Maritime Threat Detection using Plan Recognition

Bryan Auslander[1], Kalyan Moy Gupta[1], & David W. Aha[2]
[1]Knexus Research Corp.; 163 Waterfront Street; National Harbor, MD 20745
[2]Navy Center for Applied Research in Artificial Intelligence; Naval Research Laboratory, Code 5514;
4555 Overlook Ave. SW; Washington, DC 20375
firstname.lastname@knexusresearch.com | david.aha@nrl.navy.mil

Abstract. **Existing algorithms for maritime threat detection employ a variety of normalcy models that are probabilistic and/or rule-based. Unfortunately, they can be limited in their ability to model the subtlety and complexity of multiple vessel types and their spatio-temporal events, yet their representation is needed to accurately detect anomalies in maritime scenarios. To address these limitations, we apply plan recognition algorithms for maritime anomaly detection. In particular, we examine hierarchical task network (HTN) and case-based algorithms for plan recognition, which detect anomalies by generating expected behaviors for use as a basis for threat detection. We compare their performance with a behavior recognition algorithm on simulated riverine maritime traffic. On a set of simulated maritime scenarios, these plan recognition algorithms outperformed the behavior recognition algorithm, except for one reactive behavior task in which the inverse occurred. Furthermore, our case-based plan recognizer outperformed our HTN algorithm. On the short-term reactive planning scenarios, the plan recognition algorithms outperformed the behavior recognition algorithm on routine plan following. However, they are significantly outperformed on the anomalous scenarios.**

I. INTRODUCTION

Early detection and neutralization of threats from small boats is a critical requirement for the US Navy. Detection of small boat threats is particularly challenging in busy ports, harbors and riverine areas because they operate in close proximity to large but much less maneuverable vessels. Various approaches for automated threat and anomaly detection have been developed to address this problem. For example, perimeter breach detection algorithms (e.g., [1][2]) trigger an alarm when a distance threshold is crossed. However, these can lead to numerous false alarms and unacceptable operating requirements in narrow traffic lanes. To address this problem, we recently showed that some behavior-based threat detection algorithms, based on probabilistic graphical models, can outperform perimeter breach algorithms [3]. However, perimeter and behavior-based approaches rely on learning normalcy models that predominately focus on low level activities, but cannot detect anomalies that require knowledge of planned routes and schedules.

In this paper, we apply plan recognition algorithms to continuously monitor activities and identify the plans of maritime vessels in advance of any threat occurrence. We assume that, given a plan, we can identify its unique goal, and conjecture that identifying goals could help to detect threatening and anomalous situations earlier and more accurately. Also, the alerts that are generated by plan recognition algorithms are better suited for explaining threats than are opaque and non-intuitive statistical models.

Plan recognition algorithms have been used for a variety of tasks, such as detection of anomalous situation in an assisted care facility [4] and detection of terrorist threats [5]. However, they have rarely been applied to maritime threat recognition, and not empirically compared with conventional behavior recognition algorithms for threat detection.

We address these issues as follows. We apply two plan recognition algorithms; the first uses hierarchical task networks (HTNs), while the other uses case-based planning. These algorithms apply consistency-based plan recognition techniques [6]. We apply them to four simulated maritime scenarios and compare their performance versus a Markov logic network (MLN) algorithm, a probabilistic behavior recognizer that performed well in our prior studies on maritime threat detection [3]. Our results show that the plan recognition algorithms outperform this probabilistic behavior recognizer for scenarios involving longer-term plans. However, the results on scenarios with reactive plans involving anomaly situations are mixed.

We structure the remainder of this paper as follows. We provide an overview of plan recognition approaches followed by a detailed description of HTN and case-based approaches. We then describe our empirical evaluation and results. Finally, we conclude with a discussion and issues for future research.

