Imitating Radar Operator Decisions for Maritime Surveillance Missions Using Bayesian Networks

Angus Brown
University of Glasgow
Glasgow, United Kingdom
a.brown.6@research.gla.ac.uk

David Anderson
University of Glasgow
Glasgow, United Kingdom
dave.anderson@glasgow.ac.uk

***Abstract*—This paper presents the use of Bayesian networks for learning decisions made by a human radar operator carrying out a maritime surveillance mission. By imitating the operator's decisions, a significant increase in the autonomy of a radar surveillance system can be achieved as well as potentially streamlining the qualification process. For maritime scenarios, current literature has only focused on using a Bayesian network (BN) for identification and assessment, and often assumes inputs from a generic surveillance sensor. Furthermore, in both the maritime surveillance and radar operations domain, there has been no investigation into using the operator's data in order to learn the decisions made throughout the mission. This paper uses a real-time radar simulation in order to obtain the scenario and radar information that would be observed by a human operator. In conjunction with a user interface, the simulation is further used to obtain operator decisions for a given mission with the maximum likelihood approach used to obtain the BN probabilities. The BN is then used in place of the operator for interfacing with the simulation in order to test the suitability of this method. Several typical scenarios are used to demonstrate the BN's operational ability relative to that of the operator. Additionally, the required data size for sufficient performance is investigated.**

***Index Terms*—Airborne radar, surveillance, Bayesian networks, imitation learning, machine learning**

I. Introduction

Current maritime radar surveillance missions are typically carried out using an airborne platform with one or more operators on board. With progress being made towards the use of remotely operated UAVs for radar surveillance missions, imitating the decisions made by a radar operator with the use of artificial intelligence (AI) would provide a significant step forward in autonomy. By reducing the required human input, and by virtue of the AI being able to perform at all times, the surveillance can be carried out for longer. Additionally, the AI can be applied to multiple platforms simultaneously allowing for greater search area coverage. If it can be shown that the AI can sufficiently imitate the decisions made by a human operator, the qualification of the AI for practical operation could be more streamlined relative to other forms of AI that either learn unsupervised or are manually crafted.

Bayesian networks, a form of AI, have been used for a wide variety of tasks in both radar operations and in maritime surveillance. For example, investigation has been done on the use of a Bayesian network (BN) for autonomous decision making of radar control for polar ice sheet measurements [1]. In terms of maritime surveillance scenarios, a BN has been used for the identification and assessment of maritime objects [2], [3]. Furthermore, a maritime simulation of vessels was used to generate scenarios in which a BN was applied to identify suspicious ships [4].

A BN has also been used to assist a surveillance system operator for maritime situational assessment [5], [6]. Expert knowledge has been used to inform the contextual and behavioral information within a scenario in which a dynamic Bayesian network was used for behavioral based classification of maritime vessels [7]. Additionally, a BN was used for the detection of unusual maritime activity where experts were used to set the network's conditional probability tables [8].

However, while some of the above works leverage expert knowledge, they don't learn from operator data on the scenario nor do they attempt to imitate the operator's decisions. Furthermore, they don't deal with real-time scenarios in which the operator has a sequence of decisions to make where each decision results in more situational information. The use of a BN for this type of imitation task has seen little use [9].

This paper investigates the application of a BN in order to imitate radar operator decisions for maritime surveillance missions. Both the missions and the radar were incorporated within a multi-entity real-time simulation. This simulation included the dynamics of both the airborne platform and the maritime vessels, the signal-to-noise (SNR) obtained from both the environment and the vessels, the tracking of detected objects, and each vessel's automatic identification system (AIS). A graphical user interface (GUI) was also implemented in order to display the radar information as would be observed by the human operator in a realistic scenario. The simulation in conjunction with the GUI was then used to run typical maritime missions, obtain the information observed by the operator (e.g. SNR, track data, or AIS data), and obtain the decisions made by the operator (e.g. move to track, or communicate with the vessel). The operator's data is then used to learn the BN's conditional probability tables using the maximum likelihood approach to expectation maximization.

