that interact very closely and with tracks that converge in state values. If the sensors were previously cued onto these ships, then they would have no problem maintaining them even if they were to go dark. The horizontal and vertical angle measurement uncertainties were set to 50m, which may or may not be enough to account for uncertainty in sensor orientation, and position. Uncertainties that result from ownship errors and image resolution are currently being explored. Incorporating these errors would enable the simulation to more closely estimate the performance of dark ship detection and tracking of dark ships using spaceborne sensors/platforms.



(a)
Track 55 and 56 begin to converge paths

(b)
Tracks 55 and 56 nearly converge and slow to a crawl less than 50m apart

(c)
Tracks 55 and 56 diverge and separate with Track 55 continuing west and Track 56 turning around and proceeding in the opposite direction

Figure 3: Sensor Fusion Demo Video Screenshots

3.1 Sensor Resource Management in CLCSim

CLCISR performs Sensor Resource Management (SRM) functions within the ESRM (Extensible Sensor Resource Management) module, which is built as a dependency to CLCSim. Sensor Resource Management pertains to the task of managing sensor resources such that the quality of sensor measurements is

maximized, especially when the number of target clusters to monitor exceed the number of sensor resources available. The interface between CLCSim and ESRM is well characterized, thus the integration of the Neptune sensor planner module in place of ESRM is straightforward. The interface provides the following information to a sensor planner module as shown in figure 4:

- Track Information (kinematic states (position, velocity), classification, covariance, score, probability)
- Platform Information (Kinematic states, orientation)
- Sensor Information (orientation, FOV, FOR, measurement uncertainty, P_D, P_{fa})
- Sensor Planner Configuration (cost function threshold, event horizon period, sensor update period)

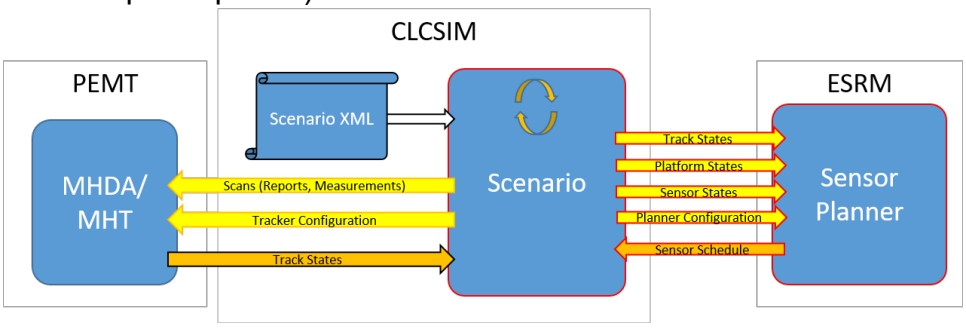


Figure 4. *Data Flow for Sensor Resource Management.*

Track, Platform and Sensor information are needed to provide the sensor planner with situational awareness needed to calculate a cost function for the planner to optimize against (i.e. find the 'best' pairing between sensors and tracks given constraints). Cost functions are determined based on the objectives of the planner. Often, it is preferable to have a planning model that minimizes the global uncertainty of track states, in which case it would be beneficial to cue sensors to tracks of highest state uncertainty. In other cases, it would be preferable to invest sensor resources in more crowded target regions with lower track hypothesis scores (less certain track custody). Sensor Planners are user configurable from the CLCSim scenario configuration. Once the Sensor plan is made, it is packaged as a Sensor Schedule containing tipping updates/instructions for the sensors.

A potential scenario that would leverage SRM, would simulate AIS measurements from a new sensor model that mimics AIS positional data from a maritime vessel. CLCSim's sensor models have parameters to toggle visibility of a platform (ship) that we could leverage to simulate a ship going 'dark'. This visibility parameter would turn on for the EO/IR/SAR platforms once they go dark which require the sensor planner to cue the field of view (FOV) to the locations of the dark ship's (fading) track to maintain custody.

The concept of tipping and cueing in the context of our problem refers to sharing track information about a particular target or set of targets among several sensors and then determining a coordinated sensor scheduling action on that target or set of targets. The Neptune sensor planner accepts track data from CLCSim, leverages that track data to determine suspicious ship activity with pattern-of-life and leverages a game theory algorithm to allow for coordination between satellite sensors.

