Neural Networks 22 (2009) 316–325

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2009 Special Issue

Coordinated machine learning and decision support for situation awareness

N.G. Brannon [a], J.E. Seiffertt [b,*], T.J. Draelos [c], D.C. Wunsch II [b]

[a] Reliability Assessment and Human Systems Integration Department, Sandia National Laboratories, Albuquerque, NM 87185, USA[1]
[b] Applied Computational Intelligence Laboratory, Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA
[c] Cryptography and Information Systems Surety Department, Sandia National Laboratories, Albuquerque, NM 87185, USA

ARTICLE INFO

Article history:
Received 24 January 2009
Received in revised form 6 March 2009
Accepted 13 March 2009

Keywords:
Neural networks
Situation awareness
Reinforcement learning
Adaptive resonance theory

ABSTRACT

Domains such as force protection require an effective decision maker to maintain a high level of situation awareness. A system that combines humans with neural networks is a desirable approach. Furthermore, it is advantageous for the calculation engine to operate in three learning modes: supervised for initial training and known updating, reinforcement for online operational improvement, and unsupervised in the absence of all external signaling. An Adaptive Resonance Theory based architecture capable of seamlessly switching among the three types of learning is discussed that can be used to help optimize the decision making of a human operator in such a scenario. This is followed by a situation assessment module.

1. Introduction

Modern information sources to support decisions in domains such as force protection are diverse. Ground, air, and space based sensors are continuing to increase in capability. Information fusion algorithms can help integrate a variety of sensor data into meaningful forms (Hall & Llinas, 1997). Applications with a complex assortment of data continue to challenge machine learning approaches to information fusion, which normally utilize a single type of learning algorithm and therefore limit the use of all available data (Brannon,Conrad,Draelos,Seiffert & Wunsch, 2007). Our approach coordinates multiple learning mechanisms to accommodate environments where ground-truth and feedback may not be consistently available and it uses Adaptive Resonance Theory (ART) based networks which are based on understanding cognition. This ties the work into other such computational architectures seeking not only the solution of engineering problems but also understand the function of the brain and mind (Werbos, 2009; Perlovsky, 2009).

1.1. Machine learning

Machine learning involves programming computers to optimize a performance criterion using example data or past experience (Alpaydin, 2004). Artificial neural networks are commonly used in machine learning and utilize supervised, unsupervised, and reinforcement learning approaches to achieve predictive properties based on example (training) data. Unsupervised learning (clustering) can be effective when ground-truth is not available with a dataset. Supervised learning (learning with a teacher) provides a means to use experience (examples with ground-truth) to correctly classify yet unseen situations. Reinforcement learning offers promise for machine learning in difficult learning environments by taking advantage of feedback about the performance of a system. The challenge addressed by the current work is to coordinate all of these learning mechanisms and utilize the appropriate one based only on available information, not human intervention.

Neural nets offer an excellent assortment of high-performance, low-cost, distributed processing options. In particular, they can be embedded into appropriate sensors for operation at the lowest levels of information fusion with effective, but low-complexity designs. At the highest levels of information fusion and situation assessment, reinforcement learning can be used with a human in the loop to provide operational feedback. Dealing with multiple sensor modalities and extracting meaningful information from massive datasets is a natural fit for these adaptive methods. Although neural networks have been applied to sensor fusion, their use in situation awareness has been limited, possibly because of the lack of rich training data for this problem.

Automated (computational) information fusion continues to suffer from very specific, ad hoc solutions (i.e., there appears to be very little general-purpose technology to apply to this problem) (Kokar, Tomasik, & Weyman, 2004). For many applications, there is also a dearth of data to use for training a computational

* Corresponding author.
E-mail address: seiffertt@ieee.org (J.E. Seiffertt).

[1] Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.03.013

engine. This reveals a challenge for the application of machine learning techniques, which are data-driven and require training—via supervised, unsupervised, or reinforcement learning. On the other hand, because they are data-driven, the advantage of machine learning techniques is that they can learn solutions to problems that are difficult for humans to codify with explicit rules or models. In other words, they can represent rules/decisions that are implicit in the training data.

1.2. Information fusion

The fusion of information has been likened to the ability of animals to utilize multiple senses to derive a better understanding of a situation (Hall & Llinas, 1997). For example, one may hear a noise and, based on the sound pressure discrepancy between each ear, localize the area of the sound source. Vision can then be used to further define and understand the source of the sound. The analogy is helpful because fusion, and more generally situation assessment, is a process rather than simply a discrete event. The process leads one from raw data to understanding and actionable knowledge. Fusion can occur over various information (sensor) modalities, over geographic space, and over time.

The sources of information potentially available to decision makers continue to expand in depth and breadth. Sensor capabilities in particular are maturing rapidly, but a valid concern is that the pace of sensor development has not necessarily been consistent with advances in human effectiveness which the sensors must ultimately support (Paul, 2001). Fusion algorithms will better support human-in-the-loop system effectiveness when the decision maker is a central and balanced design element.

2. Approach

2.1. System architecture

The design of the computational engine for information fusion and situation awareness takes advantage of the diverse utility of neural networks and integrates elements of supervised, unsupervised, and reinforcement learning. The design not only advances machine learning research, it addresses the need of situation awareness and human-in-the-loop decision support as well.

Key design attributes of our system include accepting various inputs such as binary, categorical, and real-valued data. With respect to situation assessment outputs, attributes include confidence levels as well as evidence in support or against the assessment. In the context of missing or noisy inputs, the system exhibits graceful performance degradation.

In order to address the desired design attributes of our situation awareness system, neural networks are employed for information fusion, followed by a situation assessment module. ARTMAP is based on Adaptive Resonance Theory (ART), a widely implemented approach to modeling the learning capabilities of the brain (Carpenter & Grossberg, 1988). Architectures based on ART have been used successfully in a variety of areas requiring a self-organizing pattern recognition neural network. The basic ART element supports unsupervised learning and binary inputs. Fuzzy ART is an extension to accommodate categorical and real-valued inputs. ARTMAP supports supervised learning and can accommodate real-valued inputs using fuzzy logic (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992). ARTMAP can also support reinforcement learning, for example, by adding a mechanism to implement actor–critic methods. Coordinated ARTMAP (CARTMAP) is the name given to the current approach and involves the integration of all three learning mechanisms in the same architecture.

