Sea-cret Agents: Maritime Abduction for Region Generation to Expose Dark Vessel Trajectories

Divyagna Bavikadi
Arizona State Universty
Tempe, USA
dbavikad@asu.edu

Nathaniel Lee
Arizona State Universty
Tempe, USA
nlee51@asu.edu

Paulo Shakarian
Arizona State Universty
Tempe, USA
pshakari@asu.edu

Chad Parvis
EpochGeo
Washington, DC, USA
cp@epochgeo.com

ABSTRACT

Bad actors in the maritime industry engage in illegal behaviors after disabling their vessel's automatic identification system (AIS) - which makes finding such vessels difficult for analysts. Machine learning approaches only succeed in identifying the locations of these "dark vessels" in the immediate future. This work leverages ideas from the literature on abductive inference applied to locating adversarial agents to solve the problem. Specifically, we combine concepts from abduction, logic programming, and rule learning to create an efficient method that approaches full recall of dark vessels while requiring less search area than machine learning methods. We provide a logic-based paradigm for reasoning about maritime vessels, an abductive inference query method, an automatically extracted rule-based behavior model methodology, and a thorough suite of experiments.

KEYWORDS

Trajectory Analysis, Abduction, Maritime forecasting

ACM Reference Format:
Divyagna Bavikadi, Nathaniel Lee, Paulo Shakarian, and Chad Parvis. 2025. Sea-cret Agents: Maritime Abduction for Region Generation to Expose Dark Vessel Trajectories. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

1 INTRODUCTION

Maritime vessels are equipped with the automatic identification system (AIS) to track their position on the globe [26]. However, malicious actors often disable this system - becoming "dark" when conducting illegal activities. Understanding these "dark vessel" has implications for security [27], maritime analysis [20, 32], planning [3], and forecasting [44]. Recently, with the support of the U.S. Treasury and European Union in enforcing maritime services prohibitions for seaborne Russian oil [40], industry efforts have been made in real-world scenarios including piracy, illegal fishing, human trafficking, border protection, and sanction violations [38, 41] highlighting the increasing need for efficient dark vessel detection. Recent machine learning (ML) approaches are limited to trajectory prediction with a time horizon of less than an hour [13, 18, 23, 24] or rely on satellite data susceptible to weather conditions [1, 7, 12]. Other approaches require expert intervention using radio frequency doppler shift [38]. These approaches are not data-efficient and cannot explain why they determined a given result. We note that from a practical perspective, the limited forward-prediction value of ML approaches is significant - as the ability to find the dark vessel locations degrades with increased search area and resources. Meanwhile, recent work on generating faux trajectories for human movement suggests that abductive inference can address some of these difficulties [5] - although that work does not predict real trajectories and was not applied to the maritime domain. In this paper, we combine ideas from abductive inference, logic programming, and rule learning to identify locations of dark vessels based on partial trajectory information. We show that we are able to approach full recall of dark vessel trajectories requiring less than half of the area coverage required by our machine learning baselines. Further, we found that the recall performance of the abduction-based approach *increases* with search area and resources - unlike the degradation experienced with ML. We also demonstrate data efficiency, efficient inference calculations, and describe our ongoing efforts to deploy this technology in an operational platform. After a review of background material (Section 2) we make the following contributions:

(1) We provide a formalism for reasoning about maritime vessels including a logical language for expressing maritime vessel trajectories (Section 3.1) and the framing of an abduction problem (Section 3.2) that include a top-k approximation that we explore empirically in this paper.
(2) We provide a simple but effective rule-learning approach to agent behavior modeling (Section 3.3) that enables not only allows for data-driven (and data-efficient) abduction but also affords explainability of the results.
(3) We provide a suite of experimental results (Section 4) that demonstrate how the abduction approach is area-efficient by saturating with 157% higher recall than baselines for an area of $30km^2$, provides long-term predictions where ML methods fail, and provides improved performance of 476% in recall with additional resources.
(4) We also show that the approach is efficient in both terms of runtime and data as it can be instantiated with very little data - even a single training trajectory (providing comparable

Preprint - accepted at AAMAS 2025

performance of 0.62 precision to the use of all historical data - where we show ML catastrophically fails), as well as provide various ablation studies.

(5) We describe our efforts to deploy this system in an operational platform to support real-world analysts in dark vessel discovery (Section 5).

2 BACKGROUND

Dark Vessel Analytics. Maritime vessels employ deceptive shipping practices to benefit from violating international law, conducting illicit operations, violating environmental protections, and avoiding sanctions. In the 18th century, vessels disguised their Jolly Roger flags to deceive prospective victims before attacking them. Currently, vessels manipulate their AIS to avoid being monitored while performing illicit activities. On a monthly average, 800,000 dark activity events were detected in 2020 – 2022 [41]. Lately, in the aftermath of the Russia-Ukraine war, sanctions on maritime trade have evolved [40], and monthly dark activity rose by 216% [42]. More recently, in 2024, there has been a 340% rise in dark activity [25] much of which focused in the Black Sea (the area used in our experiments in Section 4). Such activities when gone undetected, can have realistic detrimental impacts on marine ecosystems, public safety, trade, and security. To monitor and control such behavior, efforts from government agencies [1, 40], and industry [7, 22, 38, 41], have invested in various efforts that began in earnest with the DARPA PANDA program over a decade ago [15]. These program have led to a line of research that we describe in the next subsection.

Related Work. Earlier work on maritime vessel trajectory prediction relied on Markov models [43, 45], extensions have also been applied to make efficient predictions. Although they work well for simple finite parameters, they are unable to capture complex patterns and this led to the later use of deep learning techniques for the problem - enabled by the availability of large datasets of maritime trajectories. To address the complexities of spatio-temporal interactions, [10] provides a sequence-to-sequence RNN to predict future maritime trajectories. Related work looks to predict a point ship location using a LSTM-transformer combination [16, 46]. These methods differ from the approach in this paper in that they only provide accurate predictions up to an hour in the future, require large amounts of training data, and do not afford explainability (so the analyst user cannot easily justify the dark vessel predictions to operational personnel). Maritime trajectory patterns have also been studied widely with for traffic management [19] with an unsupervised hierarchical method and safety [29] where they mine patterns to focus on shipping route characterization and anomaly detection. These methods are valuable for understanding typical and atypical trajectory behavior, but they are primarily focused on identifying patterns after the fact. In contrast, our method leverages trajectory behavior through abductive reasoning to infer an agent's future locations. We also note that this work differs from other maritime applications of AI such as vessel detection [28] where a model generates bounding boxes for the object vessel in an image or tracking it in a video [21]. This work also differs from a complementary line of work of patrolling strategies [4, 6, 9] that generates optimal patrol locations to cover a set of targets as we focus on generating locations to capture a target at a time horizon (as opposed to developing patrol plans for a non-adversarial agent).

Trajectory forecasting is a separate line of work, though this work is focused on short-time horizon prediction of human or robotic movement as opposed to the long-time horizon, global-scale prediction of maritime vessels. Some other notable approaches use deep learning architectures based on convolutional networks [23], adversarial methods [13], and autoencoders [24] as well as Marov chains [11, 14].