Several typical tasks of varying complexity are outlined within this paper with each mission involving a sequence of decisions to be made based on the observed information. For each task, the BN was used in place of the operator for

This work was supported by Leonardo.

interfacing with the simulation in order to test the accuracy and error margins of the BN decisions relative to that of the operator. Conclusions are drawn from the size of the data set required by the network for sufficient performance.

II. Bayesian Networks

A Bayesian network [10], [11] represents a series of connected nodes in the form of a directed acyclic graph (DAG) with each node representing a variable (which, in this case, are discrete). Additionally, each connection represents a conditional dependency between two nodes. The nodes within the BN are represented by the vector $\boldsymbol{x}$ of length m with each node x_i having parents denoted by vector $\boldsymbol{u}_i$.

The data set obtained from the scenario and the operator's decisions is denoted by the $N \times m$ table $\boldsymbol{D}$, where N is the number of data instances, and a given data instance is denoted by the vector $\boldsymbol{d}_i$ of length m.

The network estimates θ represent the estimated conditional probabilities for each node. Furthermore, the parameter estimate $\theta_{x|\boldsymbol{u}}$ is used to define the probabilities of node x given the values of the node's parents $\boldsymbol{u}$. In the case where the operator's data is complete (i.e. there are no missing values for each vector $\boldsymbol{d}_i$ within the data set $\boldsymbol{D}$), each parameter estimate can be updated as follows:

$$\theta_{x|\boldsymbol{u}} \triangleq P_{\boldsymbol{D}}(x \mid \boldsymbol{u}) = \frac{C(x \mid \boldsymbol{u})}{C(\boldsymbol{u})}\,, \tag{1}$$

where $P_{\boldsymbol{D}}(x \mid \boldsymbol{u})$ is the probability of event $x \mid \boldsymbol{u}$ from data set $\boldsymbol{D}$. The term $C(\boldsymbol{u})$, for example, is the number of data instances $\boldsymbol{d}_i$ that satisfy event $\boldsymbol{u}$.

The likelihood of the network estimates θ is defined as the probability of observing the data set $\boldsymbol{D}$ under the estimates. However, it is often more convenient to use the log-likelihood which is defined in (2). By definition, (1) obtains the values of θ that maximizes the likelihood and thus log-likelihood.

$$LL(\theta|\boldsymbol{D}) \triangleq \log L(\theta|\boldsymbol{D}) = \sum_{i=1}^{N} \log P_\theta(\boldsymbol{d}_i)\,, \tag{2}$$

where $P_\theta(.)$ is the probability distribution induced by the network structure and the network estimates θ.

It is often the case that the data set is not complete. In other words, at least one of the data instances $\boldsymbol{d}_i$ has a missing value. For example, if the radar operator does communicate with the vessel, then vessel reply is not known for that track. For the scenarios considered in this paper, the data is missing at random (MAR). That is, the fact that the data is missing is unrelated to the missing data itself, but it is related to the observed data of other variables. Under this scenario, the maximum likelihood approach can be directly expanded to account for the missing data by using a local search method. In this case, the expectation maximization (EM) algorithm was used [11]. This method starts with an initial set of values for the estimates (denoted by θ^0). Using these initial estimates, the data set is completed by obtaining the probabilities of each possible completion for instances with missing data. Mainly, θ^k is used to complete the data set with the resulting completed data set used to obtain the new set of parameters θ^{k+1}. This algorithm guarantees that the new set of parameters has a log-likelihood no less than that of the previous set of parameters. As such, this process can be iterated until some convergence criteria is met.

For brevity, the derivation of the algorithm is not mentioned here. However, it should be noted that for the case of incomplete data, the estimates for the next iteration can be obtained as follows:

$$\theta^{k+1}_{x|\boldsymbol{u}} = \frac{\sum_{i=1}^{N} P_{\theta^k}(x\boldsymbol{u} \mid \boldsymbol{d}_i)}{\sum_{i=1}^{N} P_{\theta^k}(\boldsymbol{u} \mid \boldsymbol{d}_i)}\,. \tag{3}$$

In practice, the data is looped over with a 'for' loop, and within, a counter $c_{x\boldsymbol{u}}$ at each step is incremented by the probability $P_{\theta^k}(x\boldsymbol{u} \mid \boldsymbol{d}_i)$ at step i of the loop. It should also be noted that another counter $c_{\boldsymbol{u}}$ can be obtained by summing $c_{x\boldsymbol{u}}$ for all values x. This maximum likelihood expectation maximization algorithm in outlined below.