The situation assessment module receives state information from the information fusion module and possibly other sources and outputs a threat assessment or action to be taken.

2.2. Information fusion engine

Intelligent creatures exhibit an ability to switch seamlessly among unsupervised, supervised, and reinforcement learning as the needs arise. However, machine learning architectures, including artificial neural networks, have not yet achieved this goal. The current research contends that it is advantageous to develop this capability in a computational framework and that the ART architecture is an excellent choice for such an implementation. In the following text, we motivate, design, and give examples of this capability.

A well-designed sensor fusion algorithm, like an intelligent creature, can make informed use of all three types of learning on the dataset given. Certain information fusion paths may be pretrained prior to deployment, thus granting the human operators license to verify that the most obvious sensor patterns will be classified successfully. During operation, a reinforcement signal provided either by the environment or by the human operator acting off of the fusion algorithm's recommendations can adjust the current adaptive weight profile to curtail or retrain a faulty clustering (negative reinforcement) or to promote successful clustering (positive reinforcement) in the ART algorithm. Finally, in the absence of any external signal, the algorithm will learn in an unsupervised manner, comparing current inputs to what it already knows.

Our core algorithm—ARTMAP—already handles both unsupervised and supervised learning problems. To augment this, we need to optimize

$$J(s) = r(s) + \gamma \sum_{s'} P(s', a) J(s', a). \tag{1}$$

This is the discounted expected reward optimality criterion, a form of the Bellman equation of dynamic programming. In this equation, s represents the current state of the system, $r(s)$ represents the current reward, a the action to be taken, $J(s)$ represents the current value of a given state, s' signifies the next states, $P(s', a)$ is the transition probability matrix for the system's evolution, and a discount factor, γ, is applied to future rewards. This equation is to be maximized over all actions.

The Bellman equation (1) states that the current value of a state is equal to the immediate reward of taking an action plus the discounted future reward that accrues from that state. Other optimality criteria are possible to account for infinite horizon or non-discounted models. The task at hand is to solve this equation given an appropriate reinforcement signal.

Sutton and Barto (1998) discuss a wide variety of solution methods for these problems. Our algorithm combines ART with Q-learning. The Q-learning algorithm iteratively updates the value of each state–action pair. The appropriate modification is calculated based on the difference between the current and realized valuations, when maximized over all possible next actions. This is a key concept that establishes the foundation for the more advanced techniques discussed in the following paragraphs.

The Q-learning algorithm utilizes a lookup table to store the Q-values for each state–action pair. As the scale of the simulation grows, the amount of memory required to catalogue these values can grow at a burdensome rate. A more computationally intensive but less memory-demanding version, called Heuristic Dynamic Programming, uses function approximators in place of the table (Werbos, 1990). However, for our purposes in this architecture, the Q-learning approach will suffice.

With the ARTMAP unit taking the place of the Actor in the actor–critic implementation, the Coordinated ARTMAP (CARTMAP) algorithm behaves according to the following steps:

1. Upon receipt of an unsupervised signal, the system uses its exemplar classification scheme (the ART unit) to output an action choice, as usual. No updating of the lookup table will be necessary.
2. When presented with a supervised signal, the internal adaptive weights updates as per our normal ARTMAP rules, and the output action is set equal to the supervised training signal. Furthermore, the values in the lookup table for actions not associated with the supervisory signal are zeroed out.
3. When a reinforcement learning input signal is received, it will be interpreted according to the Q-learning algorithm. The appropriate entry in the lookup table is augmented with the new reinforcement value, and the action selected is the one with the most values accumulated in its column of the table. In our simulations, the values of the parameters delta and gamma are 0 and 1, respectively (Puterman, 1994).

In summary, the information fusion engine accepts raw data from sensors and other information sources and processes/transforms/fuses them into inputs appropriate for the Situation Awareness Assessment engine.

The information fusion system utilizes appropriate elements of its architecture based on the data presented to it. The three ART networks are linked together by an inter-ART module (Associative Memory). One ART unit handles the inputs, another ART unit processes the supervisory (or target) signal, and the other processes the reinforcement signal as an adaptive critic. This architecture is capable of online learning without degrading previous input–target relationships.

There are times when unsupervised learning is satisfactory, such as in the presentation of new input vectors to a pretrained network. Supervised learning is appropriate and desired for initial training on fixed data. However, these two types of learning do not cover every possible complication. There are times when the human operator does not know the correct classification, yet some feedback on the decision can be provided. These situations fall into the reinforcement learning category. One aspect of developing this information fusion engine, therefore, is adding the reinforcement learning capability to the ARTMAP neural network.

3. Application

We designed our situation awareness system to operate in an environment involving distributed sensors and a central collection site for protection of a facility. Information sources in such an environment can include seismic, magnetic, acoustic, passive infrared (PIR), and imaging sensors as well as weather, time/day information, various intelligence information, local/regional/federal threat levels or law enforcement bulletins, and any other information that might be relevant to the security of a particular facility, such as current traffic situations or health issues.

Conditions of interest to force protection decision makers include: no activity, severe weather, unauthorized people or vehicles in certain locations, and certain types of unauthorized vehicles or humans with weapons in any areas. Actions include: doing nothing, identifying the type and location of a moving object (vehicle or human), commands to turn sensors on or off, dispatching forces, and/or notification of higher authorities. The information sources can include binary data, such as motion detection, categorical data, such as the type of day (weekend, holiday, etc.), and real-valued time-series data, such as seismic, acoustic, and magnetic energy levels.

Before being deployed, the system must be pretrained with information the human operator knows about the system. For example, if the data signature of a thunderstorm is easy to demonstrate (due to specific acoustic, magnetic, etc. levels), then that information can be included in the supervised training portion of the system. The information fusion engine will adaptively learn many more data-observation relationships during online operation, but having basic readings pretrained will aid in initial operation.