Abductive inference has provided a natural paradigm for locating unobserved adversarial agents - requiring much less data and providing more transparency than ML methods. Early work in this area offered simple models relating the adversary's point location to geospatial phenomenon [37]. Later work took a data-driven approach to learn a model of the adversarial behavior that enables abductive inference [31]. None of the aforementioned prior work on abduction involves trajectories nor does it involve making predictions of agent behavior over a long time horizon. Complementary to abduction work is the generation of spatial regions [8], which aims to maintain meaningful spatial boundaries for transportation services by partitioning an area of interest via region clustering (we employ similar techniques durin pre-processing). More recent work on abductive inference has been applied to human movement [5]. However, that work is designed to produce faux movement trajectories and not identify actual future agent locations. We note that it relies on a different approach (the use of A*) to create movement trajectories as opposed to this work that examines the problem as a top-k entailment query.

3 APPROACH

3.1 Logic for Maritime Agents

Logical Language. To define various aspects of the maritime domain environment, we use an annotated language [17, 34] with temporal semantics [2, 5, 36]. The language is defined with a set of constants that is partitioned into multiple domains ($\mathcal{D}_i \subset C$), one such subset, $\mathcal{D}_{loc}$, is a set of all potential locations of the vessel in a continuous space (called an "area of interest" or AOI) of dimensions $M \times N$. As usual in first-order logic, we define a corresponding set of variables ($\mathcal{V}$), and a set of predicate symbols ($\mathcal{P}$). Additional sets of constants include $\mathcal{D}_{\mathsf{r}}$ - which is the set of all regions within the AOI (and in practice, we will compute this based on historical trends ahead of time). When it is relevant, we shall subscript such constants with the upper-right and lower-left locations - e.g. $\mathsf{r}_{l1,l2} \in \mathcal{D}_{\mathsf{r}}$ is a region with upper-right corner $l1$ and lower-left corner $l2$ (note $l1, l2 \in \mathcal{D}_{loc}$). In our definitions, we will treat a r as a set of all locations enclosed by the region. We also define set $\mathcal{D}_{agt}$ which is the set of agents (in our application, shipping vessels).

In addition to the first-order logic syntax and semantics, we allow for annotation $[\ell, u]$ which is simply a subset of the unit interval $[0, 1]$ - which generalizes both fuzzy and classical logic. We write $a : [\ell, u]$ to mean that a has truth value associated with interval $[\ell, u]$. We refer the reader to [17, 34] for lattice-theory justification of this approach and how it generalizes other logical paradigms. We also note that we have learned our logic programs in a way to treat these bounds as confidence (see Section 3.3). We also follow the extension of temporal syntax and semantics [2, 5] to form temporally

annotated facts (TAFs) and annotated formulae. For an annotated literal f that is true at time t, f_t is a TAF. Annotated formulae are constructs formed with operators like AFTER(f, f'). For annotated formulae f, f', AFTER(f, f') is interpreted as f occurs after f'.

Example 3.1 (Language). *In our use-case, we consider an agent $agt \in \mathcal{D}_{agt}$ that travels among $loc1, loc2.. \in \mathcal{D}_{loc}$ in an AOI. The agent can be at a location covered by a region $r \in \mathcal{D}_r$ where $r \subseteq \mathcal{D}_{loc}$. We also define domain-specific binary predicate at where $at(agt, r)$ is a ground atom for an agent $agt \in \mathcal{D}_{agt}$ at a located in r indicating that the agent is within the region of r. We also define domain-specific unary predicates formed with $\mathcal{D}_{agt}$ constants: nearport, change-direction, high-speed, low-speed, hotspot, draught, ais-off and stay (expressing that the agent is near a port, changed its course sharply, has a high/low speed compared to an average, at a high-density hotspot, varied its draught, stopped transmitting AIS signals, and is at an anchor point by staying put for a long duration).*

As per previous work on temporal annotated logic [2, 5, 36], given a set of timepoints T, a set of all (ground) literals $\mathcal{G}$, an interpretation I is any mapping $\mathcal{G} \times T \rightarrow \mathcal{L}$. We define a satisfaction relationship "$\models$" and rules for temporally annotated extensions [2, 36]. A program Π is a set of rules, where each has an annotated atom in the head and a conjunction of annotated formulae in the body. An interpretation I is said to satisfy Π, if and only if I satisfies every rule and TAF in Π. The minimal model is an interpretation that can be thought of everything that can be concluded from deductive inference and commonly used for entailment queries in annotated logic [2, 17, 34, 36]. This is often computed using a fixpoint operator as done in the aforementioned work - and refer the reader to the well-established work on that topic for details. In this work, we slightly abuse the notation of [17] and use $\Gamma^*(\Pi)$ to denote the minimal model of Π.

Initial and Predicted Locations. In our problem, we must represent the initial conditions of the agent - in other words, the areas the shipping vessel has traveled in the first part of its voyage. We represent this simply with the logic program consisting of a set of temporally annotated facts formed with the predicate at introduced in Example 3.1. Here, we would expect fine-grain information on the location of the shipping vessel from information such as AIS - so each region (the second argument associated with the at-formed temporally annotated fact). We can think of such an initial logic program, Π_{init} being complemented by an additional logic program - also created with temporally annotated facts - used to represent the agent's behavior in the future - Π_{pred}. Intuitively, the elements of Π_{pred} would resemble the elements of Π_{init} except that they would occur after the facts of Π_{init}. Further, in practice, we would expect regions associated with Π_{pred} to be larger than Π_{init}. We shall refer to these logic programs Π_{pred}, Π_{init} as *region set program* and provide an example below.

Example 3.2. *Consider an agent $agt \in \mathcal{D}_{agt}$ in the Figure 10 that travels from time t_1 to t_i (we notate timestamps to be a set of timepoints = $\{t_1, ., t_i, .., t_n\}$ with a precedence relationship) and then goes dark at t_i, a time horizon t_{i+1}, then the initial conditions are represented as follows, Π_{init} = $\{$ $at(agt, r_{(31.14,46.12),(31.11,46.09)})_{t_1}$, .., $at(agt, r_{(30.88,46.48),(30.86,46.45)})_{t_i}\}$ and the predictions are represented as follows, Π_{pred} = $\{at(agt, r_{(30.87,46.51),(30.85,46.48)})_{t_{i+1}}$, $at(agt, r_{(30.82,46.51),(30.79,46.48)})_{t_{i+1}}$, $at(agt, r_{(30.88,46.48),(30.85,46.45)})_{t_{i+1}}$, $at(agt, r_{(30.87,46.50),(30.84,46.47)})_{t_{i+1}}$, $at(agt, r_{(30.87,46.49),(30.84,46.47)})_{t_{i+1}}\}$.*

Behavior Rules. We also envision a logic program consisting of a set of behavior rules of what the shipping vessel normally does (Π_{behav}). While it is possible to make these rules function as hard constraints, we instead make them soft constraints and instead measure how well an agent complies with these rules. This allows us to easily build a parsimony function. We provide example rules mined from data in Table 1.