3.2 Sensor Coordination

While the PoL algorithms provide a way to judge the value of a target, there still remains a need to coordinate Neptune's collection of sensors platforms. Effective coordination and cooperation amongst the platforms (and even between sensors on a platform) can take advantage of the information being generated by the PoL component of Neptune. Furthermore, if the coordination can be orchestrated in a decentralized manner, the reliance on a central planner is relieved, resulting in a network of sensors that is not rendered useless by the failure of that single, critically important sensor platform. There are numerous coordination strategies available in the multi-agent/multi-robot literature. Game theory is an established framework for such strategic coordination [22]. This type of coordination is different from the usual kinematic coordination, such as formation control, which is prevalent in the multi-agent community. A typical game consists of a set of players, a set of actions each player can take, and individual utilities. The players (or agents) in Neptune are the individual satellites in the constellation. Similar to a typical satellite, each player is assumed to have limited communication, information, and implementation requirements. A global objective is identified from the system-planner point of view. Players are rational agents, interested only in maximizing their own individual utilities. Despite players behaving in a self-interested manner, game theory provides useful convergence guarantees. Before we can delve into that, we will need to briefly work through some notations.

There are quite a few convergence concepts in game theory, but the most well-known is the Nash equilibrium. Agents in a Nash equilibrium are in a configuration such that there is "no unilateral incentive to deviate" [22]. In other words, an action profile is a Nash equilibrium if asked one by one, no agent would be better off by changing its action, given what everyone else is doing. This feature of agents agreeing to an arrangement of actions is a desirable feature considering the type of multi-agent system that is being proposed to tackle the problem of dark ship detection: autonomous and consisting of selfish agents acting in a decentralized manner.

This equilibrium may or may not be optimal – that depends on how we design the individual utilities. There is a notion of "alignment" between the global objective and individual utilities and Arslan et al. [22] describe in detail how to pick individual utilities such that a global objective is maximized. The importance of individual utilities is described in that work with the help of a simple example. Two autonomous vehicles are considered, both trying to assign two targets between themselves. There is a high-value target (value of 10) and a low-valued target (value of 2). The players (player set $\boldsymbol{P} = \{V_1, V_2\}$) choose a target to engage, which signifies their action. The agents can also choose to do nothing (T_0 in the matrix below, Figure 5). The matrix form shown below allows for a simple representation when a game consists of two players and there are a finite set of actions. For the game on the left (see Figure 5), the payoffs to each agent are split evenly when they both engage the same target. This utility model is referred to as the "Equally-Shared" utility. For the game on the right (see Figure 5), individual agents are implementing the "Wonderful Life" utility model. In the Wonderful Life utility, the payoff to a player is the marginal contribution of that player's engagement with a target. Arslan et al. [22] describe in detail how the choice of individual utility functions affects the multi-agent system's ability to maximize a global objective. For the game on the left (see Figure 5), when both vehicles choose to engage the high-valued target, we

are in the (T_H, T_H) cell of the matrix and the payoffs to each player is 5 (first number in the cell goes to the first vehicle, the second number to the other vehicle). The circles indicate the "best response" of a given player. For example, the best thing for V_1 to do when V_2 does nothing is to engage the high-value target. Hence the 10 in $(10, 0)$ is circled. The Nash equilibrium for this game is both players engaging the high-value target, which yields a value of 5 for each player and a combined value of 10, if we consider the global objective to be the sum of the target values that are covered. The tweak from the Equally-Shared utility model to the Wonderful Life utility model leads to two Nash equilibria, both maximizing the global objective.

V_1 \ V_2	T_0	T_L	T_H
T_0	0, 0	0, 2	0, 10
T_L	2, 0	1, 1	2, 10
T_H	10, 0	10, 2	5, 5

V_1 \ V_2	T_0	T_L	T_H
T_0	0, 0	0, 2	0, 10
T_L	2, 0	0, 0	2, 10
T_H	10, 0	10, 2	0, 0

Figure 5. *Two unmanned vehicles, V_1 and V_2 are playing a game. Each vehicle can select an action from the set $\{T_0, T_L, T_H\}$. T_L refers to a vehicle choosing to engage the low-valued target; T_H refers to the high-valued target; and T_0 refers to neither. The game on the left uses Equally-Shared utility versus the one on the right, which uses the Wonderful Life utility. Each of the Nash equilibria for the game on the right also maximizes the global objective. This example was presented by Arslan et al. [22] and tying back to the problem of dark ship detection, a vehicle can be thought of as a stand in for a sensor platform.*