When an intruder, be it an unauthorized vehicle or a human with a weapon, breaches the sensor range of a protected facility, the triggered sensor data stream into the information fusion engine. The CARTMAP network then maps these data into observations, such as a vehicle heading north at high speed. These pairings represent novel data readings that were not anticipated, which are then categorized via the CARTMAP algorithm in relation to the pretrained data.

The observation is then sent to the situation assessment engine, which follows the partially observable Markov decision process (POMDP) formulation to calculate a probability distribution over the state space. This information represents a confidence level that the system is in any given state. The state with the highest confidence from this calculation represents the system's choice for the current state. All this probability information is then passed to the human operator, who uses this evidence in making a final decision about how to respond to the situation.

Adapting online is an important element of the system and is accomplished through reinforcement signals that can be sent through the system in two ways. First, if the probabilities of each state are too low, so that the human operator would not be able to distinguish the state from simple background noise, then the situation assessment engine may issue a command to gather more information from additional sensors. Second, the human operator may disagree with the system's assessment of the current state. A reinforcement signal is then sent to the information fusion engine and the data-observation mappings will adapt online. Both of these reinforcement signal loops are noted functionally in the block diagram in Fig. 1. This feature of the system allows it to maintain relevance in a changing environment.

The operation of Fig. 1 is as follows. Unsupervised learning occurs using a single ART unit. The cluster that forms is the one which maximizes the signal strength of the input with respect to a match criterion. Many forms of both the signal and the match criterion are in use in various implementations of an ART architecture. Amis and Carpenter (2007) provide default values which work in general scenarios. Supervised learning occurs when the clusters formed by the unsupervised learning unit are given labels through interaction with supervisory inputs. This interaction is mediated by an associative learning field as explained in Amis and Carpenter (2007). This process forces a reset in the input cluster if the label does not match the supervisory signal closely enough. Finally, Reinforcement learning is handled in a similar manner. The RL signal can update the associate weights following the Q-learning explained in Section 2.2.

The CARTMAP algorithm was implemented in Matlab and applied to information fusion in a vehicle tracking scenario that is described in more detail below. ART is at the core of the fusion engine. During offline training, an input pattern is presented to the CARTMAP network and, depending on its similarity to existing category templates, it is either assigned to a current winning category or a new category is created for it. Categories may exist indefinitely without an assigned class. However, if a supervisory signal accompanies the input, the target class is immediately associated with the category. During offline reinforcement learning, an input pattern is presented to the CARTMAP network and a winning category is determined. A reinforcement signal is computed based on the class of the winning category and the ground-truth class. For example, if the category's class matches the ground-truth class, the reinforcement signal is assigned a positive reward; if not, then a penalty is assigned. A range of reinforcement values are assigned based on the quality

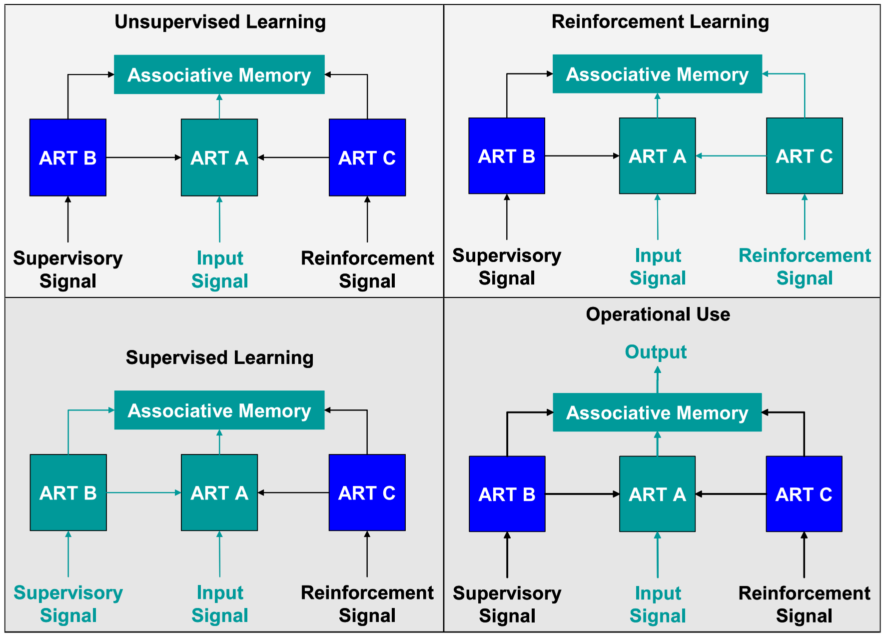

Fig. 1. CARTMAP input and system activity associated with unsupervised learning, supervised learning, reinforcement learning, and standard operational use. Available inputs to the system shown in green are the active elements involved in learning. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the match. A reinforcement lookup table (RLUT) is used to track input pattern's relationship with possible classes. The RLUT stores input patterns and an accumulated reinforcement signal for each possible class. CARTMAP weights are updated according to the following criteria.

1. If no category encodes the input pattern, then a new category is created without a class assignment.
2. If the winning category has an unassigned class, then the RLUT is searched for the input pattern. If the pattern is found in the RLUT, then the reinforcement signal is applied to the class of the winning category and the class with the highest reinforcement is used as the target in supervised learning. If the pattern is not found in the RLUT, then nothing is done to the CARTMAP weights.
3. If the winning category has an assigned class, then this class and reinforcement signal are used by a critic function to determine how to update CARTMAP weights. The RLUT is searched for the input pattern. If the pattern is not found, unsupervised learning is performed and the pattern is added to the RLUT along with the reinforcement signal. If the pattern is found in the RLUT, then the reinforcement signal is applied to the RLUT for the class of the winning category and the class with the highest reinforcement is used as the target in supervised learning.

The decision support graphical user interface (GUI) consists of three screens. The center screen is primarily imagery (i.e., from cameras, photography augmented with graphics, and/or fully synthetic renderings) (see Fig. 2). The second screen displays a log of temporal track data (see Fig. 3). The log reflects temporal features, such as how long ago an unauthorized vehicle breached a sensor field and how soon another track might reach a key threshold (e.g., a fence or different sensor field). The third and most detailed screen provides track detail and assessment bases (see Fig. 4).