Algorithm 1: ML Expectation Maximization

Input:
G: Bayesian network structure (X and $\boldsymbol{U}$)
θ^0: inital parameters
$\boldsymbol{D}$: Data set of size N
η: Tolerance for change in log-likelihood value
Output:
θ_{mll}:
begin
  $k \leftarrow 0$
  $\lambda \leftarrow \infty$
  while $\lambda < \eta$ **do**
    $c_{x\boldsymbol{u}} \leftarrow 0$
    for $i = 1;\ i < N;\ i \leftarrow i + 1$ **do**
      foreach $x\boldsymbol{u}$ **do**
        $c_{x\boldsymbol{u}} \leftarrow c_{x\boldsymbol{u}} + P_{\theta^k}(x\boldsymbol{u} \mid \boldsymbol{d}_i)$
      end
    end
    $c_{\boldsymbol{u}} = \sum_{i=1}^{n_x} c_{x_i\boldsymbol{u}}$
    $\theta^{k+1}_{x|\boldsymbol{u}} = c_{x\boldsymbol{u}}/c_{\boldsymbol{u}}$
    $\lambda = LL(\theta_{k+1} \mid \boldsymbol{D}) - LL(\theta_k \mid \boldsymbol{D})$
    $k \leftarrow k + 1$
  end
  $\theta^{\text{mll}} = \theta^k$
end

Since the algorithm is a local search, it may converge on different parameter estimates based on the initial parameter estimates θ_0. As such, the algorithm was ran 5 times with the values for θ_0 normalized on the first run and randomized on subsequent runs. The run which provided the highest log-likelihood was then chosen as the final network. Additionally, each run was terminated when the log-likelihood for the network was less than a set tolerance which in this case was heuristically set to 1×10^{-6}.

III. Radar Simulation and Operator Interface

In order to test the algorithm for a variety of maritime radar surveillance missions, a radar simulation was developed. This simulation was developed and integrated as part of the MAVERIC simulation engine [12] which allows for the simulation of multiple entities (e.g. UAVs, boats) within a set 3D operational environment. MAVERIC, and consequently this simulation, was developed in C++ with the use of Qt for graphical components.

The real-time radar simulation includes power returns from targets with the use of the radar range equation, an approximated beam pattern, and the curvature of the earth accounted for. Sea clutter is also modeled by considering the grazing angle to the surface and using a constant gamma model. Targets are then detected using both local area average (LAA) and M out of N detection, after which a Kalman filter is used to track the detected targets. Furthermore, the entities under surveillance (i.e. the vessels) have the ability to contain an AIS. The user can then fuse the radar tracks with the AIS data to obtain further information—or lack of information—on the tracks.

In conjunction to the radar simulation, a graphical user interface (GUI) was simultaneously developed. This allows for an operator to carry out a maritime radar surveillance mission in a similar way to an operator using an actual radar system within a real environment. The GUI contains a plan position indicator (PPI) to display the radar returns after LAA detection has occurred. The current tracks are overlayed on the PPI display (shown in Fig. 2) with the operator able to select a track resulting in its information being displayed. The GUI also contains radar controls (shown in Fig. 1) to allow for the operator move the platform, and set the sector scan. Furthermore, there is a control for the operator to communicate with a tracked vessel and also a control to indicate for the platform to move towards the track. Additionally, any operator designated zones (e.g. fishing zones) are overlayed on the display.