The remaining piece to design for Neptune is the way in which players choose actions. So far, we've reasoned through what a Nash equilibrium is, but how do we arrive there? It turns out, for a certain class of games, known as a *potential game* [23], there exists negotiation mechanisms (also known as "learning algorithms" in the game-theoretic literature) that can be used by an agent to pick its actions. In a potential game, an improvement in any agent's individual utility that takes place from changing its action corresponds to an equal improvement in some global potential function that is not specific to any agent. In traffic settings, congestion serves as the potential function, for instance. There are numerous algorithms in the multi-agent game-theoretic literature and the one we implemented first is the Spatial Adaptive Play (SAP) [22, 29]. Compared to other learning algorithms, such as Regret Monitoring [22, 30], which require agents to maintain their own history of actions, SAP operates on the most recent set of assignments. This reduced burden on the amount of information required by an agent to maintain throughout the course of a game makes SAP an attractive choice for Neptune. A detailed description of SAP is provided in Appendix A. One additional advantage of SAP is that with high probability, players are guaranteed to converge to a Nash equilibrium – one that maximizes a global objective that is given by the sum of the target values, given that the Wonderful Life utility is employed.

Agents maintain beliefs and can take actions (based on the CLCSim environment). The key design choices are the individual (or local) utility functions and the negotiation mechanisms. We assign local utility functions in a way that allows agents to act rationally while maximizing global utility. We then pick negotiation

mechanisms over local interactions that lead to agreeable action profiles. In particular, actions on ships that we consider dark targets. A significant benefit of this game theory approach for tipping and cueing is that it can deliver convergence guarantees (with high probability) to max or near-optimal global utility values, whereas such guarantees cannot be given by heuristics like greedy algorithms. That being said, we must note that, there may be cases where it is not advantageous to use this game theory approach. In some cases, a greedy algorithm approach may be the right decision for sensor selection.

4. Pattern of Life

The final piece of Neptune assigns priority values to ships based on their location and kinematic history using an anomaly detection technique. The Neptune Pattern-of-Life (PoL) bridges the gap between the on-orbit detection and sensor resource management of Neptune components.

Orbit-based ship detection is a direct response to dark ships that hide their location. However, darkness and ill-intentions are not equivalent. While darkness is arguably a big red flag, faulty equipment, poor reception and signal interference can result in accidental darkness. Another consideration is that bad actors can operate without going dark or while still complying to regulation. For example, crewmembers aboard the MSC Gayane were caught smuggling narcotics in 2019 under the cover of a legitimate voyage that transmitted AIS. Drug smugglers may also use small vessels that are not required to transmit AIS. The main point is that while a detection capability does respond to the dark ship problem, there is additional benefit if an algorithm can use these detections to flag anomalous ship behavior, whether from a dark ship or not.

A capability that tags ships with anomaly scores is immediately applicable to the sensor management strategies described in the previous section. The scores can be used to influence satellite coordination and to direct targeting. As previously mentioned, several commercial space companies, e.g., Planet, have provided data streams to which anomaly detection can be applied; however, no such company has yet to offer such anomaly detection as part of their product suite. If we consider the preceding statement to reflect the general state of commercial industry, then our work advances the state-of-the-art.

4.1 Variational Autoencoders

Neptune PoL is currently based on running ship tracks (detection time-series for a single ship) through a recurrent variational autoencoders (VAE), an unsupervised deep learning sequence model. The recurrent VAE technique has its roots in Bayesian statistics and variational inference [24, 25]. A core theme in Bayesian statistics is the idea that it is useful to define plausible probability models that emulate the real-world processes that generated a dataset. The dataset is then used to infer distributions over latent variables, quantities that have meaning with respect to the model but are not part of the measured dataset. A common concern with Bayesian methods is that key distributions are often analytically intractable. Traditional approaches such as variational inference or Markov-chain Monte-Carlo (MCMC) sampling parameterize approximations of these distributions. [24, 25] demonstrate that neural networks can be configured to do

the same. The claim is that neural networks, as flexible nonlinear function approximators, may arrive at a more accurate approximation. Appendix B presents equations key to our PoL approach.

In terms of ships tracks, the Neptune PoL model learns in theory how to encode and decode latent track representations. Distributional constraints [24, 25] are softly-imposed on the latent representation while requiring it to retain enough information to reconstruct the original track. A perfect reconstruction means that decoding a latent track representation will produce a track that is identical to the original track. This can be quantified using measures such as pointwise cross-entropy or any other reasonable metric. Tracks that the model struggles to encode and decode can be considered anomalies as the model should have learned to encode and decode a wide variety of "normal" tracks during training.