The log screen and track detail screen utilize features found in the Tactical Decision Making Under Stress (TADMUS) system (Morrison, Kelly, Moore, & Hutchins, 1997). The TADMUS system had similar motivations to the current research in that more content need to be devoted to supporting an understanding of

Fig. 2. Vehicle tracking scenario map. The dots represent seismic/acoustic sensor nodes. The speed, heading, location, and vehicle type are estimated by independent CARTMAP networks using binary data from all sensor nodes as input.

a given context. In both TADMUS and our situation awareness approach, less emphasis is placed upon evaluating possible courses of action.

The track detail GUI provides typical track parameters such as an object's course and speed, but significant detail is provided with respect to the basis for assessment. Evidence in support and evidence against a given assessment is displayed. The machine learning algorithms share the evidence used to derive assessments with the operator. Such an approach provides greater transparency and allows the operator to interrogate assessments.

For the example scenario of an unauthorized vehicle, the assessment could be a "threat". Evidence in support of such an assessment includes sensor data such as explosives detected, but also local law enforcement data that the license plate returns as a stolen vehicle. Evidence against the assessment could include a relatively slow speed and the use of the vehicle for construction when there has been ongoing construction activity. Alternative

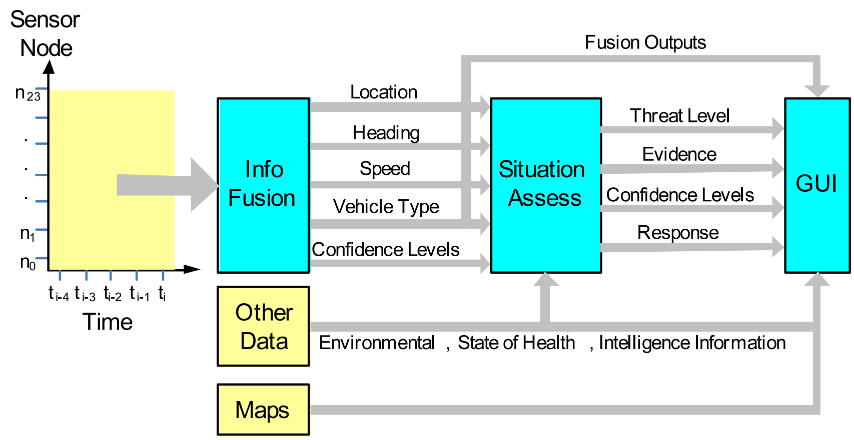

Fig. 3. Force protection experiment using UW vehicle data. Multiple time steps of binary sensor data are used as input to the CARTMAP Information Fusion module. Vehicle information from the Fusion module and other additional data are used as input to the Situation Assessment module, which outputs an actionable information to the user.

assessments are shown along with their respective evidence in support or against.

The operator can investigate various assessments along with corresponding courses of action. For example, a patrol vehicle is in the vicinity of the unauthorized vehicle and could be directed closer to the possible threat. Further, other types of sensors can be activated to generate additional points of reference and work towards higher levels of assessments such as possible intent.

3.1. Vehicle tracking

The situation awareness technology was applied to tracking vehicles in the vicinity of a facility under force protection. A dataset suitable for testing and demonstrating our technology was collected during a DARPA SensIT program in November, 2001 at Twenty-Nine Palms, CA and exists at the University of Wisconsin (UW) (Duarte & Hu, 2004). The dataset consists of raw time-series (acoustic and seismic) and binary detection decisions from 23 sensor nodes distributed along three intersecting roads as one of two vehicles travels along a road. Fig. 2 includes a map illustrating the force protection scenario, with a fence line and an Entry Control Point (ECP) providing protection for a facility on the North Road. The two vehicles used in the scenario are a light armored vehicle (AAV) and a heavier, tracked transport vehicle (DW). A scenario was developed whereby a facility under protection is assumed to exist along one of the roads, and binary sensor data processed by our fusion and situation assessment algorithms are used to inform a human decision maker.

3.2. Analysis

3.2.1. Force protection experiments

In order to demonstrate the capabilities of the situation awareness system, neural networks were trained to perform sensor fusion, a situation assessment formula was constructed/calculated, and a GUI was developed, all to increase the awareness of a human decision maker of the situation around that facility under their protection. The scenario consists of a virtual checkpoint partway up the north road on the way to a sensitive facility with 23 sensor nodes scattered along three intersecting roads. Each sensor node outputs a binary detection decision at fixed time intervals (0.75 s in the original test set). The sensor detections derive from seismic, acoustic, and passive infrared energy levels. The (AAV and DW) vehicles move from one end of a road, through the intersection to the end of another road. The total number of runs is 40, which includes 20 original datasets from the SensIT experiment. An additional twenty runs were created by artificially reversing the direction of the vehicle. This is possible by simply presenting the data in reverse. In other words, the sensor record from the last time step would be presented to the information fusion system first and the first time step would be presented last and so on for all the time steps in the run. It is plausible that the information is accurately represented in these runs since the data is binary decisions and the ground is relatively flat so that the engine speed and noise is presumably similar in both directions.

The primary piece of information that a decision maker wants to know is the current threat level around his facility. The threat level is a function of the location, speed, heading, and type of vehicle detected by the sensor array and other variables that are independent of the sensor array, such as: Department of Homeland Security (DHS) advisory level, wind speed, average batter level of the sensors, time of day, and day of week.