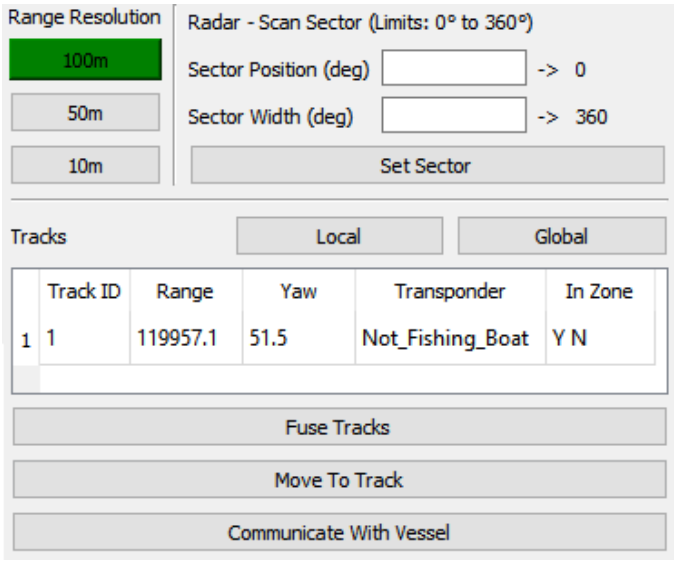

Fig. 1: Section of the radar GUI within MAVERIC.

IV. Preliminary Results

Three tasks representing various stages of a maritime surveillance mission were used to test this method. Each maritime radar surveillance task was simulated and ran a task-specific number times in order to capture both the information observed by the operator and the operator's decisions. The scenario used for all tasks involved two designated fishing zones which are under surveillance by the radar. The operator is monitoring the search area for suspicious activity that may occur in the fishing zones such as illegal fishing operations. Several vessels of random position, speed, heading, RCS, and type are placed within the scene. Note that the first author of this paper acted as the operator by following the operational advice provided by advisors at Leonardo.

The first task involves selecting a track by the process of elimination. Given whether or not there are tracks within any of the zones, within a $10\,\mathrm{km}$ border of the zones, and/or not within $10\,\mathrm{km}$ of the zones, the operator will decide which region to look at first. This decision eliminates all tracks not within the selected region, and as such only the information from the remaining tracks is used for the next decision. The next decision is based on whether or not any of the remaining tracks have AIS information. If there are unknown tracks, the third decision is skipped. If there are no unknown tracks, an additional decision is made on the available tracks based on whether any of the tracks display abnormalities. In this case, the abnormalities are whether the track speed and/or size does not match the information obtained from the AIS. Whether there were known or unknown tracks, a final decision is made based on whether any of the remaining tracks are heading towards the center of the nearest designated zone.

This task is represented by the BN shown in Fig. 3. Note that the white nodes indicate information obtained by the operator while the gray nodes indicate decisions from the operator. The dotted lines indicate that the nodes are not connected within the Bayesian network, but the decision influences the information available to that node. The first decision node TS1 represents the decision to select the tracks within a given region based on the observed information nodes WZ (within zone), W10 (within $10\,\mathrm{km}$), NW (not within $10\,\mathrm{km}$). The next decision node TS2 represents the decision to select tracks based on the observed information nodes K (known) and UK (unknown). The additional decision TS3 represents the decision to further select known tracks based on anomalous information. The observed information nodes for anomalous AIS information ASAS (abnormal size, abnormal speed), ASNS (abnormal size, normal speed), NSAS (normal size, abnormal speed), and NSNS (normal size, normal speed) are determined depending on whether or not the observed size and speed are within the normal operational bounds provided by the AIS.

Using the track selected by the first decision process, the operator will further investigate the track. The operator then