Ground Truth Trajectories. Based on historical data, we also assume we have trajectory data for a given agent that occurs outside of Π_{init}. For a given agent, such a trajectory is simply a series of location-time tuples that were observed in the ground-truth data. So for agent agt, trajectory $\tau_{agt} = \langle((loc_1, t_1), \ldots, (loc_i, t_i), \ldots, (loc_n, t_n)\rangle$. We will define a notion of entailment of a trajectory at the syntactic level (though it is trivial to derive a semantic version). We say that program Π entails an agent's trajectory τ_{agt} if for all $(loc, t) \in \tau_{agt}$ there is some TAF $\text{at}(agt, \text{r})_t \in \Pi$ (which occurs at the same time) such that $loc \in$ r.

Example 3.3. *Following the notion built in Example 3.2, the trajectory for agent agt is, $\tau_{agt} = \langle((31.11, 46.00), t_1), .., ((30.87, 46.47), t_i), ((30.85, 46.48), t_{i+1}), ((30.81, 46.49), t_{i+2}), .., ((31.07, 46.00), t_n)\rangle$, then $\Pi_{init} \bigcup \Pi_{pred} \models \tau_{agt}$. Note that tuples of τ_{agt} - τ^1, τ^i are entailed by TAFs in Π_{init} - $(31.11, 46.00) \in r_{(31.14,46.12),(31.11,46.09)}$, $(30.87, 46.47) \in r_{(30.88,46.48),(30.86,46.45)}$ and the others can be entailed from Π_{pred} - $(30.85, 46.48) \in r_{(30.87,46.50),(30.84,46.47)}$, and $(30.81, 46.49) \in r_{(30.82,46.51),(30.79,46.48)}$.*

3.2 Abducing Agent Trajectories

For a single agent, we can think of finding Π_{pred} as an abduction problem. In other words, given an agent agt, initial conditions Π_{init}, behavioral rules Π_{behav}, and ground-truth trajectory τ_{agt} we want to find Π_{pred} such that:

(1) $\Pi_{init} \cup \Pi_{behav} \cup \Pi_{pred}$ is consistent (i.e., $\Gamma^*(\Pi_{init} \cup \Pi_{behav} \cup \Pi_{pred})$ exists).
(2) For each Π_{pred} entails τ_{agt}

If these criteria are met, we say Π_{pred} is an explanation for $\langle agt, \Pi_{init}, \Pi_{behav}, \tau_{agt}\rangle$. In this paper, our goal is to find a function that, based on historical data, can return an explanation. We define an *explanation function* as follows.

Definition 1 (Trajectory Explanation Function). *Given agents $agt^1, \ldots, agt^n$, initial condition programs $\Pi^1_{init}, \ldots, \Pi^n_{init}$, behavioral rules Π_{behav}, and trajectories $\tau^1, \ldots, \tau^n$, we say an explanation function f_E that takes as arguments and agent and two programs and returns a region set program such that $f_E(agt^i, \Pi^i_{init}, \Pi_{behav})$ is an explanation for $\langle agt^i, \Pi^i_{init}, \Pi_{behav}, \tau^i\rangle$.*

We note that Definition 1 is quite strict as it requires the result of f_E to produce a region set that models the entire trajectory for all agents. At the same time, it does not distinguish among different explanations. We introduce an approximation, $\hat{f_E}$ that is designed to meet the entailment requirement for as many agents as possible. Our solution is to leverage a notion of parsimony, defining $\hat{f_E}$ in terms of a parsimony function (σ) - which maps agents and logic

programs to scalars. The idea is to use σ to measure the quality of an explanation so that we can find quality explanations that cover most of the ground truth trajectories. We provide the following examples of such a function.

$$\hat{f}_1(agt, \Pi_{init}, \Pi_{behav}) = \arg\max_{\Pi'} \sigma(agt, \Pi_{init} \cup \Pi_{behav} \cup \Pi')$$

$$\hat{f}_2(agt, \Pi_{init}, \Pi_{behav}) = \{\arg\max_{\phi} \sigma(agt, \Pi_{init} \cup \Pi_{behav} \cup \{\phi\})\}$$

In these two examples, we note the first has a combinatorial flavor - finding the best set of regions, while the second identifies the best singleton set - a notion that we can extend to find the top k singletons (which correspond to the top k regions formed with the at). This can be easily solved by multiple entailment problems for each relevant singleton formed from atoms created with set $\mathcal{D}_{\mathsf{r}}$ (which we assume is known a-priori). We also note that the computation of $\hat{f}_2$ can be computed in linear time (in the number of TAFs) which results directly from the prior results on annotated logic [17, 34] and allows us to leverage existing efficient implementations [2]. We verify this empirically (Figure 9). In this work, we examine the top-k variant of $\hat{f}_2$ and provide empirical evidence that supports it. In practice, we compute top-k regions -corresponding to the TAF $at(agt, \mathsf{r})$ (picking r from $\mathcal{D}_{\mathsf{r}}$) in parallel.

3.3 Rule-Based Agent Behavioral Modeling

As described in Section 3.1 we assume that there exists a set of rules Π_{behav} specifying the behavior of the agents. While we could design Π_{behav} to allow for hard constraints on consistency (and while there are good reasons for doing so), we instead leverage the fuzzy nature of our underlying logic (as described in Section3.1) which can then allow us to easily build an explainable parsimony function σ. Again, this function takes an agent and a logic program as arguments (and the logic program, Π, is the union of the initial conditions Π_{init} and behavior rules Π_{behav}) and returns a scalar. As we use the logical paradigm of [17, 34], each logical atom is associated with a subset of the unit interval - $[\ell, u]$. In this work, define the parsimony function as the aggregate over the lower bound of the interval, formally:

$$\sigma_t(agt, \Pi) \quad = \quad lb\Big(\Gamma^*(\Pi)(normal(agt))(t)\Big)$$

Intuitively, we have a predicate *normal*, such that atoms formed with that predicate are annotated with an interval measuring the agent's level of normalcy. The minimal model of the program, $\Gamma^*(\Pi)$ provides this annotation for a particular atom - here $normal(agt)$ (the normalcy of agent agt) and time t (we can define σ for a particular time - in practice we use the maximum time as it allows us to cover long-term predictions). Finally, lb returns the lower bound of the interval (as we will learn rules in a manner where we set the upper bound to 1 to easily ensure consistency).

Rule Learning Algorithm. From the training set, a set of rules is learned to model the normal behavior of the vessels based on the historical co-occurrences of periodic sequences among similar types of ships in similar waters. They are learned in a method akin to rule learning in [5, 33] where we restrict the body to have a single sequence of movement, refer Algorithm 1. These rules are population-specific among the vessels. Here, consider τ to be a set of the associated region of the trajectory. We note that Algorithm 1 is quite efficient. It scans all trajectories in a given data. The quantity of trajectory size in terms of regions can be treated as a constant as it's from a data source. Hence, it turns out that Algorithm 1 is linear in terms of the size of the dataset (number of trajectories).