II. PLAN RECOGNITION OVERVIEW

Plan recognition is the task of inferring plan(s) of an intelligent agent by observing the agent's actions or the effects of those actions. It involves mapping a temporal sequence of *observed actions* to some *plan representation* that identifies the plan's goal and the relation of actions among the plan's components [7]. Plan recognition algorithms can be categorized into consistency-based and probabilistic approaches [8]. The former include hypothesize and revise algorithms, version space techniques, and other closed-world reasoning algorithms, while probabilistic algorithms include those that use stochastic grammars and probabilistic relational models. The maritime environment is a continuous, non-deterministic domain, making it challenging to apply any plan recognition approach. As a first attempt, we consider only consistency-based approaches.

III. MARITIME PLAN RECOGNITION

We develop consistency-based plan recognition algorithms for the maritime domain as follows (see Figure 1). We assume

Report Documentation Page

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Maritime Threat Detection using Plan Recognition	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Knexus Research Corp,163 Waterfront Street,National Harbor,MD,20745	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the Twelfth Annual IEEE Conference on Technologies for Homeland Security, 13-15 Nov 2012, Greater Boston, MA, pp. 249-254. U.S. Government or Federal Rights License

14. ABSTRACT
Existing algorithms for maritime threat detection employ a variety of normalcy models that are probabilistic and/or rule-based. Unfortunately, they can be limited in their ability to model the subtlety and complexity of multiple vessel types and their spatio-temporal events, yet their representation is needed to accurately detect anomalies in maritime scenarios. To address these limitations, we apply plan recognition algorithms for maritime anomaly detection. In particular, we examine hierarchical task network (HTN) and case-based algorithms for plan recognition, which detect anomalies by generating expected behaviors for use as a basis for threat detection. We compare their performance with a behavior recognition algorithm on simulated riverine maritime traffic. On a set of simulated maritime scenarios, these plan recognition algorithms outperformed the behavior recognition algorithm, except for one reactive behavior task in which the inverse occurred. Furthermore, our case-based plan recognizer outperformed our HTN algorithm. On the short-term reactive planning scenarios the plan recognition algorithms outperformed the behavior recognition algorithm on routine plan following. However, they are significantly outperformed on the anomalous scenarios.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

that an agent controlling a small boat or ferry can sense the environment, maintain beliefs about the world state, and derive a plan (i.e., a series of actions, such as *turn, cruise,* and *ram*) to accomplish a given goal (e.g., destroy a Navy vessel). We assume the recognizer has access to a library of plans (e.g., create a terrorist event, catch fish, or ferry passengers). It observes the world state and the actions of the agent to predict the next or some future activity. The recognizer compares the observations resulting from those actions with the observations expected from executing its predicted activity. If these observations are *consistent* with one or more of the plans from the library, it repeats its inference process. Otherwise, it flags the vessel as a potential threat or anomaly and presents the information to a decision-maker along with an explanation. In the following subsections we describe plan recognition algorithms that use hierarchical task networks (HTN) or case-based reasoning (CBR) methods to implement these steps.
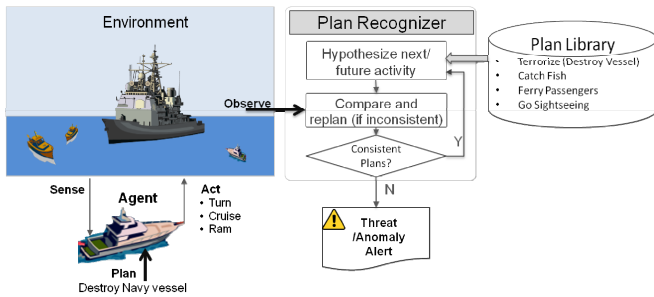

Figure 1. Consistency based approach to maritime plan recognition

A. HTN Plan Recognition

HTN planners operate by recursively decomposing high-level tasks into lower level components. This decomposition process results in a series of low-level actions or primitive tasks, constituting a plan [9]. We adapted an HTN planner for consistency-based plan recognition.