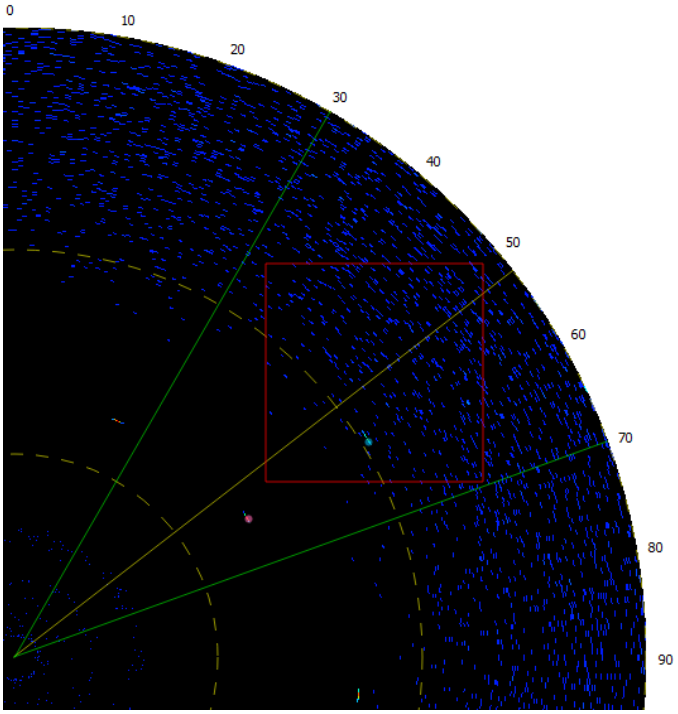

Fig. 2: Radar PPI display within the GUI. The pink dots indicate tracks while the blue dot indicates the currently selected track. The yellow solid line indicates the current azimuth of the radar boresight while the green lines indicate the current sector scan limits. The red box outlines an operator designated zone.

uses the track's AIS data in conjunction with the zone region the track lies within and decides whether or not to apply a sector scan on that particular track. Additionally, the operator also decides whether or not to communicate with the vessel (in the case where the vessel may have their AIS switched off, for example). The vessel's reply to the operator provides further information about the track's AIS data (e.g. the lack of a vessel reply may indicate suspicious activity). Lastly, based on the vessel's reply, the track direction, and track speed, the operator will make a decision about whether or not to move towards the track in order to get a closer look.

This task is represented by the BN shown in Fig. 5. The observed information nodes IZ, AIS, VR, TD, TS represent the track being in one of the fishing zones, the track AIS data, the tracked vessel's reply, the track direction, and the track speed respectively. The state of both AIS and VR indicate whether or not the vessel type is shown as being a fishing boat, not a fishing boat, or that there is no AIS data. The track direction indicates whether or not the track is heading towards the center of the fishing zone. For the BN, the track speed is discretized into three possible states: $0\,\mathrm{m\,s^{-1}}$ to $10\,\mathrm{m\,s^{-1}}$, $10\,\mathrm{m\,s^{-1}}$ to $20\,\mathrm{m\,s^{-1}}$, and above $20\,\mathrm{m\,s^{-1}}$. The operator decision nodes SS, CMV, MTT represent the decision to apply a sector scan to the track, communicate with the track, and move towards the track respectively.

The final decision process occurs when the operator decides to move towards the track in order to get a closer look. The operator has the option of moving towards the track, to the left of the track, or to the right of the track. For the BN, this decision is based on the previous two movement directions as well as whether or not the change in probability was positive, negative, or remained the same. Within a real mission, the change in position influences the ISAR image seen by the operator which determines the decision made regarding whether or not the track is likely to be performing an illegal operation.

Due to the real-time simulation requirement, and for simplicity, an ISAR image generation method in conjunction with an image classifier was avoided. Instead, the ISAR image for maritime surveillance was represented by a probability matrix in terms of sea state, relative azimuth and elevation between the track and the platform, SNR, number of revisits, and vessel type. In essence, this probability could be thought of as the output from an ISAR image classifier. A low probability indicates that the observed track is unlikely to be performing any illegal operations whilst a high probability indicates that the track is likely performing an illegal operation. Furthermore, a probability such as 0.5 indicates uncertainty in whether the track is performing illegal operations or not.