Algorithm 1 Behavioral Rule Learner

Input: A set of trajectories T, atom $normal(agt)$
Output: A set of rules Π
function RULES($Body$)
  $\Pi \leftarrow \emptyset$
  for all $moves \in Body$ **do**
    if length($moves$) = 2 **then**
      $mov \leftarrow Body[moves][0]$
      $\Pi \leftarrow \Pi \bigcup \{normal(agt) : [\frac{Body[moves]}{Body[mov]}, 1] \leftarrow \bigwedge_{m \in moves} m(agt)\}$
    end if
  end for
  return Π
end function
function TRAINMODEL(T)
  Initialize dictionary $Body \leftarrow \emptyset$
  for all τ in T **do**
    for $n \leftarrow 1$ **to** length(τ)-1 **do**
      $Body[\tau[n]] \leftarrow Body[\tau[n]] + 1$
      $Body[\tau[n-1]] \leftarrow Body[\tau[n-1]] + 1$
      $Body[(\tau[n-1], \tau[n])] \leftarrow Body[(\tau[n-1], \tau[n])] + 1$
    end for
  end for
  $\Pi \leftarrow$Rules($Body$)
  return Π
end function

Here the movement is considered to be among regions representing features like port regions, density-based historical hotspots, anchor points, destinations, and typically observed maritime feature (speed over ground, course over ground, and heading) spikes in the waters. We define two kinds of rules based on the movement from the current region. It could be one (single-hop rules (SH)) or multiple hops (multi-hop rules (MH)) away to the next region. For multi-hop rules, intuition is to capture movements that occur eventually rather than in the next movement from the current region. Some sample rules that we actually mined from maritime vessel data are shown in Table 1. The annotations on the head of the rules note the measure of confidence in the normalcy of the rule.

4 EXPERIMENTAL RESULTS

Setup. We parsed Automatic Identification System (AIS) data of 614 vessels across the Black Sea area of interest (AOI) from January 2022 to March 2023. This involves the trajectory data τ of each vessel in addition to its dynamic and statistical information. This data has trajectories of the length 2 to 165,000 data points (i.e., the vessel's latitude, longitude, timestamp, other features[1]) that

[1]Information from an AIS signal, https://spire.com/whitepaper/maritime/introduction-to-automatic-identification-systems-ais/

Table 1: Example Rules Mined From Historical Data

Rule	Natural Language
$normal(AGT) : [0.8, 1] \leftarrow nearport(AGT) : [1, 1] \wedge high - hotspot(AGT) : [1, 1] \wedge \mathsf{AFTER}(high - hotspot(AGT), nearport(AGT)) : [1, 1]$	*Example Multi-hop rule.* The confidence of a vessel exhibiting normal behavior is at least 0.8 when the agent goes from a near port to a high-hotspot region in more than a single movement.
$normal(AGT) : [0.9, 1] \leftarrow low - speed(AGT) : [1, 1] \wedge sharp - course - change(AGT) : [1, 1] \wedge \mathsf{AFTER}(sharp{-}course{-}change(AGT), low{-}speed(AGT)) : [1, 1]$	*Example Single-hop rule.* The confidence of a vessel exhibiting normal behavior is at least 0.9 when the agent changes its course direction after lowering its speed in a single movement.

span from 1 to 264 days. For all our experiments, we use a high memory compute node, Dell PowerEdge R6525 with the AMD EPYC 7713 64-Core Processors and 2TB RAM, and a GPU node, Dell PowerEdge R7525 with the AMD EPYC 7413 24-Core Processors, 512GB RAM along with three A30 GPUs. The region size is fixed arbitrarily at $0.025^\circ \times 0.025^\circ$ which comes to $5.45km^2$ in our AOI for our experiments unless specified.

Extending prior work [18, 30] where similar vessels were grouped, we perform trajectory clustering [39] to group trajectories into 9 subset. Clustering is performed with DBSCAN and we report average metrics across all clusters for both our method and our deep learning baseline.

Since we aim to generate regions at a future time, we mask each test trajectory to obtain a partial trajectory. The masked part is considered the ground truth (ground truth trajectory τ_{agt} in our notation) while the unmasked part is used to set the initial condition (logic program Π_{init}). We mask half the trajectory from its midpoint in all our experiments unless specified.

Methods. We examine three methods, described as follows.

Random baseline (RND). The random method randomly generates regions from the AOI grid. The AOI grid is formed with cells of the fixed region size. The average performance of three random generators is reported.

Deep learning baseline (DL). For the DL baseline, we use a sequence-to-sequence model [10] to predict future trajectories. To perform a comparable evaluation, the predicted sequence is mapped to regions in the AOI grid. Here k is a hyperparameter considered as the first k boxes encountered by the predicted sequence. We also evaluated a deep learning baseline trained on all the data (DL-ALL), which generally was not performant beyond $k = 4$ limiting its F1 - we include results from that model only in experiments where it significantly outperforms DL models on subsets. We experimented with variants of [10] with alternative architectures to mimic similar to point-based prediction models [16, 46] but these achieved worse results than DL and DL-ALL.

Abduction (ABD). The abduction method uses train data to obtain a set of regions (which is the subset of the AOI grid), from which it learns SH rules to obtain Π. Given a test trajectory, it then generates top k regions using $\hat{f}_E$ via abductive inference.

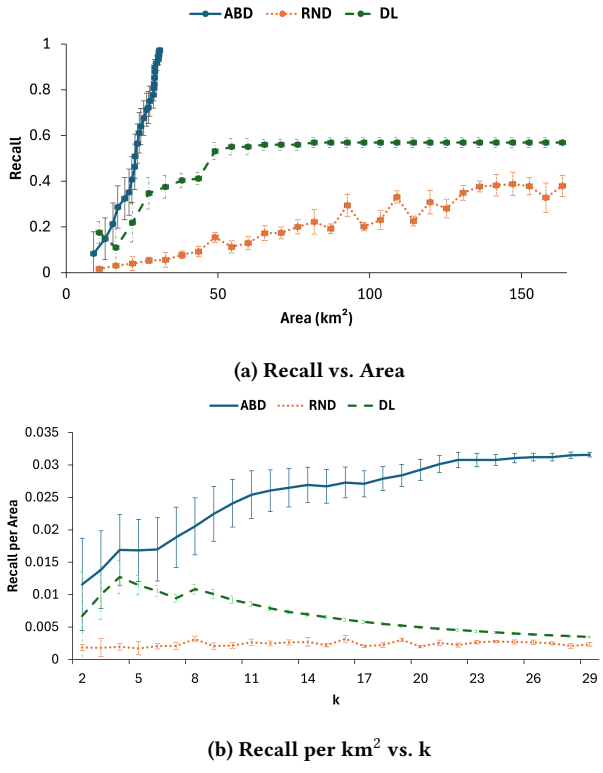
Metrics. We report precision as the fraction of returned regions that contain points in the ground truth trajectory. Likewise, recall is the ratio of returned regions containing ground truth points to all regions containing irredundant points from the ground truth trajectory. The F1 is the harmonic mean of these quantities.

4.1 Experiments

We examine the ABD, RND, and DL approaches when applied to AIS data. We first inspect the area efficiency, which has practical significance. We then evaluate the methods for long-term reasoning capabilities. Further, we compare all approaches as a function of k in a standard setting. We also provide hyperparameter sensitivity concerning region size as well as ablation studies for Π (based on different rule types, e.g. SH, MH rules), and the versatility to masking methods of the test trajectory. Finally, we assess ABD while limiting the training data. We conclude by showing the interpretability of results in ABD.