We begin by providing the HTN Recognizer with access to all the routine goals or intents that an agent could follow. For each intention, the Recognizer generates a plan comprising a set of activities that the agent would perform if it tried to pursue it. Each plan is a *travel path* comprising a list of waypoints (i.e., locations the vessel would travel through) together with its *expected behavior* on the travel segments (i.e., the path between two consecutive waypoints). We characterize an agent's behavior on a travel segment by its *speed* and the activity label (e.g., *cruising*, *debarking*, *docking*, or *ramming*). As shown in Figure 1, the Recognizer continuously observes the agent vessel's actions and compares them with the expected actions in all the active (still valid) plans for consistency (see Table 1 for criteria). It retains all the consistent plans and discards all inconsistent plans.

TABLE 1. CONSISTENCY CHECK CRITERIA

Feature	Value constraint
Vessel-speed	Within specified threshold
Vessel-track	Within specified boundary
Bearing-to-waypoint	Within specified threshold
Bearing-changes	Within specified threshold
Distance-to-Waypoint	Select next waypoint

In a dynamic maritime environment, it is necessary to account for plan deviations in the face of unexpected situations. For example, a vessel may temporarily change its course to avoid a collision with another oncoming vessel. To account for such local variations, when the observed and expected activities deviate, the Recognizer *replans* in an attempt to explain the deviation. First, it updates the active plan based on the current state and evaluates it for consistency. If the consistency check fails, it reattempts to find a consistent plan using a preset number of historical states. If the updated plan remains inconsistent, it is removed from future consistency evaluations. Therefore, at the start of a plan recognition process, the Recognizer experiences its largest computational load as all the plans must be evaluated for consistency. Its search load reduces as inconsistent plans are discarded. By default, if no plan remains consistent then the situation is deemed anomalous.

B. Case-Based Planning

Case-based reasoning (CBR) is a process for solving problems by retrieving and reusing solutions from problems that are similar to the problem at hand [10]. A CBR system relies on a memory of problem-solution pairs called *cases*. These cases can be automatically learned from observations or hand-engineered, depending on the application. The CBR process includes an algorithm for case retrieval, which involves case similarity assessment.

We adopt CBR for consistency-based plan recognition as follows. We manually engineered two case bases that cover selected scenarios of interest. The first case base, called the *Primary Case Base* (PCB), covers the entire travel path of an agent. The second case base, called the *Secondary Case Base* (SCB), only covers the local variations (e.g., path deviations) to account for collision avoidance among two vessels. The reason for using two case bases is that, unlike the HTN algorithm, which uses a single representation for plans, the CBR algorithm cannot effectively account for unforeseen local deviations from the plan using a single case representation.

TABLE 2. PRIMARY CASE REPRESENTATION

Feature	Description
Vessel-Type	Type of vessel performing an activity
Travel-Path <[location, <Map [speed, activity]>]>	Travel path comprising waypoints. Each travel segment points to a hashmap of speed to activity label

TABLE 3. SECONDARY CASE REPRESENTATION

Feature	Description
Vessels-(Ref, Approaching)	The reference and approaching vessel types
Relative-bearing	Their relative bearing
Primary-case-pointer	Pointer to a primary case
Speed (Ref, Approaching)	The speeds of the two vessels
Avoidance-Actions	Ordered list of actions that the reference vessel will execute

Tables 2 and 3 display case representations for the PCB and SCB. The *vessel type* and its *travel path* features, as defined above for our HTN plan recognizer, are used to represent the PCB cases. In addition to the vessel's speed and its activity

label, the case-based algorithm also uses a Boolean feature (not shown in Table 2) to indicate whether the travel segment is the last one of the entire path.