The system used to produce the threat level is illustrated in Fig. 3. The system consists of three modules: (1) Information Fusion, (2) Situation Assessment, and (3) a Graphical User Interface (GUI) focused on human decision makers in force protection applications. Multiple time steps of binary sensor data serve as input to the Information Fusion module, which implements the CARTMAP algorithm. This introduces an element of relative time, which is a necessary component in estimating speed and heading. The output from the Fusion module consists of vehicle type, speed, location, and heading, each with a corresponding confidence level, and will serve as input to the Situation Assessment module. This module consists of rules that represent the conditions under which a Threat is defined. The output of the assessment module will feed the graphical user interface (GUI) with a threat level (low, medium, high), an associated confidence level, a suggested response, and evidence in support of or against its output. The GUI will also have access to the output from the fusion module, maps, and other available data, such as time, date, and environmental data. All the elements of the situation awareness system were implemented in Matlab and tested with the vehicle tracking data from UW in a force protection scenario just described.

3.2.2. Results of training the fusion module

The fusion module consists of four different CARTMAP networks, one for each fusion output (location, heading, speed, and vehicle type). The output of a network will be of a categorical type or class except for the confidence levels, which will be a real number. Table 1 presents the classes for each information fusion network. Note that for each network, if the input is all zeros, the

Table 1
Information fusion output classes for the four CARTMAP networks (Vehicle type, location, heading, and speed).

Vehicle type classes	Location classes	Heading classes	Speed classes
0: zero input	0: zero input	0: zero input	0: zero input
1: AAV	1: West road	11: N	1: <10 km/hr
2: DW	2: North road	14: NE	2: 10–20 km/hr
	3: East road	13: E	3: 20–30 km/hr
	4: Intersection	8: SE	4: 30–40 km/hr
		4: S	5: 40–50 km/hr
		1: SW	6: 50–60 km/hr
		2: W	7: 60–70 km/hr
		7: NW	8: 70–80 km/hr
			9: 80–90 km/hr
			10: >90 km/hr

Table 2
Distribution of vehicle runs used to experiment with different learning modes. Experiments 1/2/3 and 4/5/6 used the same data, but uses learning modes in a different order.

Experiment #	# Supervised runs	# Unsupervised runs	# Reinforcement runs	# Test runs
1 & 4	2	26	0	12
2 & 5	2	13	13	12
3 & 6	2	0	26	12

output will be zero by virtue of a simple fixed rule (i.e., no learning is involved).

Out of the 40 total runs available for the force protection experiments, 70% were used for training and the remainder for testing. Table 2 shows the number of runs used in the six experiments. In real-world applications, it is expected that the amount of supervised training data is limited. In the force protection experiments, only 2 of the 28 training runs are used for supervised learning.

Experiments 1–3 use the same runs as Experiments 4–6, but the order of training is reversed. In Experiments 1–3, supervised learning is conducted first, followed by reinforcement learning, and finally unsupervised learning. Experiments 4–6 use the opposite order of learning, using the data with the least amount of information first and finishing with supervised learning, which utilizes training data with the most amount of information. In this case, one expects the richer datasets and training modes to correct errors and refine the classification performance of previous learning modes.

For each force protection experiment conducted, the same test set was used, consisting of 12 runs with 1755 input/output pairs. The performance (% correct classification) was computed based on this test set. For some sensor modes, such as speed and heading, a classification error may not necessarily indicate poor performance. For example, if the ground-truth heading of a vehicle is North and the fusion module output is Northeast, it would be counted as a classification error even though the output is quite satisfactory. Experiments 1–6 were conducted using various combinations of learning modes for each of the information fusion networks. The best results for each network are presented in Table 3.

In the Classified Correct (%) column of the tables, there are three numbers separated by colons (e.g., 1:2:3). The numbers in position one represent the percentage of test samples that have a target value that exactly matches the output value from a CARTMAP network.

The numbers in the second position represent the percentage of test samples that have a target value that exactly or partially matches the output value from a CARTMAP network. An exact match increments the total number of correct classifications by 1, whereas a partial match increases the number by 0.5. Partial matches are possible only with the Heading and Speed networks, where the class adjacent to the target class is considered a partial match. For example if the target class is N, then a network output of NW or NE would result in a partial match. Note that for the Vehicle Type and Location networks, there are no partial matches so the first and second numbers in the Classified Correct column should be the same.

The numbers in the third position represent correct classification percentages of networks that have had two passes through the training set. During the first pass, the reinforcement lookup table is updated during reinforcement learning. The updated table may be an advantage for second pass unsupervised and reinforcement learning. Correct classification percentages are computed using partial matches. Each network was trained using vigilance parameters that resulted in a reasonable number of categories.

In the next section, a weighted rule for determining the threat level of the situation awareness system is discussed. The rule combines the outputs of the fusion module and environment conditions and its output is categorized into High, Moderate or Low threat based on human judgment. Ground-truth exists for the threat level, so performance of trained fusion networks with specified environmental conditions can be measured. Two environmental conditions are specified: (1) Benign – each environmental condition is set to its lowest value, and (2) Severe – each environmental condition is set to its highest value. For each of the learning modes, the correct classification percentage is measured against ground-truth. The results are given in Table 4.

In practice, if unlabeled data is all that is available, then machine learning is typically not used at all. Machine learning is most often used when some labeled data are available and supervised learning is then used to its maximum extent, while other learning techniques are not employed. The advantage of a using a variety of machine learning techniques is evident in Tables 3 and 4 above, but a single set of networks (possibly a different network for each sensor mode) must be chosen since one cannot generally anticipate the environmental conditions. Table 5 summarizes the performance results of using the best combination

Table 3
The best fusion test results of the four CARTMAP networks. Reinforcement learning followed by supervised learning worked best for estimating vehicle type and location, while supervised learning followed by unsupervised learning, then reinforcement learning worked best for vehicle heading and speed. In the Classified Correct (%) column of the table, there are three numbers separated by colons (e.g., 1:2:3). The numbers in position one represent the percentage of test samples that have a target value that exactly matches the output value from a CARTMAP network.