Area Efficiency. In our application, we wish to identify the greatest number of locations for dark vessels while searching the smallest area possible - as identification of dark vessels would require resources such as aerial or satellite imaging. We examine recall as a function of area in Figure 1a. We found that recall for ABD saturates at 30km^2 - achieving a recall of 0.99, which is 157% higher recall than DL for that area. DL meanwhile saturates at 81.75km^2 - achieving recall of only 0.57. This difference suggests that ABD provides more efficiency per unit area. To further investigate this efficiency, we examine how it trends as a function of k (number of regions) in Figure 1b. It turns out that the recall per square kilometer monotonically increases with k for ABD while it decreases for the baselines. This implies that ABD can continue to produce quality regions. This is significant for practitioners because, when additional search resources are available, ABD can continue to improve search efficiency with the addition of more search resources.

Long-term Reasoning. The prior experiments examined performance under the assumption of a fixed time horizon. Next, we examine performance across multiple time horizons and show the results in Figure 2. Here we examine each approach with different settings for k but find that ABD again consistently outperforms other methods in terms of F1. We also note that ABD is the only approach where an increase in k improves results (e.g., DL achieves poorer performance with $k = 10$ vs. $k = 5$). This suggests that our previously described efficiency results likely hold to the case of multiple time units while DL convergers by leveling off after



(a) Recall vs. Area

(b) Recall per km^2 vs. k

Figure 1: Area Efficiency: (a) Relationship between Recall and Area, (b) Recall per km^2 as a function of k

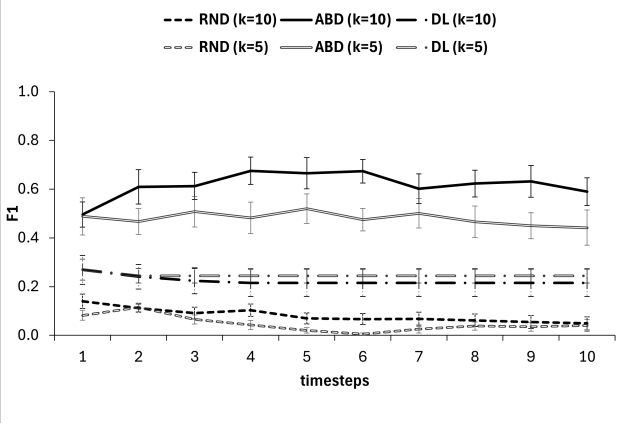
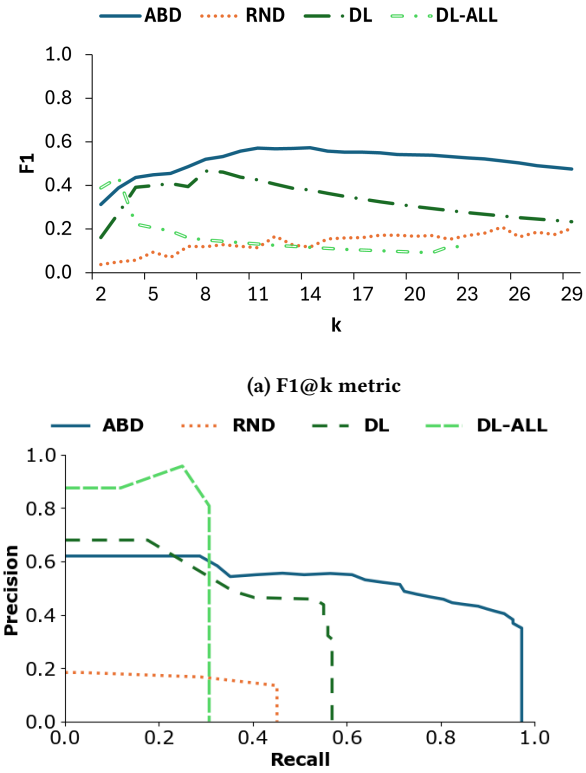


Figure 2: Long-term reasoning. F1@{k=5,k=10} for Abduction, DL, and Random baselines.

the first time horizon. This illustrates that with increasing time horizons, DL is not able to predict long-term trajectories.

Vessel Recall and Accuracy. We examine ABD, DL (when trained on the subset datasets individually, as well as on the entire train set), and RND allowing for different values of k (number of regions). Note that the default DL is DL-Subset. Figure 3a shows that across all settings of k, ABD outperforms all other methods in terms of

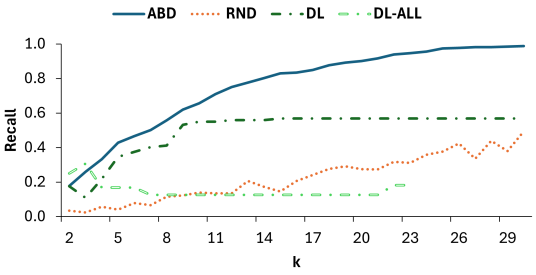


(a) F1@k metric

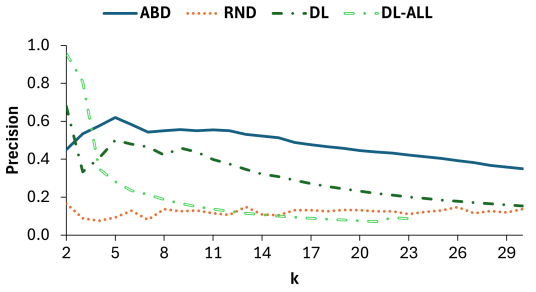
(b) Precision-Recall curve

Figure 3: Comparison of (a) F1@k metric and (b) Precision-Recall curve

F1 - and ABD on average provides a 51% increase over DL. We note for higher values of k, DL starts to converge with the random baseline (i.e., around $k = 28$) while ABD maintains approximately double the F1 score. When we examine the precision-recall curve in Figure 3b, we gain an intuition as to why the F1 flags for the DL approach - and the answer is that the recall of DL saturates at 0.57 - indicating limited value in adding more regions (increasing k) where ABD can obtain a recall approaching 1 while increasing k, with graceful degradation of precision. ABD has a 476% increase in the recall by adding more regions up to $k = 30$. In this experiment, we also recorded the results of DL-ALL (a single model trained with the whole dataset instead of the sub-datasets). For $k = 3, 4$, DL-ALL gave the highest F1 due to larger precision values (as seen in Figure 4b) but the performance degrades by 56% for a unit increase in k, and as k increases to 30, it degrades further. Additionally, as seen in Figure 3b, similar to DL, DL-ALL also saturates and does not achieve a recall beyond 0.3, explaining the decrease in F1. Also, DL-ALL performed 50% lower on F1 when compared to standard DL. DL-ALL gives the highest F1 for time horizons very near to the current time. While this is less relevant for our current application, it may provide insight for further inquiry (e.g., a neurosymbolic approach leveraging both abduction and a model trained on a large number of trajectories).



(a) Recall@k metric



(b) Precision@k metric

Figure 4: Comparison of ML metrics- (a) Recall@k and (b) Precision@k.