The SCB cases are represented by the two vessel types, their relative bearing, a pointer to the primary case, the speeds of the reference and the approaching vessels, and the avoidance actions that the reference vessel executes. The relative bearing feature measures how the two vessels' bearings relate, and is calculated as follows:

$$rb_{x,y} = rb_x + rb_y$$

where rb_x is the angle between the velocity vector of vessel x and the vector pointing from vessel x to vessel y. The value of $rb_{x,y}$ ranges from 0 to 360. A value of 0 implies the vessels are moving directly at each other and a value of 360 implies they are moving directly away from each other.
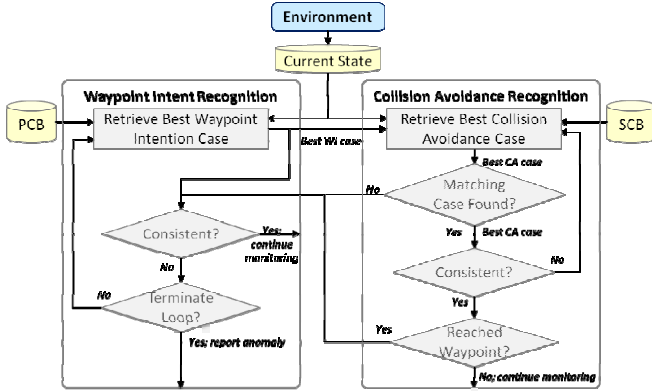

Figure 2. Case-based plan recognition algorithm

Plan recognition using the case bases involves two concurrent CBR tasks: waypoint intent recognition and collision avoidance recognition (see Figure 2). Waypoint intent recognition proceeds as follows. Given the current state involving an agent vessel v, the best matching case c_{pcb} is retrieved from the PCB and the criteria shown in Table 1 are used to check consistency with c_{pcb}. This is repeated until a consistent case is found or the loop terminates, whereupon v is marked as an anomalous situation (e.g., to report to a user). For the collision avoidance recognition task, if a case c_{scb} matching the current situation is found in the SCB, then the waypoint recognition task is suspended and the collision avoidance task becomes active. This latter task ends when v's behavior becomes inconsistent with c_{scb}'s expected avoidance actions or it reaches the next waypoint in c_{pcb}, at which time control passes back to the waypoint intent recognition task.

IV. EVALUATION

A. Objectives

Our objectives were as follows:

1. Compare the activity labeling performance of the two plan recognition algorithms against each other and versus a selected behavior recognition algorithm (see Section 4.D) on simulated long-term and reactive scenarios. Our two plan recognition algorithms should outperform the behavior recognition algorithm on the long-term activities

because the former leverage additional domain knowledge (i.e., expected paths).

2. Compare the lead-time to recognize a change in activity of the three algorithms. We conjecture that the plan recognizers will outperform the behavior recognizer because they can detect more subtle anomalies (e.g., moving to an unexpected dock).

B. Test Scenarios

We developed four simulated test scenarios that take place on a model of the Potomac River in Washington, DC. Besides the *real* docks of National Harbor (NH) and Old Town (OT), we introduce a *fictitious* dock called Reagan Airport Dock (RAD) to evaluate the impact of long-term goals and intentions. We developed the first two scenarios to examine the performance on long-term intent following, while the others allow us to explore performance on short-term reactive events. The descriptions of these scenarios (see Figure 3) are as follows:
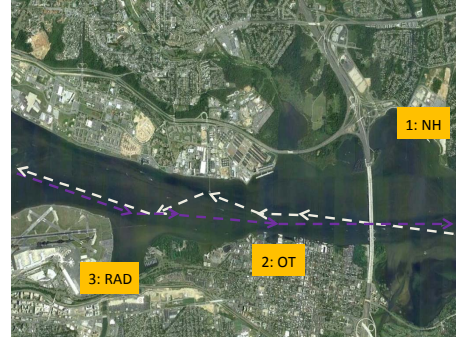

Figure 3. Scenario of two motorboats avoiding a head on collision

1. *Routine Ferry* (between NH to RAD): In this scenario the ferry travels between NH and RAD each hour. The paths between the two docks are similar until they reach the dock near OT, where they diverge.
2. *Non-Routine Ferry* (to OT from RAD while ferrying between NH and RAD): This is an anomalous counterpart to scenario #1. When the ferry reaches RAD, instead of returning to NH, it ferries to OT and back.
3. *Collision Avoidance*: This is a head-on collision avoidance scenario among two motorboats. They follow collision avoidance regulations, turning to starboard to avoid a head-on collision when they approach each other.
4. *Traffic Violation*: This is an anomalous counterpart to scenario #3, where one of the motorboats intentionally disobeys collision regulations and continues on its path, thereby simulating a potentially dangerous and threatening situation to the other motorboat.