Sensor mode	Experiment #	Learning mode	Vigilance	# Categories	Classified correct (%)
Vehicle type	6	Reinforcement	0.7	36 : 108	92.6 : 92.6 : 91.7
		Supervised	0.65	44 : 112	92.7 : 92.7 : 91.7
Vehicle location	6	Reinforcement	0.7	22 : 58	96.8 : 96.8 : 98.0
		Supervised	0.65	31 : 61	96.9 : 96.9 : 98.0
Vehicle heading	2	Supervised	0.9	39	68.4 : 69.6 : 69.6
		Reinforcement	0.7	45 : 59	66.6 : 79.9 : 80.3
		Unsupervised	0.7	45 : 59	62.7 : 75.8 : 81.7
Vehicle speed	2	Supervised	0.9	46	72.4 : 79.9 : 79.9
		Reinforcement	0.7	53 : 77	74.3 : 82.1 : 82.0
		Unsupervised	0.7	53 : 77	73.4 : 81.3 : 81.8

Table 4
Best test results of situation assessment threat level performance using a combination of learning modes under benign and severe environmental conditions. Different learning modes for different CARTMAP fusion networks are necessary to produce the best situation assessment results.

Environment condition	Vehicle exp #	Location exp #	Heading exp #	Speed exp #	Reinforcement iterations	Classified correct (%)
Benign	1	2	3	3	1	88.9
Benign	1	3	2	3	2	89.5
Severe	1	2	2	6	1	86.8
Severe	3	2	5	3	2	87.7

Table 5
CARTMAP fusion performance results of using multiple machine learning modes in comparison to supervised learning alone.

Learning approach	Vehicle %	Location %	Heading %	Speed %	Avg. Threat %
SL	81.8	95.6	69.6	79.9	78.5
SL with UL and/or RL	92.7	98.0	81.7	81.9	87.6

of supervised (SL), unsupervised (UL), and reinforcement learning (RL) in comparison to the more common uses of supervised learning alone. Table 6 lists the machine learning approaches used by each CARTMAP network to produce the best situation assessment threat level performance averaged over benign and severe environmental conditions.

An important conclusion drawn from the experimental results is the utilization of multiple training approaches that can take advantage of additional and different data, and produces superior results for situation awareness compared to supervised training alone. The reason performance goes down with UL after SL is that with SL alone, all test patterns get encoded by a labeled category whereas after UL, there are now unlabeled categories that may encode test patterns producing classification errors. Even though these unlabeled categories sometimes lowered the performance, they may eventually add value after subsequent labeling during SL or RL. Unsupervised input patterns that get encoded by existing categories with a class label can contribute to the quality of the category in representing the class in feature space. In addition, since the CARTMAP has access to a reinforcement lookup table (RLUT), if an unlabeled pattern matches a pattern in the RLUT, the corresponding class label from the RLUT can be assigned to the unlabeled pattern. This feature is used during unsupervised learning. Originally, the RLUT is generated from the supervised training data. It expands when new unlabeled patterns are encoded by categories with class labels and the pattern and its label are added to the RLUT.

Results for Experiment 3 (SL followed by RL) reveal a strong relationship between the hints that RL provides and partial matching in scoring the classification performance. When exact classification matches are required, hints may not be good enough. However, if a "close enough" match is sufficient, then improved performance results from RL hints. Even though multiple vigilance values were used in the force protection experiments, it is expected that performance will improve when the vigilance is optimized for the type of fusion mode and the type of learning. It is important in RL to have data representing all classes that a network is designed to classify. If a class is not represented in the data, RL will not be able to establish a label for this class.

Vehicle location is the easiest piece of information to learn with binary sensor data. Location is inherent in the sensors themselves because their position is fixed.

Since 54.4% of the input patters are all zeros, if a correct classification percentage of greater than 54.4% is achieved after UL only, then the reinforcement lookup table is being used to correctly label some patterns. During reinforcement learning, an input pattern is submitted to a network and a reinforcement signal is generated. This signal offers negative or positive feedback on the output of the network. The following steps are taken at this point of reinforcement learning. When a reinforcement signal is received, the RLUT is updated, and SL is performed if the input pattern is found in the RLUT (the action associated with the input pattern with the highest value is used as the target). Unsupervised learning is performed if the input pattern is not found in the RLUT and the reinforcement signal is positive.

Table 6
The combination of learning approaches that produced the best threat level performance. Three different combinations were used for the four different fusion modules (vehicle type, location, heading, and speed).

	Vehicle	Location	Heading	Speed
Learning approach	SL, UL	SL, RL	SL, UL, RL	SL, RL
Reinforcement iterations	1	2	2	2

In general, SL should be used to create as many categories as possible within reason, while subsequent non-supervised training should take advantage of these existing categories and enrich them without corrupting them. The coordination of three machine learning modes therefore offers potential benefit from every sample of data available in an application.

3.2.3. *Situation assessment module*

The situation assessment module takes as input information from the information fusion module and any other information relevant to the evaluation of the situation in the current environment. The following are inputs used in our vehicle tracking scenario:

- Vehicle type, location, heading, and speed
- Wind speed
- Department of Homeland Security (DHS) advisory level
- Average battery life of sensor modules
- Day of week
- Time of day.

The situation assessment module provides the highest level information about the situation to the human decision maker as well as meta-information about its assessment. Its outputs include the following information:

- Threat Level (low, moderate, or low)
- Evidence in support of the threat level
- Evidence against the threat level
- Confidence in the threat level
- Suggested response(s) to the threat level.

A situation assessment module is performed by a weighted rule and a Bayesian filter.

The weighted rule approach to situation assessment first transforms each input into a category according to Table 7.

The next step is to compute the assessed threat level from a linear combination of all of the input categories, weighted according to their relative importance.

$$\text{Threat} = 5L + 4S + 3H + 2V + W + B + D + T + \text{DW}. \quad (2)$$

Table 7
Weighted rule approach to situation assessment using Threat $= 5L + 4S + 3H + 2V + W + B + D + T + \mathrm{DW}$.