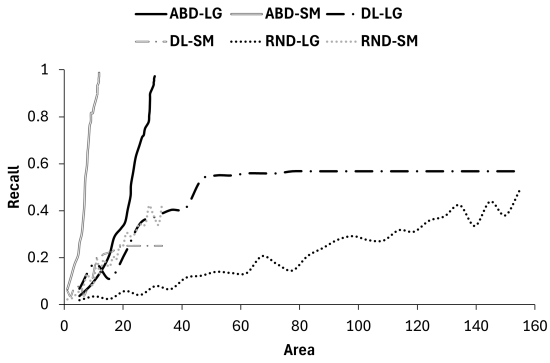


Figure 5: Region size sensitivity for different region-sizes (of LG denotes the larger region size = 5.45km^2 and SM denotes the smaller region size= 1.1 km^2)

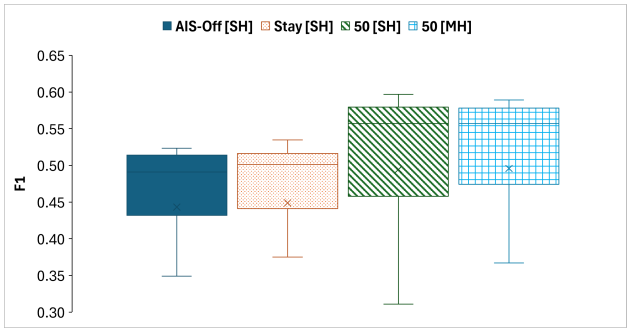


Figure 6: F1 scores for various masking methods and rules, including AIS Off (SH), Stay (SH), 50 End (SH), and 50 End (MH), where SH represents single hop rules and MH represents multi-hop rules.

Region Size Sensitivity. In the aforementioned experiments, we determined the region size by considering the computational efficiency of rule learning and generating regions with fair coverage- so that a single region does not end up covering the entirety of the vessel-search space. We now call that setting LG, while the setting SM is when we reduce the region size by 80%.

Note that reducing the region size (SM) is effective by itself as seen by RND-SM achieving comparable performance to DL-LG up to a certain extent. However, ABD outperforms all baselines particularly when the region size is reduced. The curve is steeper for ABD when the region size is decreased by 80% from LG to SM depicted in Figure 5, while that of DL resembles its performance in the earlier experiment. We found that not only that our results maintain with reduced region size, but they also led to improved performance in ABD (reducing the total search area by about 60%) while the reduction in region size did not meaningfully change the performance of DL.

Ablation by Rule Type and Masking Sensitivity. As described in Section 3.3 we developed several methods to learn rules (see Table 1). In Figure 6, our abduction (ABD) approach still works well for different kinds of rules like single-hop and multi-hop rules. Note that single hop has a slightly wider range of F1 scores with respect to the lower extremes by 0.05 while the upper extremes and medians are similar. Additionally, we also wanted to examine the impact of the type of masking on the results - from a practical standpoint to model applications for detecting deceiving vessels who tamper with their AIS transmitter. Here, the masking would typically start from a point where the AIS is tampered with. The abduction model works under various masking settings like when the making starts where the vessel does not transmit any AIS signal for more than normal time or stays in the same region for too long based on historical data. This shows that the model can be applied to both the detection of dark vessels and detecting vessels when they tamper with their AIS to violate sanctions.

Data Efficiency. The abduction (ABD) model also works well with limited training trajectories as seen in Figure 7 while DL-based methods are more data-driven as seen in Figure 8. Note that for ABD, the use of a single training trajectory versus all of the historical data gave the same precision of 0.62 and a 0.13 difference in F1. On the other hand, as expected, DL has a huge variation with increasing training trajectories by boosting its performance by 254% as seen in Figure 8. This gives scope for the application of this model with expensive or limited available data.

Runtime. In Figure 9 we examine the runtime of ABD as a function of the number of regions k. As expected, the runtime increases linearly with k as this simply involves additional deductive steps. Further, we note that the deduction itself is efficient (linear in the size of nodes) as previously reported [2, 34].

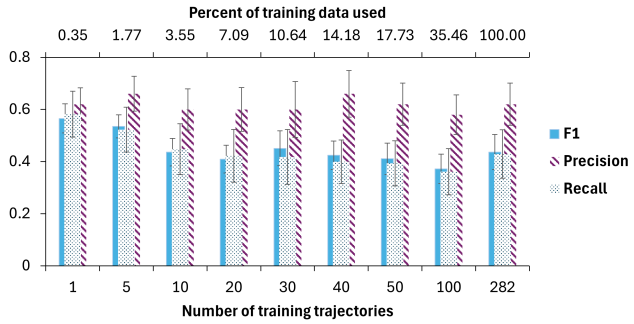


Figure 7: Evaluation of ABD with limited training data (trajectories). ABD performed smoothly, with variations in the number of training trajectories.

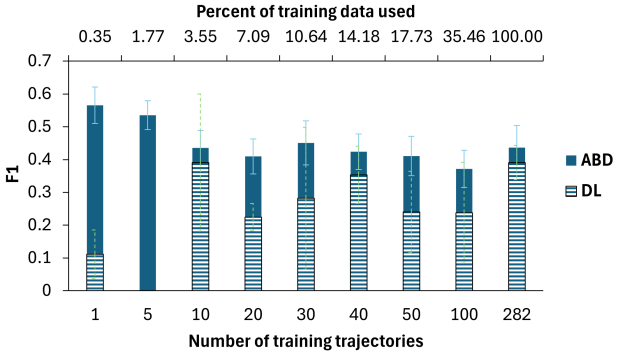


Figure 8: Evaluation of ABD and DL with limited training data (trajectories). Note that DL achieved an F1 near zero with five training trajectories.

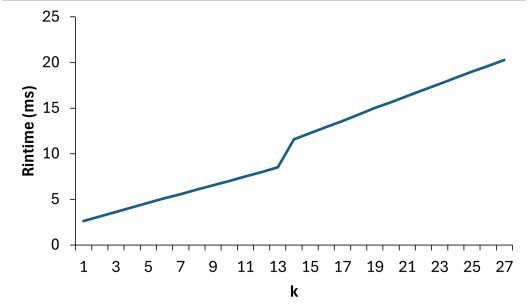


Figure 9: Evaluation of runtime in terms of milliseconds of ABD as a function of k

Explainability. All regions are symbolic in nature, every inference can be backtracked to the sequence of historically learned rules, in addition to its confidence as seen in Figure 10. This gives scope for domain experts in analyzing false predictions, assess vessel behavior, and even incorporating domain knowledge into the rules.