We encoded these scenarios in a maritime traffic simulation environment, which we present next.

C. Test Environment for Maritime Traffic Simulations

We modified a Navy simulation tool, named the *Tactical Actions Officer* (TAO) *Sandbox*, to conduct our evaluation. The TAO Sandbox is used to train Tactical Action Officers on deployment and management of Navy assets for anti-submarine warfare and related missions [11]. We modified it

as follows: (1) its agents are controlled by our HTN algorithm, and (2) it visually displays the behavior of our plan and behavior recognition algorithms in the Potomac River map.

First, we used an HTN planner, namely SHOP2$_{PDDL+}$ [12] to *generate* the plans used by all the simulated agents controlling the vessels in our scenarios (i.e., motorboats and ferries). Among HTN planners, we selected SHOP2$_{PDDL+}$ because it can be used to generate behaviors in our continuous planning domain. It uses a *wait* action to represent durative activities. For our scenarios, we encoded a set of SHOP2$_{PDDL+}$ methods (e.g., for ferrying from one dock to another) that allows it to generates plans such that, given a scenario's start state and goal state, the resulting plan can be given to and executed by the TAO Sandbox. To respond to a dynamic situation (e.g., collision avoidance situations), we added an action that can be used to perform conditional planning; it allows an agent to monitor the TAO Sandbox environment for any specified situations and, if matched, triggers a replan. The revised plan's actions are then sent to the TAO Sandbox simulator for execution.

Second, we used Google Earth to accurately scale (from pixels to nautical miles) and depict the Potomac River area, including the NH, OT, and RAD ports (see Figure 3). We modeled the waypoints for the four scenarios, their associated ferry schedules, and boat transit speeds by observing actual maritime traffic on the Potomac River. We specified the waypoints by considering the upstream and downstream traffic lanes that are marked with buoys. We added random variations to the scenario instances to simulate realistic maritime conditions and path variations. We included up to four concurrent agent-controlled vessels in a scenario to control for extraneous vessel interactions. Finally, we ran ten instances of each scenario and collected our observations.

D. Threat Detection Algorithms

We implemented two *plan recognition* algorithms for comparison with the best-performing *behavior recognition* algorithm from our previous studies [3].

1. *HTN Plan Recognizer* (**HTN-PR**): This implements the HTN plan recognition algorithm described in Section 3. The Recognizer is connected to and observes the TAO Sandbox environment (e.g., the vessels' activities). It uses the same planning model given to SHOP2$_{PDDL+}$ (see Section 4.C). Also, for each of the four scenarios, we provide this planner with the set of possible goal states (e.g., ferry passengers to a particular dock).

    For dynamic situations, the Recognizer periodically replans to mimic the conditional replans that the agent might execute. It uses these to detect anomalies by noticing when the actual and expected actions deviate during a specified time frame (e.g., as in scenario #4, where the agent-controlled motorboat deliberately disregards collision avoidance regulations).

2. *CB Plan Recognizer* (**CB-PR**): This implements the case-based plan recognition algorithm described in Section 3. We manually populated the PCB with 4 cases (one per vessel instance) and the SCB with 2 cases (one per

motorboat). Also, for dynamic replanning situations, like HTN-PR, if the SCB retrieves a case c_{scb} and the agent does not take any of c_{scb}'s actions within a prespecified time, then the activity is marked as anomalous.