Task 3 is represented by the BN shown in Fig. 7. The first decision node M represents the move made in terms of left, forward, and right. This decision is based on the information nodes PI1, PD1, PI2, and PD2. PD1 represent the move made at the previous step, and PI1 represents the change in probability at the previous step. Similarly, PD2 and PI2 represent the move made and change in probability two steps ago. For the BN, the upper and lower limits for determining if the probability had changed positively or negatively were 0.05 and -0.05 respectively. The final decision node A represents the action performed based on the updated probability I. The action consists of three options: continue moving towards the track, return to the surveillance path and note that the track is not likely to be performing an illegal operation, or declare the track as potentially carrying out an illegal operation. For node I, the probability was discretized into probability bands of 0.05 with a two larger bands of 0.2 in the middle as it was assumed that the operator was likely to continue move towards the track when the uncertainty was high.

When comparing the decisions made by the BN relative to the operator, the recorded operator decision data was split into training data and test data. The training data is the data instances used to train the network, with the size of the training data equal to N. The size of the test data was fixed at 100 data instances for all tasks. The size of training data (i.e. N) varied depending on the tasks as certain tasks are able to learn faster. For each value of N, 20 networks were obtained (with the order of the training data randomized for every network). Each of these networks was tested within the simulation against the operator's decisions using the test data. In other words, given the same observed information, the decisions of the BN were compared with the decisions previously made by the operator.

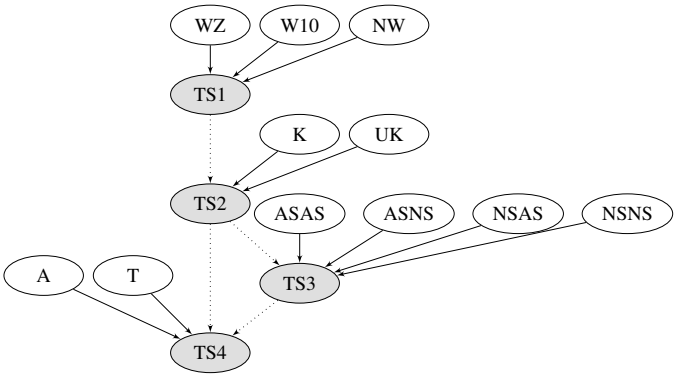

Fig. 3: Task 1 represented as a Bayesian network.

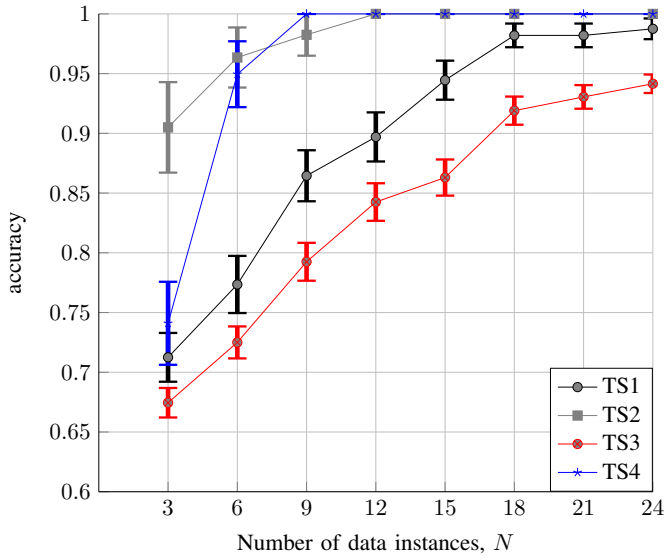

Fig. 4: Accuracy and error of each decision node for task 1.

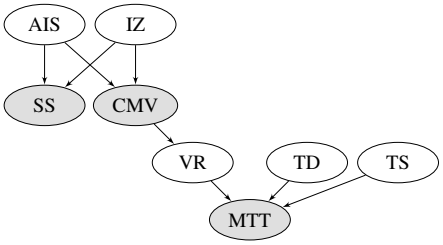

Fig. 5: Task 2 represented as a Bayesian network.

The accuracy for each decision was simply the number of correct decisions relative to the operator divided by N. The results of task 1, task 2, and task 3 in terms of accuracy and error margins are shown in Fig. 4, 6, 8 respectively. These graphs shown the mean accuracy of each decision node for a given number of data instances with error bars shown representing 1 standard error. Note that the accuracy axis starts at 0.6.