Input	Range	Category	Numeric value
Vehicle location (L)	North road	High	2
	Intersection	Moderate	1
	East/West road	Low	0
Vehicle speed (S)	>40 km/hr	High	2
	20–40 km/hr	Moderate	1
	<20 km/hr	Low	0
Vehicle heading (H)	NW, N, NE	High	2
	W, E	Moderate	1
	SW, S, SE	Low	0
Vehicle type (V)	Tracked	High	2
	Light	Moderate	1
	Anomaly	Low	0
Wind speed (W)	>40 km/hr	Moderate	1
	≤25 km/hr	Low	0
Battery capacity (B)	≤50%	Moderate	1
	>50%	Low	0
DHS advisory level (D)	High, Severe	High	2
	Elevated, Guarded	Moderate	1
	Low	Low	0
Time of day (T)	Off hours	Moderate	1
	Normal hours	Low	0
Day of week (DW)	Weekend, Holiday, Special day	Moderate	1
	Normal weekday	Low	0

The Threat Index is then converted to a threat category, to be presented to the decision maker.

$$\begin{array}{ll} \text{Threat Index} > 22: & \text{High Threat} \\ 11 \leq \text{Threat Index} \geq 22: & \text{Moderate Threat} \\ \text{Threat Index} < 11: & \text{Low Threat.} \end{array}$$

The second approach to situation assessment designed for use in the force protection scenario involved a Bayesian filter.

We generate reasonable estimates of the conditional probabilities, given expectations about the environment and interactions. For example, quantities such as the probability that a vehicle is a threat, given that it is moving at a certain speed is set a priori. Our method runs the above calculation for each of the four "state" calculations and then selects the maximum of that set. Further research may upgrade the efficacy of the a priori estimates while the system runs online. The weighted rule formula used in the previous section can be used to establish initial conditional probabilities for the Bayesian Filter.

3.2.4. Graphical user interface module

The GUI designed to provide decision support for a force protection decision maker includes three screens – the Track Detail screen, the Log screen, and the Map screen.

The Track Detail screen (Fig. 4) consists of four general sections of information. The upper-most left section provides basic track parameters that are largely generated by the fusion module. Beneath this section (titled, "Basis for Assessment") are fields of information that convey how the assessments were derived. Further, there are details that show how the assessment may be invalid ("Against Evidence"). Such an approach offers some transparency that facilitates objective situation assessments. The lowest section that spans the width of the screen is a list of all tracks, including friendly forces that have most recently arrived in the track cue and are available to be specified in greater detail in the screen sections above. The section on the right is generally the course of action information. A list of possible operational activities by the threat is listed alongside how defense forces should respond. A basis for the corresponding defense operations is provided that conveys practical capabilities in the current context and possible constraints.

The Log screen (Fig. 5) affords the decision maker temporal information. A critical element in situation awareness is time-oriented information (Endsley, 1995). Pace of events and time available to decide and act facilitate situation awareness and more effective decision making. The vertical bar in Fig. 5 indicates the current time (i.e., "now"). The numbers across the top are time increments and move right to left in the application. The boxes represent events or tracks and are organized vertically with respect to priority, so the green box at the top is the most important event involving a track that may be approaching the gate. The green coloring corresponds to a low level threat. The box turns yellow for a moderate threat and red for a high threat. The threat assessment is driven by the same data used to drive threat assessments in the track detail screen. If the track moves towards the gate, the box moves right to left, and the estimated time to reach the gate decreases. In this case, the decision maker knows that there are at least 28.2 min until the track reaches the gate. If the vehicle were to pass the gate, then the box would have passed the vertical bar, and the box would indicate how long ago the vehicle passed the gate. The log screen shows general event data, such as computer network activity/announcements, and events that are significant for perimeter security, such as sunset times and high winds that may affect sensor reliability.

4. Future work

Arguably the most immediate area of future work is in establishing principles and practices for employing the three learning modes. There are different ways of combining three modes of machine learning and many options for how and when to employ each mode. The current research offers a preliminary perspective on leveraging each learning mode for highest system performance. It stands to reason that a CARTMAP network can be tailored for each information fusion mode (vehicle type, speed, heading, and location). The vigilance parameter may be different for each mode. The vigilance may also require adjustment based on the type and ordering of the learning modes.

The core of our machine learning approach is an ART neural network. Other algorithms and architectures should be explored with the same goal in mind, that of integrating multiple learning modes. Reinforcement learning is a general area of research worth pursuing in the area of situation awareness where there is often not a clear win or lose outcome from which to measure success. There are also many ways of performing reinforcement learning, some closer to supervised learning, with stronger hints, and others that