3. *MLN Behavior Recognizer* (**MLN-BR**): This algorithm is based on our recent work on probabilistic graphical models [3], where we showed that MLNs performed well on maritime threat detection tasks. MLNs combine first-order logic with a probabilistic interpretation to represent expert domain knowledge [13]. We used Alchemy [14] to implement MLN-BR. It interfaces with the TAO Sandbox to obtain a representation of the world state as a set of first-order logic clauses. MLN-BR performs MAP inference for each agent to recognize its behavior. We encoded knowledge for MLN-BR to identify the following activities: Debarking, Cruising, Docking, and Docked. This knowledge includes five rules that encode discretized values of a vessel's speed, proximity to a dock, and its bearing relative to docks. In addition, to handle the collision avoidance scenarios, we encoded seven more logic rules. These rules involve additional features such as the possibility of head-on collision, and whether the agent is on its intended path to its next waypoint. Finally, the MLN is given a series of rules regarding the vessel type (i.e., motorboat or ferry). For example, the rule:

   ```
   5 !Speed(v,0,t) => !Action(Docked,v,t)
   ```

specifies that, if the speed of a vessel *v* is not 0 at time *t*, then *v* is not Docked at time *t*, and this rule has a weight of 5. Here a weight is a measure of the strength of a constraint for a given world. A world's likelihood is lower if it violates a constraint with a high weight.

E. Metrics

We used the following measures to compare the algorithms:

1. *Activity Labeling Accuracy*: This is the number of correctly predicted labels for an event type divided by the total number of that type of event. Our scenarios included the following event labels: *Avoiding Collision*, *Cruising*, *Debarking*, *Docked*, *Docking*, and *Anomaly*. We computed the accuracy for each label type and the accuracy over all the label types.

2. *Anomaly Labeling Latency*: Each time step during the course of an activity represents an event, which the algorithms label. The ability to label an anomalous activity as soon as it starts is desirable for the algorithms. Therefore, we measure the time elapsed between the onset of an anomaly and the time it takes for an algorithm to correctly detect it. Smaller time lags imply a better performing algorithm.

F. Results

Table 4 displays the accuracy of three algorithms on our four scenarios. In Scenarios 1 and 2, which pertain to longer-term behaviors, the plan recognition algorithms significantly outperform MLN-BR. For instance, in the Routine Ferry Scenario (#1), HTN-PR and CB-PR recorded higher

accuracies than MLN-BR (0.859, 0.839 vs. 0.776). However, the performance difference among the two plan recognition algorithms is small. In Scenario #2, concerning non-routine ferries, HTN-PR and CB-PR again significantly outperform MLN-BR (0.928, 0.938 vs. 0.564), which cannot detect anomalies because it has no knowledge of the intended routes. Furthermore, CB-PR outperforms HTN-PR overall and on anomaly detection for this scenario (0.984 vs. 0.931).

Performance on the reactive scenarios is mixed. In particular, in Scenario #3, one of the plan recognition algorithms (CB-PR) significantly outperforms the behavior recognizer (MLN-PR) on the avoidance activity (1.0 vs. 0.773). However, in Scenario #4, MLN-BR outperforms both the plan recognition algorithms on anomaly detection (0.881, 0.905 vs. 0.971), although the differences were not significant. We believe this occurred because MLN rules generalize better in a reactive situation, a conjecture that we will explore in our future work. CB-PR outperforms HTN-PR in Scenario 4, and predicts all avoidance actions correctly on Scenario 3.

TABLE 4 . AVERAGE ACCURACY PER ACTIVITY (10 RUNS)

	Overall	Cruise	Debark	Docked	Dock	Avoid	Anomaly
Scenario 1. Routine Ferry							
HTN-PR	**0.859**	0.936	**0.796**	0.471	**0.947**	na	na
CB-PR	0.839	**0.946**	0.788	**0.871**	0.778	na	na
MLN-BR	0.776	0.828	0.646	0.714	0.938	na	na
Scenario 2. Non-Routine Ferry							
HTN-PR	0.928	0.931	**0.940**	0.775	0.929	na	0.931
CB-PR	**0.938**	**0.966**	0.866	**1.000**	0.929	na	**0.984**
MLN-BR	0.564	0.862	0.673	0.750	0.929	na	0.000
Scenario 3. Collision Avoidance							
HTN-PR	0.825	**0.863**	na	na	na	0.683	na
CB-PR	**0.885**	0.854	na	na	na	**1.000**	na
MLN-BR	0.518	0.452	na	na	na	0.773	na
Scenario 4. Traffic Violation							
HTN-PR	0.873	**0.907**	na	na	na	0.681	0.881
CB-PR	0.879	0.839	na	na	na	1.000	0.905
MLN-BR	**0.908**	**0.907**	na	na	na	0.728	**0.971**

Therefore, this supports our first conjecture that these plan recognizers can recognize some longer-term behaviors more accurately than this behavior recognition algorithm.