4.2 Training Details and Experimental Results

Tracks were extracted from data sourced from MarineTraffic [34], an organization that curates AIS. To reflect the goal of modelling normal kinematics, the only tracks considered were those that had moved 10 nautical miles from their original position, had reported AIS for at least 90 minutes, and had transmitted at least once every 10 minutes. The experiment also focused on tracks in the South China Sea. Under these conditions 2423 tracks were available for training on March 15, 2019 and 2974 tracks were available for validation on March 16, 2019. The validation set was then augmented with 881 randomly-generated synthetic loiter tracks as anomalous behavior targets. The synthetic ships were then scattered in locations of normal traffic. All tracks were linearly-resampled to a 5-minute sample rate. Empirically, Neptune PoL struggles to faithfully-reconstruct tracks (section 4.1) however the amount of reconstruction error still seems to be discriminative in nature.

Post-training, the PoL model was evaluated against 2 discriminative tasks using the median reconstruction error. The first task was to discriminate between the training and validation set. This task is a sanity check for the degree of model overfitting. Ideally the training and validation distributions over reconstruction error are identical. The second task was to discriminate between the validation tracks and synthetic loiter tracks. Figure 6 presents ROC curves associated with these tasks as well as a number of example tracks from the extended validation set for one of the trafficked regions in the South China Sea,

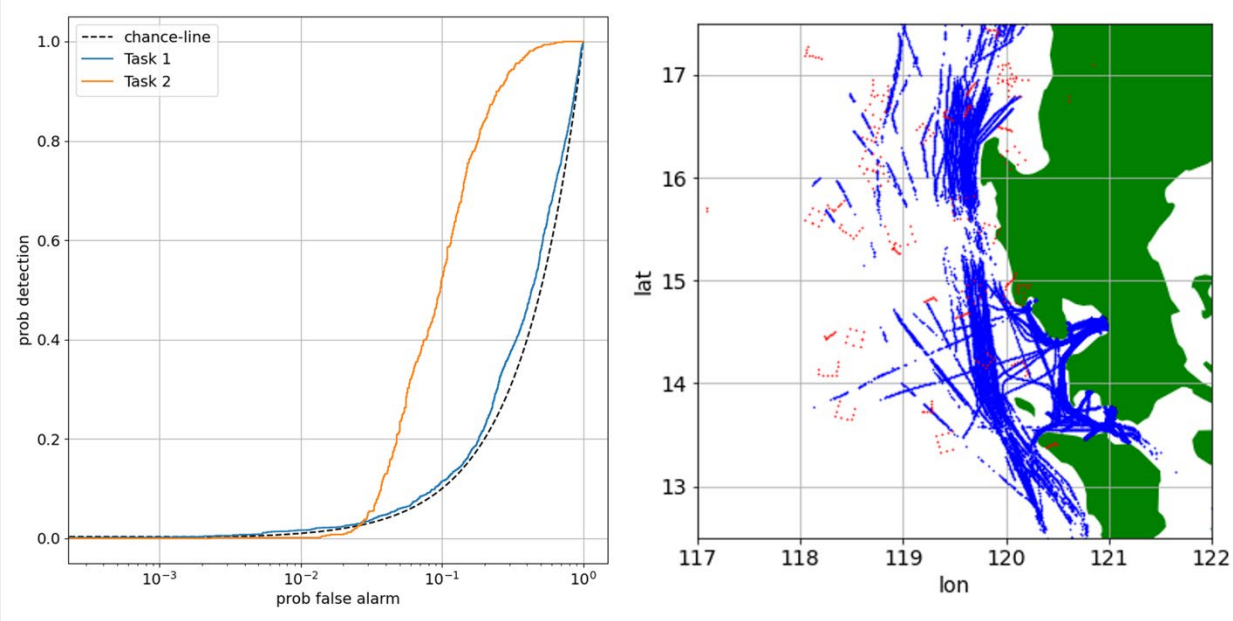


Figure 6. *Loiter track flagging*

5. Sensor Planning Demonstration

5.1 Demo Description

The demonstration scenario simulates 25 abnormal ship routes among 228 real ship routes as the ground truth. The abnormal ship routes are represented as synthetic loiterers with route patterns resembling a boxlike shape and real ship routes represented with actual AIS data.