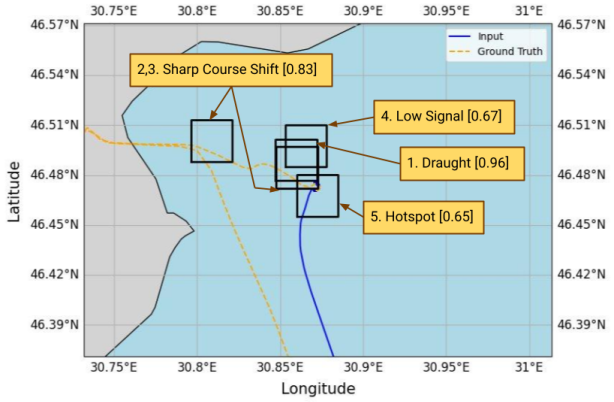


Figure 10: Abduction model predictions. The dashed trajectory is the ground truth. Black regions are the generated regions with corresponding confidence and region types

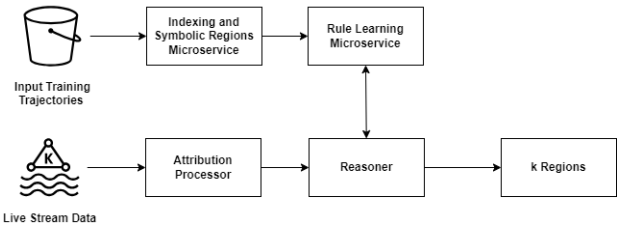


Figure 11: Deployment of abduction model in an online learning setting with kafka

5 DEPLOYMENT

We have designed a prototype system based on the abduction model with a live feed of trajectories, where it continuously updates its logic program as it generates regions in an online learning setting. This architecture is depicted in Figure 11. We have selected a microservices-based architecture for near real-time detection of maritime dark vessels that receives input training data delivered by data providers to an Amazon S3 bucket. The arrival of new data triggers a batch process that performs data indexing and generates symbolic regions. This processed data is then fed into a rule-learning microservice, which is subsequently transformed into a logic program by learning rules that are staged into the S3 bucket.

In the production environment, live data is streamed via a near real-time Kafka feed. We use Apache Kafka to consume the AIS data stream in near real-time as a streaming architecture. An attribution processor subscribes to this feed and enriches the incoming data by tagging it with the necessary symbolic region and indexing metadata. The enriched data is then integrated into the logic program (which includes both updated rules and TAFs), before being fed into the reasoner (Γ^*), which infers k regions at a given time horizon. We then use Quantum Geographic Information System software to visualize the regions in the area of interest for an end-user.

6 CONCLUSION

We identify the locations of dark maritime vessels using a combination of abductive inference and rule learning and provides explainable long-time horizon prediction - an area where machine learning approaches fail. These aspects were validated by our experimental results and we provide our deployment architecture with a live feed of data. This work can be extended by leveraging environmental knowledge in the logic program, which has a significant role in the maritime domain where we look to utilize techniques from neurosymbolic AI [35] that will enable the use of larger scale models for enhanced near-term precision while retaining the long-term reasoning ability of the abduction methods introduced in this paper.

ACKNOWLEDGMENTS

This work was funded by ONR grant N00014-23-1-2580. This research used maritime data provided by Spire.

REFERENCES

[1] 2021. Government of Canada launches international program to track illegal fishing using satellite technology. https://www.canada.ca/en/fisheries-oceans/news/2021/02/government-of-canada-launches-international-program-to-track-illegal-fishing-using-satellite-technology.html. Accessed: 2024-08-31.

[2] Dyuman Aditya, Kaustuv Mukherji, Srikar Balasubramanian, Abhiraj Chaudhary, and Paulo Shakarian. 2023. PyReason: Software for Open World Temporal Logic. https://doi.org/10.48550/arXiv.2302.13482 arXiv:2302.13482 [cs.LO]

[3] Adriana Ávila-Zúñiga-Nordfjeld, Hans Liwång, and Dimitrios Dalaklis. 2023. *Implications of Technological Innovation and Respective Regulations to Strengthen Port and Maritime Security: An International Agenda to Reduce Illegal Drug Traffic and Countering Terrorism at Sea.* Springer International Publishing, Cham, 135–147. https://doi.org/10.1007/978-3-031-25296-9_7

[4] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies, Vol. 1. 57–64. https://doi.org/10.1145/1558013.1558020

[5] Divyagna Bavikadi, Dyuman Aditya, Devendra Parkar, Paulo Shakarian, Graham Mueller, Chad Parvis, and Gerardo I Simari. 2024. Geospatial Trajectory Generation via Efficient Abduction: Deployment for Independent Testing. In *Proceedings of the 40th International Conference on Logic Programming (ICLP 2024).*

[6] Branislav Bosansky, Viliam Lisý, Michal Jakob, and Michal Pechoucek. 2011. Computing time-dependent policies for patrolling games with mobile targets. *10th International Conference on Autonomous Agents and Multiagent Systems 2011, AAMAS 2011* 2, 989–996.

[7] Capella Space. 2024. https://www.capellaspace.com/press-releases/capella-space-introduces-fully-automated-vessel-classification-capability. Accessed: 2024-08-31.

[8] Liyue Chen, Jiangyi Fang, Zhe Yu, Yongxin Tong, Shaosheng Cao, and Leye Wang. 2023. A Data-driven Region Generation Framework for Spatiotemporal Transportation Service Management. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23).* ACM. https://doi.org/10.1145/3580305.3599760

[9] Fei Fang, Albert Xin Jiang, and Milind Tambe. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13).* International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 957–964.

[10] Nicola Forti, Leonardo M. Millefiori, Paolo Braca, and Peter Willett. 2020. Prediction oof Vessel Trajectories From AIS Data Via Sequence-To-Sequence Recurrent Neural Networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 8936–8940. https://doi.org/10.1109/ICASSP40776.2020.9054421

[11] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility Markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility* (Bern, Switzerland) *(MPM '12).* Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages. https://doi.org/10.1145/2181196.2181199

[12] Maria Daniela Graziano and Alfredo Renga. 2021. Towards Automatic Recognition of Wakes Generated by Dark Vessels in Sentinel-1 Images. *Remote Sensing* 13, 10 (2021). https://doi.org/10.3390/rs13101955

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. arXiv:1803.10892 [cs.CV] https://arxiv.org/abs/1803.10892

[14] Zhihui He, Lei Ning, Baihui Jiang, Jiajia Li, and Xin Wang. 2023. Vehicle Intersections Prediction Based on Markov Model with Variable Weight Optimization. *Sustainability* 15, 8 (2023). https://doi.org/10.3390/su15086943

[15] Homeland Security News Wire. 2009. https://www.homelandsecuritynewswire.com/us-navys-panda-technology-detect-deviant-ships. Accessed: 2024-08-31.

[16] Dapeng Jiang, Guoyou Shi, Na Li, Lin Ma, Weifeng Li, and Jiahui Shi. 2023. TRFM-LS: Transformer-Based Deep Learning Method for Vessel Trajectory Prediction. *Journal of Marine Science and Engineering* 11, 4 (2023). https://doi.org/10.3390/jmse11040880

[17] Michael Kifer and V.S. Subrahmanian. 1992. Theory of Generalized Annotated Logic Programming and its Applications. *J. Log. Program.* 12, 3&4 (1992), 335–367. https://doi.org/10.1016/0743-1066(92)90007-P

[18] Huanhuan Li, Hang Jiao, and Zaili Yang. 2023. AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. *Transportation Research Part E: Logistics and Transportation Review* 175 (2023), 103152. https://doi.org/10.1016/j.tre.2023.103152

[19] Huanhuan Li, Jasmine Siu Lee Lam, Zaili Yang, Jingxian Liu, Ryan Wen Liu, Maohan Liang, and Yan Li. 2022. Unsupervised hierarchical methodology of maritime traffic pattern extraction for knowledge discovery. *Transportation Research Part C: Emerging Technologies* 143 (2022), 103856. https://doi.org/10.1016/j.trc.2022.103856

[20] Maohan Liang, Lingxuan Weng, Ruobin Gao, Yan Li, and Liang Du. 2024. Unsupervised maritime anomaly detection for intelligent situational awareness using AIS data. *Knowledge-Based Systems* 284 (2024), 111313. https://doi.org/10.1016/j.knosys.2023.111313

[21] Xintong Liu, Yutian Hu, Huiting Ji, Mingyang Zhang, and Qing Yu. 2023. A Deep Learning Method for Ship Detection and Traffic Monitoring in an Offshore Wind Farm Area. *Journal of Marine Science and Engineering* 11, 7 (2023). https://doi.org/10.3390/jmse11071259

[22] Lockheed Martin Advanced Technology Laboratories. 2006. https://news.lockheedmartin.com/2009-11-11-Lockheed-Martin-Program-Helps-Detect-Deviations-in-Maritime-Vessel-Normal-Behavior?mobile=No&asPDF=1. Accessed: 2024-08-31.