Table 5 displays average anomaly labeling latencies for the anomaly recognition scenarios (i.e., #2, and #4). For longer-term intent violations (Scenario 2), CB-PR substantially outperforms HTN-PR (27 vs. 118 seconds). Thus, CB-PR is a more accurate activity labeler and is also faster at detecting an anomalous situation. In contrast, MLN-BR's time is infinite since it does not detect any anomaly in this scenario. However, it performs much better in the reactive Scenario 4. For instance, it detects collision avoidance anomalies after an average of 10.7 seconds after its onset compared to at least a minute (i.e., 64.7 seconds) for the other algorithms.

TABLE 5. ANOMALY LABELING LATENCY (SECONDS)

Scenario	#2	#4
HTN-PR	118.0	90.4
CB-PR	**27.0**	64.7
MLN-BR	*	**10.7**

V. DISCUSSION

Many methods have been investigated for plan and behavior recognition (e.g., [15][16][17]). They differ along many dimensions, such as their agent relation (e.g., keyhole, intention, adversarial), what they infer (e.g., action, plan, goal), model representation type (e.g., decision theoretic, probabilistic), whether they are generative or discriminative, their model family (e.g., Bayesian, MDP, random field), and how they detect anomalies. These have been applied to a plethora of tasks, such as adversarial strategy detection, user modeling, human activity recognition, and threat detection, which is our focus. However, little published research exists on applying plan recognition techniques for *maritime* threat recognition, although Nicolescu and her colleagues have studied this topic in recent years, including the application of hidden Markov models (HMMs) and spreading activation techniques for detecting potentially deceptive behavior [18].

Our own related work has most recently focused on applying probabilistic graphical models to synthetic and real-world threat recognition scenarios [3]. We found that Markov logic networks possess some advantages in comparison with HMMs and conditional random fields, though they require substantial care to design and time to train. Our experience with these and simpler approaches for threat recognition [19] led us to consider knowledge-intensive plan recognition approaches that can provide additional constraints to the prediction task. From our prior work and this investigation, our observation is that plan recognition algorithms can be preferable to behavior recognition algorithms in scenarios with longer-term anomalies that are detectable using background knowledge, though the inverse can be true for reactive scenarios. This suggests that a hybrid approach may be useful for maritime threat detection, which we will investigate as part of our future research.

VI. CONCLUSION

Existing algorithms for anomaly and threat detection predominantly use statistical normalcy models. Although these are somewhat successful in thwarting short-term reactive changes, they cannot detect subtle but critical violations in routine plans. To address this issue, we developed plan recognition algorithms to detect longer-term and reactive violations of expected vessel behaviors in a simulated maritime environment. In particular, we developed two consistency-based plan recognition algorithms based on HTN and case-based techniques. We evaluated them on a set of four scenarios and found evidence that they significantly outperform a behavior recognition algorithm on longer-term plan recognition tasks.

To our knowledge, this is the first published evaluation of plan recognition algorithms for maritime threat detection

tasks. This is an initial study; further research is needed to better understand the pros and cons of these algorithms. For example, our evaluation was limited in the types of scenarios and the number of concurrent agents active in a scenario. In our future work, we will extend our evaluation to scenarios of coordinated small boat attacks, denser traffic conditions, and schedule violations to investigate the generality of our approach. Also, although we considered stochastic variations in our scenarios, they were limited and our consistency-based approaches were not stochastic. Therefore, we will examine probabilistic approaches to plan recognition in our future research. Finally, we assumed that a library of HTN plans is available to our system at the outset. However, in practice, they will need to be acquired from subject matter experts or learned from observations. We plan to explore other hierarchical and case-based learning algorithms to automatically acquire such models (e.g., [20][21]).