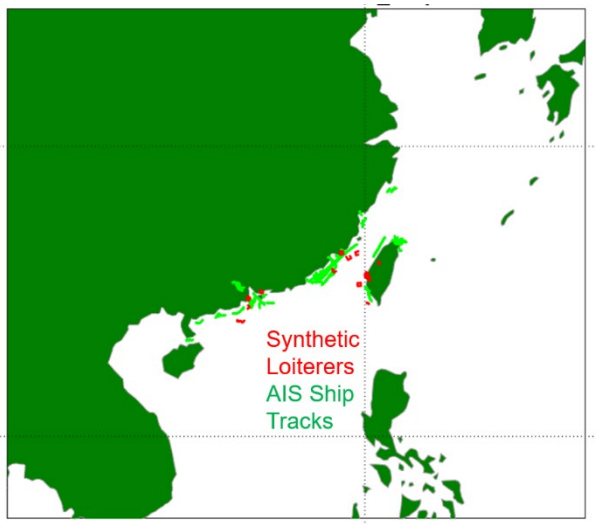


Figure 7. *Ship route laydown of demonstration Scenario*

Both the synthetic loiterers and the AIS ship data was interpolated in time for a 5 minute update rate on the ship coordinates. The placement of the synthetic loiterers were such that they are mostly co-located among the AIS ship data, in order to challenge the multiple hypothesis tracker. In addition, the close proximity between the synthetic

loiterers and AIS ship data enable sensors to obtain a fix on the (dark) synthetic loiterers serendipitously. As stated in section 4, we seek a sensor planning algorithm that can use detections to flag anomalous ship behavior, whether from a dark ship or not. To test the algorithm, the simulation stops all ships from beaconing one minute into the simulation, effectively making them all go “dark”. At this stage, the effectiveness of PoL and GT in sensor planning and scheduling can be measured by the ability to distinguish between loiterer and AIS ships. Once this distinction is established, it can be exploited to enable preferential assignment of sensors to the loiterers.

5.2 Sensor Configuration

The representative sensors used in this demo are the same Capella Space polar orbiting sensors as those from the Sensor Fusion demo. The sensor update rate was set to 2 minutes, which approximates the prospective onboard SAR image formation and ATR latency time (SAR chosen as a more challenging modality in terms of latency). The track update rate was set to 5 minutes, which matches the data rate that the PoL model was trained on. To use another rate for track updates would risk introducing unmodeled features to the data that could interfere with the scoring. Once the AIS detections cease, tracks are only updated on a ship if that ship enters a sensor’s field of view.

Track histories generated are fed into the PoL data buffer to undergo model prediction. The track history data is auto encoded and the encoded features are compared to the feature distribution of known ‘normal’ ship routes. The more closely the input track histories are to the distribution of ‘normal’ ship routes, the lower the ‘PoL Score’: a measure of dissimilarity to commonly traveled route features. In this sense, the PoL model should produce a higher score for synthetic loiterer ships than ‘normal’ AIS beaconing ships.

5.3 Results

For the 4 hour scenario of 228 AIS derived ship routes and 25 synthetic loiterer routes, the PoL scores are shown in Figure 8. Across the scenario, the normal ship scores are relatively consistent, and are consistently lower than those of the synthetic loitering ships. The mean and median score metrics are representative of the entire set of normal and synthetic loitering ships at each time point. The mean and median values of PoL scores stay consistent after the AIS detections are removed, which suggests that 1) the tracker maintains custody of both normal and synthetic loitering ships and 2) The encoded track history features remain relatively consistent even with a reduced rate of detections (no more than 3 sensors have LOS to any given surface coordinate).