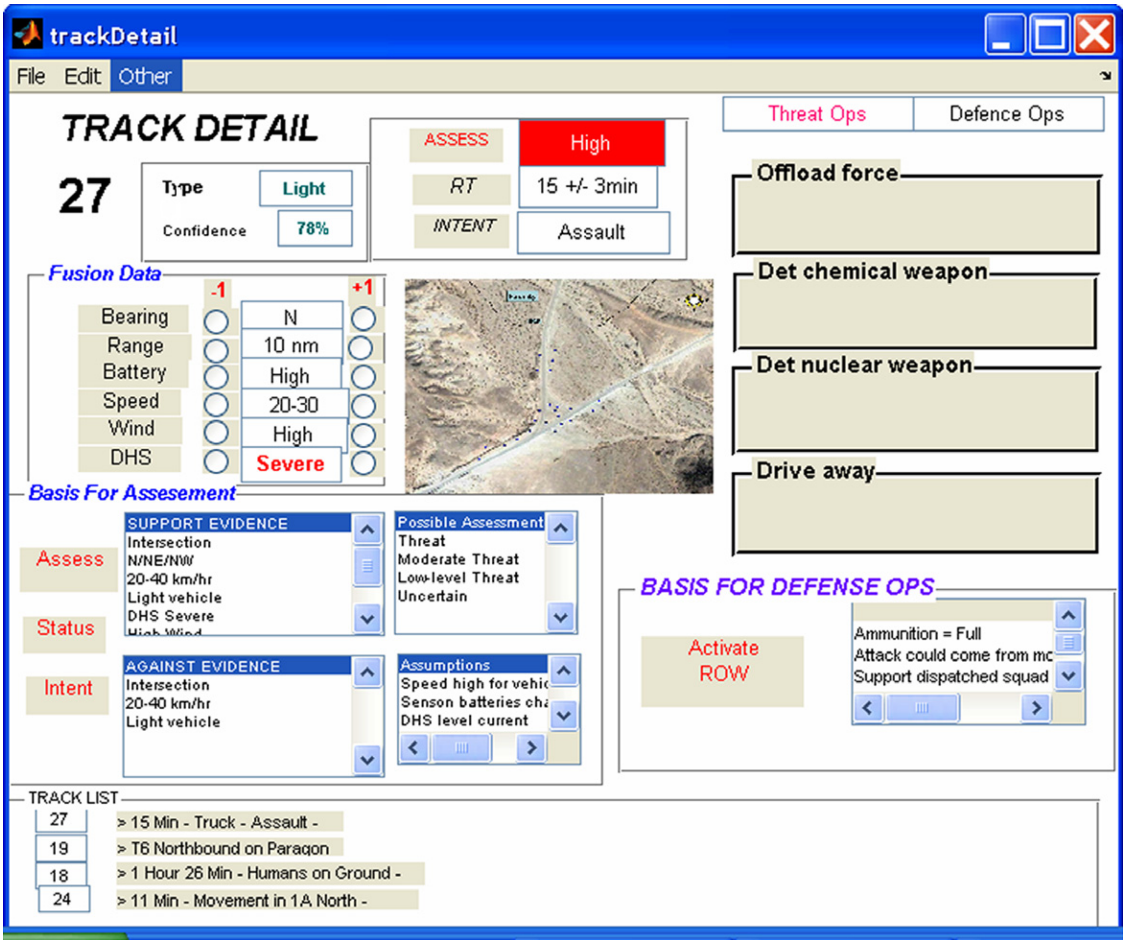

Fig. 4. Track detail interface presenting to a user the threat level, response suggestions, vehicle and environment data, and evidence for and against the system's outputs.

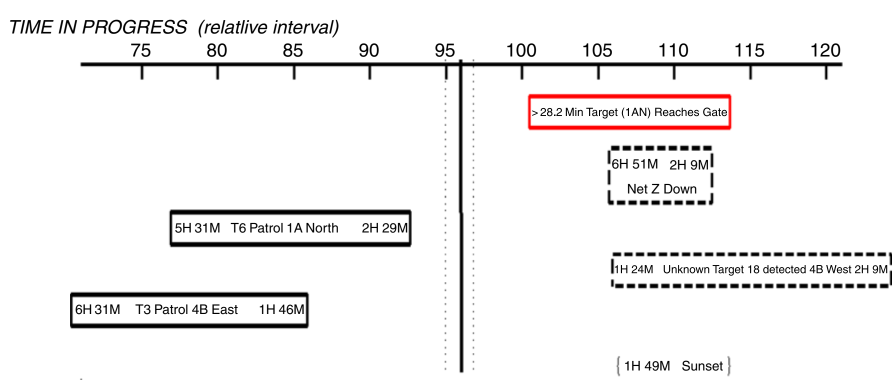

Fig. 5. Decision support log screen showing relative timing of critical events.

provide rare, but consistent hints about the system's performance. How many iterations to use in reinforcement learning on this problem is a legitimate research question, as is how best to acquire feedback from human decision makers or the overall force protection system, either directly or indirectly.

Another avenue of future machine learning research is to explore the use of ensembles or bagging for supervised learning (Dietterich, 2000). The use of ensembles employs multiple "experts" that train the same network using a different sampling with replacement from the original supervised training set. The combination of the experts' solutions results in higher performance than the use of a single network trained on the original dataset.

5. Conclusion

The coordination of the three major machine learning approaches in a single architecture, using ARTMAP at its core, is an innovation that should prove valuable in addressing real-world problems. Many domains offer a limited amount of information with ground-truth that can be used with supervised learning algorithms. More available is data with hints from the environment that can be used with reinforcement learning. Almost always, data is available without labels that can be used with unsupervised learning. Allowing these three modes of learning to be used in the same framework is an important contribution. Interesting advantages emerge when these three approaches leverage one another. For example, reinforcement learning can utilize supervised learning when enough information about class labels is available from the environment. Unsupervised learning can take advantage of stored reinforcement learning information to go beyond mere clustering. There is potential for interplay between the learning modes that does not exist with a single mode.

Acknowledgments

The authors would like to thank the contributions to this work by Pengchu Zhang and Gregory Conrad.

References

Alpaydin, E. (2004). *Introduction to machine learning*. Cambridge, MA: MIT Press, pp. 1–3.

Amis, G., & Carpenter, G. (2007). Default ARTMAP 2. In *Proceedings of the international joint conference on neural networks*.

Brannon, N., Conrad, G., Draelos, T., Seiffert, J., & Wunsch, D. (2006). Information fusion and situation awareness using ARTMAP and partially observable markov decision processes. In *Proceedings of the international joint conference on neural networks*.

Carpenter, G., & Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, *21*(3), 77–87.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J., & Rosen, D. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*(5), 698–713.

Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, *40*(2), 139–157.

Duarte, M., & Hu, Y. (2004). Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, *64*(7), 826–838.

Endsley, M. (1995). Toward a theory of situation awareness. *Human Factors*, *37*(1), 32–64.

Hall, D., & Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, *85*(1), 6–23.

Kokar, M., Tomasik, T., & Weyman, J. (2004). Formalizing classes of information fusion systems. *Information Fusion*, *5*(3), 189–202.

Morrison, J., Kelly, R., Moore, R., & Hutchins, S. (1997). Tactical decision making under stress (TADMUS) decision support system. In *Proceedings IRIS national symposium on sensor and data fusion*.

Paul, J. (2001). Smart sensor web web-based exploitation of sensor fusion for visualization of the tactical battlefield. *IEEE AESS Systems Magazine*,.

Perlovsky, L. (2009). Language and cognition. *Neural Networks*, *22*(3), 247–257.

Puterman, M. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York: John Wiley & Sons.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Werbos, P. (2009). Intelligence in the brain: A theory of how it works and how to build it. *Neural Networks*, *22*(3), 200–212.

Werbos, P. (1990). Consistency of HDP applied to a simple reinforcement learning problem. *Neural Networks*, *3*(2), 179–189.