[23] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. 2020. From Goals, Waypoints Paths To Long Term Human Trajectory Forecasting. arXiv:2012.01526 [cs.CV] https://arxiv.org/abs/2012.01526

[24] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. 2020. It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction. arXiv:2004.02025 [cs.CV] https://arxiv.org/abs/2004.02025

[25] Marine Link. 2024. Dark Tanker Fleet Activity. https://www.marinelink.com/news/dark-tanker-fleet-activity-morphs-517608. Accessed: 2024-08-31.

[26] U.S. Department of Homeland Security Navigation Center, United States Coast Guard. 2024. Automatic Identification System (AIS) Overview. (2024). https://www.navcen.uscg.gov/automatic-identification-system-overview Accessed: 2024-08-31.

[27] Trung Nguyen. 2023. The Challenges of Dark Ships to the Safety and Security of Commercial Shipping and the Way Forward. *Asia-Pacific Journal of Ocean Law and Policy* 8, 2 (2023), 310 – 328. https://doi.org/10.1163/24519391-08020007

[28] Irwan Priyanto and Aniati Murni Arymurthy. 2021. Vessel Detection Based on Deep Learning Approach. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI).* 91–96. https://doi.org/10.1109/ISRITI54043.2021.9702879

[29] H. Rong, A.P. Teixeira, and C. Guedes Soares. 2020. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Engineering* 198 (2020), 106936. https://doi.org/10.1016/j.oceaneng.2020.106936

[30] H. Rong, A.P. Teixeira, and C. Guedes Soares. 2024. A framework for ship abnormal behaviour detection and classification using AIS data. *Reliability Engineering System Safety* 247 (2024), 110105. https://doi.org/10.1016/j.ress.2024.110105

[31] Elham Shaabani, Hamidreza Alvari, Paulo Shakarian, and J. E. Kelly Snyder. 2016. MIST: Missing Person Intelligence Synthesis Toolkit. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 1843–1867. https://doi.org/10.1145/2983323.2983346

[32] Amir Yaghoubi Shahir, Mohammad A. Tayebi, Uwe Glässer, Tilemachos Charalampous, Zahra Zohrevand, and Hans Wehn. 2019. Mining Vessel Trajectories for Illegal Fishing Detection. In *2019 IEEE International Conference on Big Data (Big Data).* 1917–1927. https://doi.org/10.1109/BigData47090.2019.9006545

[33] Paulo Shakarian, Austin Parker, Gerardo Simari, and Venkatramana V. S. Subrahmanian. 2011. Annotated probabilistic temporal logic. *ACM Trans. Comput. Logic* 12, 2, Article 14 (jan 2011), 44 pages. https://doi.org/10.1145/1877714.1877720

[34] Paulo Shakarian and Gerardo I. Simari. 2022. Extensions to Generalized Annotated Logic and an Equivalent Neural Architecture. In *Proceedings - 2022 4th International Conference on Transdisciplinary AI, TransAI 2022*. Institute of Electrical and Electronics Engineers Inc., 63–70. https://doi.org/10.1109/TransAI54797.2022.00017

[35] Paulo Shakarian, Gerardo I. Simari, Chitta Baral, Bowen Xi, and Lahari Pokala. 2023. *Neuro Symbolic Reasoning and Learning*. Springer-Nature.

[36] Paulo Shakarian, Gerardo I. Simari, and Robert Schroeder. 2013. MANCaLog: a logic for multi-attribute network cascades. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1175–1176.

[37] Paulo Shakarian, V. S. Subrahmanian, and Maria Luisa Sapino. 2011. GAPs: Geospatial Abduction Problems. *ACM Trans. Intell. Syst. Technol.* 3, 1 (2011), 7:1–7:27. https://doi.org/10.1145/2036264.2036271

[38] Spire Global, Inc. 2024. AIS Position Validation. https://insights.spire.com/maritime/dark-shipping-detection. Accessed: 2024-08-31.

[39] Cynthia Sung, Dan Feldman, and Daniela Rus. 2012. Trajectory clustering for motion prediction. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1547–1552. https://doi.org/10.1109/IROS.2012.6386017

[40] Joe Wallace, Anna Hirtenstein, and Costas Paris. 2024. Russia's Oil Trading Thrives Amid Secret Network and Sanctions. *The Wall Street Journal* (19 Feb. 2024). https://www.wsj.com/business/energy-oil/russia-oil-trading-secret-network-sanctions-fc3981b7 Accessed: 2024-08-31.

[41] Windward Ltd. 2022. https://windward.ai/content/hiding-in-plain-sight-not-all-that-transmit-are-legit/. Accessed: 2024-08-31.

[42] Windward Ltd. 2024. https://windward.ai/knowledge-base/2-years-of-turmoil-that-transformed-the-maritime-ecosystem/. Accessed: 2024-08-31.

[43] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. 2018. A Generative Model of Urban Activities from Cellular Data. *IEEE Transactions on Intelligent Transportation Systems* 19, 6 (2018), 1682–1696. https://doi.org/10.1109/TITS.2017.2695438

[44] Xiaocai Zhang, Xiuju Fu, Zhe Xiao, Haiyan Xu, and Zheng Qin. 2022. Vessel Trajectory Prediction in Maritime Transportation: Current Approaches and Beyond. *IEEE Transactions on Intelligent Transportation Systems* 23, 11 (2022), 19980–19998. https://doi.org/10.1109/TITS.2022.3192574

[45] Xianyang Zhang, Gang Liu, Chen Hu, and Xiaolong Ma. 2019. Wavelet Analysis Based Hidden Markov Model for Large Ship Trajectory Prediction. In *2019 Chinese Control Conference (CCC)*. 2913–2918. https://doi.org/10.23919/ChiCC.2019.8866006

[46] Shenjie Zou, Jin Liu, Xiliang Zhang, Zijun Yu, Bing Han, and Jiamao Zhi. 2023. A Dynamic Spatio-Temporal Refinement Network for Multi-Vessel Trajectory Prediction. In *Proceedings of the 2023 International Conference on Computer, Vision and Intelligent Technology* (Chenzhou, China) *(ICCVIT '23)*. Association for Computing Machinery, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3627341.3630416