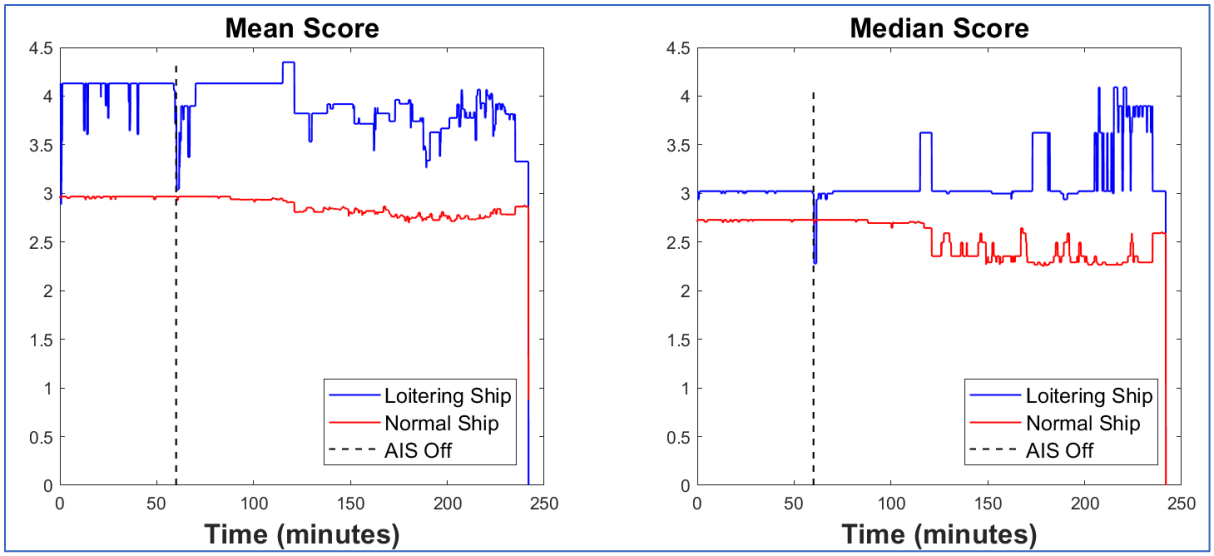


Figure 8: PoL Score Results for Loitering vs. Normal Ships in Sensor Planner Demo (Left: Spatial Adaptive Play Assignment, Right: Selective Spatial Adaptive Play Assignment)

6. Conclusions

The down-selection and filtering of raw data, from on-sensor data returns, to fused target tracks, to pattern-of-life contextual labeling, gives Neptune the opportunity to quickly and reliably locate and identify dark ships anywhere in the world. By relying on APL's position as a trusted agent for the government, Neptune has the ability to enable our sponsors to augment national tactical means with the large number of commercial satellites that are being launched by a number of companies in order to achieve situational awareness across the world's oceans. It will be impossible for the government to own sufficient assets to achieve the global coverage necessary to target dark ships anywhere in the world. Therefore, Neptune's approach of leveraging commercial capabilities, in conjunction with sensor fusion and pattern-of-life analytics, will allow our sponsors to solve this critical challenge. Success in this domain potentially motivates applications of Neptune to other domains such as pedestrian or ground transportation ATR and behavior modeling.

From the prototype Neptune currently in place, there are several key next steps we wish to address. First, the on-orbit target recognition work has thus far been conducted with a processor-in-the-loop test bench, but the actual models analyzed have yet to be flown, either in space or an appropriate surrogate, e.g., high altitude balloon. Demonstrating a full flight article will ensure the entire on-board ATR approach is not only feasible, as we have begun to show here, but also efficient and usable. A second step will be to incorporate a wider variety of sensor modalities, beyond the AIS transmissions and SAR imagery we have used to date, e.g., electronics intelligence. Finally, the demonstration of the pattern of life analysis in detecting an example known dark ship will be necessary to demonstrate the full capability of the PoL analysis.

Citations

1. International Maritime Organization. (1974). *SOLAS, 1974: International Convention for the Safety of Life at Sea, 1974*. London: International Maritime Organization.
2. McFarlane, S., & Faucon, B. (2017, July 6). *Ships Exporting Iranian Oil Go Dark, Raising Sanctions Red Flags*. The Wall Street Journal. https://www.wsj.com/articles/ships-exporting-iranian-oil-go-dark-raising-sanctions-red-flags-1499333402.
3. *MSC Gayane Crew Member Pleads Guilty to Cocaine Trafficking Stemming from One of the Largest Drug Seizures in U.S. History*. The United States Department of Justice. (2020, June 19). https://www.justice.gov/usao-edpa/pr/msc-gayane-crew-member-pleads-guilty-cocaine-trafficking-stemming-one-largest-drug.
4. Pike, J. (n.d.). *Military*. China Named in Ambitious New Anti-Illegal Fishing Strategy for the U.S. Coast Guard. https://www.globalsecurity.org/military/library/news/2020/09/mil-200918-rfa01.htm.
5. https://globalfishingwatch.org/
6. https://www.he360.com/
7. https://www.iceye.com/
8. https://spire.com/
9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386
10. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, & Piotr Dollár. (2018). Focal Loss for Dense Object Detection.
11. Zhou, Yiqing & Cai, Zemin & Zhu, Yuting & Yan, Jingwen. (2020). Automatic ship detection in SAR Image based on Multi-scale Faster R-CNN. Journal of Physics: Conference Series. 1550. 042006. 10.1088/1742-6596/1550/4/042006.
12. Wang, Y., Wang, C., Zhang, H., Dong, Y., & Wei, S. (2019). A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote Sensing, 11(7), 765.*
13. Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. (2015). Deep Residual Learning for Image Recognition.
14. Zhu, Mu. (2004). Recall, Precision and Average Precision.
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S.. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.
16. https://www.capellaspace.com/
17. Federal Communications Commission. (2019). Appendix A: Response to Question 7 of FCC Form 442 (Purpose of Experiment). Retrieved from https://apps.fcc.gov/els/GetAtt.html?id=247843&x=.
18. https://sites.utexas.edu/tsl/serpent/
19. Adams, C., Spain, A., Parker, J., Hevert, M. Roach, J., & Cotton, D. (2019). Towards an Integrated GPU Accelerated SoC as a Flight Computer for Small Satellites. *2019 IEEE Aerospace Conference.* https://doi.org/10.1109/aero.2019.8741765