REFERENCES

[1] Lipton, A.J., Heartwell, C.H., Haering, N., & Madden, D. (2002). Critical asset protection, perimeter monitoring, and threat detection using automated video surveillance. In *Proceedings of the Thirty-Sixth Annual International Carnahan Conference on Security Technology.* Atlantic City, NJ: IEEE Press.

[2] RemoteReality (2011). OmniAlert360. Retrieved on 20 November 2011 from [http://www.remotereality.com].

[3] Auslander, B., Gupta, K.M., & Aha, D.W. (2012). Maritime threat detection using probabilistic graphical models. In *Proceedings of the Twenty-Fifth International FLAIRS Conference.* Marco Island, FL: AAAI Press.

[4] Cook, D.J., Augusto, J.C., & Jakkula, V.R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, **5**, 277-298.

[5] Jarvis, P.A., Lunt, T.F., & Myers, K.L. (2004). Identifying terrorist activity with AI recognition technology. *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (pp. 858-863). San Jose, CA: AAAI Press.

[6] Genesereth, M.R. (1984). The use of design descriptions in automated diagnosis. *Artificial Intelligence,* **24***,* 411-436.

[7] Schmidt, C.F., Sridharan, N.S., & Goodson, J.L. (1979). The plan recognition problem: An intersection of artificial intelligence and psychology, *Artificial Intelligence*, **10**(1), 45-83.

[8] Kautz, H. (1987). A formal theory of plan recognition. Doctoral Dissertation: Department of Computer Science, University of Rochester, NY.

[9] Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, **20,** 379-404.

[10] Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications,* **7**, 39-59.

[11] Auslander, B., Molineaux, M., Aha, D.W., Munro, A., & Pizzini, Q. (2009). *Towards research on goal reasoning with the TAO Sandbox.* (Technical Note AIC-09-155). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research on Artificial Intelligence.

[12] Molineaux, M., Klenk, M., & Aha, D.W. (2010). Planning in dynamic environments: Extending HTNs with nonlinear continuous effects. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

[13] Domingos, P., & Lowd, D. (2009). *Markov logic: An interface layer for AI.* Morgan & Claypool.

[14] Alchemy (2011). Alchemy ─ Open source AI. [http://alchemy.cs.washington.edu]

[15] Kautz, H., & Allen, J.F. (1986). Generalized plan recognition. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 32–38). Philadelphia, PA: AAAI Press.

[16] Ramirez, M., & Geffner, H. (2009). Plan recognition as planning. *Proceedings Twenty First International Conference on Artificial Intelligence* (pp.1778-1783). Pasadena, CA: AAAI Press.

[17] Goldman, R.P., Geib, C.W., Kautz, H., & Asfour, T. (Eds.) (2011). Plan recognition (Report from Dagstuhl Seminar 11141). [drops.dagstuhl.de/opus/volltexte/2011/3195/pdf/dagrep_v001_i004_p001_s11141.pdf]

[18] Nicolescu, M., Nicolescu, M., & Louis, S. (2012). Context-based intent understanding for autonomous systems in naval and collaborative robotics applications. *Presentation at the ONR Cognitive Science and Human Robot Interaction Program Review*, June 2012.

[19] Auslander, B., Gupta, K.M., & Aha, D.W. (2011). Comparative evaluation of anomaly detection algorithms for local maritime video surveillance. In *Proceedings of the Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X.* Orlando, FL: SPIE.

[20] Geib, C.W., & Goldman, R.P. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, **173**(11), 1101-1132.

[21] Kerkez, B. & Cox, M T 2003. Incremental case-based plan recognition with local predictions. *International Journal on Artificial Intelligence Tools: Architectures, languages, algorithms*, **12**(4), 413-464.