20. Jon DeSena, “Closed-Loop Collaborative ISR Simulation Test Bed (CLCSIM) Overview – EP1 Tool Quad Day,” January 8, 2020 (Power Point Presentation).

21. Andrew J. Newman and Jonathan T. DeSena, “Closed-Loop Collaborative Intelligence, Surveillance, and Reconnaissance Resource Management,” in Johns Hopkins, APL Technical Digest, Volume 3, Number 3, pp. 183-214, 2013.

22. Gürdal Arslan , Jason R. Marden , Jeff S. Shamma, “Autonomous vehicle-target assignment: a game theoretical formulation,” Journal of Dynamic Systems, Measurement and Control, vo. 129 (5), pp. 584-596, 2007. https://doi.org/10.1115/1.2766722

23. D. Monderer and L.S. Shapley, “Potential Games,” Games Econ. Behav., 14, pp. 124–143, 1996.

24. Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

25. Chung, Junyoung, et al. "A recurrent latent variable model for sequential data." arXiv preprint arXiv:1506.02216 (2015).

26. Park, Jaeyoon, et al. "Illuminating dark fishing fleets in North Korea." Science advances 6.30 (2020): eabb1197.

27. Nguyen, Duong, et al. "GeoTrackNet--A Maritime Anomaly Detector Using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection." IEEE Transactions on Intelligent Transportation Systems (2021).

28. Bishop, Christopher M. *Pattern recognition and machine learning*. springer, 2006. Chapter 7.2.

29. Durieu, J. and Solal, P. *Adaptive play with spatial sampling*, Games and Economic Behavior, 43 (2), 2003, pp. 189-195.

30. Greenwald, A. and Jafari, A. *A General Class of No-Regret Learning Algorithms and Game-Theoretic Equilibria*, Learning Theory and Kernel Machines, Lecture Notes in Computer Science, vol 2777.

31. Jones, H. (2018). The recent large reduction in space launch cost. 48th International Conference on Environmental Systems.

32. https://www.planet.com

33. https://simdis.nrl.navy.mil/index.aspx

34. “Data Services (API).” *AIS API Service*, MarineTraffic, www.marinetraffic.com/en/ais-api-services.

35. Byerly, A., Sookoor, T., Iwarere, S., Zhang, W., Haque, M., Malik, W., Bish, S. “Neptune: An Automated System For Dark Ship Detection, Targeting, And Prioritization”. Johns Hopkins APL Technical Digest (Pre-publication)

Appendix A

In the Spatial Adaptive Play (SAP) algorithm described by Arslan et al. [22], each player probabilistically picks an action. At every iteration, the SAP algorithm can be used by a player to assign a probability to its available actions. Each player then selects an action based on the distribution resulting from SAP. Staying consistent with the original notation presented by the authors, the probability distribution over actions for player i is denoted as $p_i(k)$ at step k. If the available actions of player i (from the set $\boldsymbol{A}_i$) are listed as $\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{|\boldsymbol{A}_i|}$, where $|\boldsymbol{A}_i|$ is the cardinality of the set $\boldsymbol{A}_i$, then $p_i(k)$ is given by:

$$p_i(k) = \sigma\left(\frac{1}{\tau}\begin{bmatrix} U_{V_i}(\alpha_i^1,\ a_{-i}(k-1)) \\ U_{V_i}(\alpha_i^2,\ a_{-i}(k-1)) \\ \vdots \\ U_{V_i}(\alpha_i^{|\boldsymbol{A}_i|},\ a_{-i}(k-1)) \end{bmatrix}\right)$$

where, $\tau > 0$ is a small constant and $\sigma(\cdot)$ is the soft-max[1] or logit function. A vector is maintained by each player. Each entry in the vector is the utility a player would receive and is derived from playing a particular action against the most recent action taken by the other players (i.e., step $k-1$). This vector is passed through a soft-max function to constrain the entries to values between 0 and 1 and produce a distribution, which is then used to probabilistically pick an action at step.

Appendix B

Variational deep learning combines the theory behind variational inference and the function approximation qualities of neural networks. Ultimately, this results in a technique that is similar to variational inference but does not need to rely on extra assumptions. The core concept in variational inference is that the marginal distribution with respect to x can be broken down into the sum of a quantity termed the evidence lower bound (ELBO) and the KL divergence between a distribution q and the posterior p.

$$\log p(x;\theta) = log\frac{p(x,z;\theta)}{q(z|x,\theta;\phi)} + log\frac{q(z|x,\theta;\phi)}{p(z|x;\theta)} = E_{q(z|x,\theta;\,\phi)}[\log p(x;\ \theta)]$$

$$E_{q(z|x,\theta;\,\phi)}[\log p(x;\ \theta)] = E_{q(z|x,\theta;\phi)}\left[log\frac{p(x,z;\theta)}{q(z|x,\theta;\phi)}\right] + KL(q(z|x,\theta;\phi)||p(z|x;\theta))$$

The marginal distribution is constant with respect to q(.) parameters phi. Thus, the update rule below indirectly minimizes the KL divergence between q and p.

$$\phi^* = argmax_\phi\ E_{q(z|x,\theta;\phi)}\left[log\frac{p(x,z;\theta)}{q(z|x,\theta;\phi)}\right]$$

[1] The soft-max function takes a vector $x = [x_1, \dots, x_n]$ and produces a vector whose i^{th} entry is $e^{x_i}/(e^{x_1} + \cdots + e^{x_n})$

Kingma and Welling demonstrated in [24] that a reformulation of ELBO is compatible with stochastic gradient methods such as those used to train neural networks.

$$E_{q(z|x,\theta;\phi)}\left[log\frac{p(x,z;\theta)}{q(z|x,\theta;\phi)}\right] = E_{q(z|x,\theta;\phi)}[\log p(x|z;\theta)] + E_{q(z|x,\theta;\phi)}[\log\frac{p(z;\theta)}{q(z|x,\theta;\phi)}]$$
$$= E_{q(z|x,\theta;\phi)}[\log p(x|z;\theta)] - KL(p(z;\theta)||q(z|x,\theta;\phi))$$
$$\approx \frac{1}{N}\sum_{n}\log p(x_n|z_n;\theta) - KL(p(z;\theta)||q(z|x,\theta;\phi))$$

This formulation is compatible with neural networks. An encoder network computes $q(z|x,\theta;\phi)$ given measurements x. A decoder network then then estimates $E_{q(z|x,\theta;\phi)}[\log p(x|z;\theta)]$ from samples drawn from $q(z|x,\theta;\phi)$. The encoder and decoder are then optimized against the reformulated ELBO objective above. Networks constructed and trained in this way are referred to as variational autoencoders (VAE).

Literature [25] extended the VAE-ELBO formulation in [24] to support sequential data as well, by conditioning on previous values in the sequence. This final formulation allows VAEs to be used with recurrent neural network architectures. The hybrid architectures are appropriately called variational recurrent neural networks (VRNN).

$$E_{q(z|x,\theta;\phi)}\left[log\frac{p(x_t,z_t|h_{t-1};\theta)}{q(z_t|x_t,h_{t-1},\theta;\phi)}\right]$$
$$\approx \frac{1}{N}\sum_{n}\log p(x_{t,n}|z_{t,n},h_{t-1,n};\theta) + KL(p(z_t;\theta)||q(z_t|x_t,\theta;\phi))$$
$$where\ h_{t,n} = f(h_{t-1,n},x_t,z_t)$$

The dependency on previous measurements and latent variables is implemented through a state variable h. This technique is common and well-known in recurrent neural network literature. Furthermore, this construction is necessarily non-Markovian as the value of $h_{t,n}$ is influenced by all previous x and z values.