For task 1, the decision node with the lowest accuracy is the selection of tracks based on anomalous AIS information. This

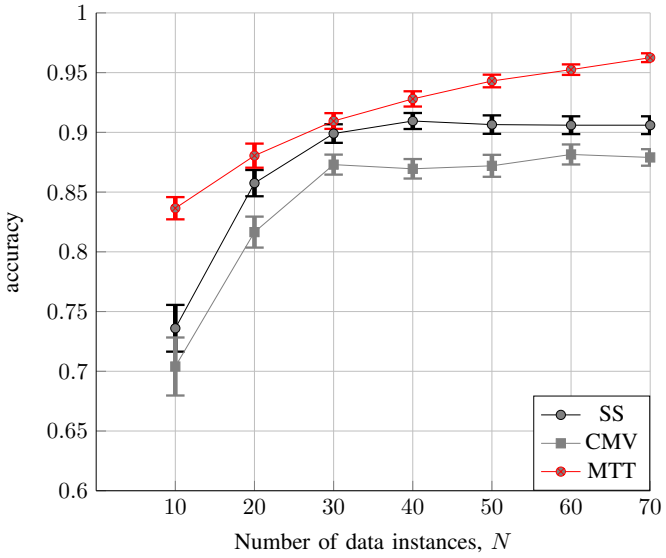

Fig. 6: Accuracy and error of each decision node for task 2.

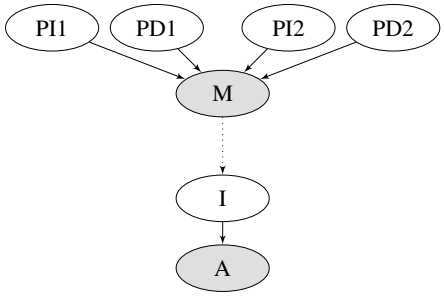

Fig. 7: Task 3 represented as a Bayesian network.

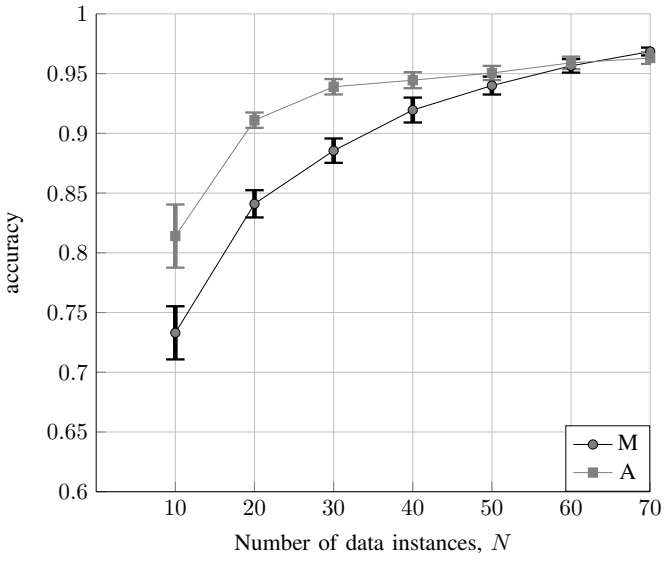

Fig. 8: Accuracy and error of each decision node for task 3.

is due to the fact that this node is only used when the AIS data is known. Additionally, if the scene has few tracks available,

the track is occasionally selected before the fourth decision is required. As such, with few data instances, the fourth AI decision performs poorly, but with high data instances is able to perform accurately. It is also unsurprising that decisions further down the network required more data to achieve the same performance. Data would only be seen for nodes further down the network if previous decisions were made that led to those nodes. As such, any missions with a long sequence of decisions will require significantly more operator data than a mission with a comparable number of decisions but performed in parallel. However, the BN still provided high levels of accuracy for this mission.

For task 2, it is surprising that the node furthest down the network performs the best. However, this is due to the fact that this decision is more deterministic than the other two decisions in the network. In other words, the operator will likely only move towards the track when there in specific and defined situations. The stochastic nature of the SS and CMV decisions results in the accuracy plateauing at around 0.9. Whilst this may seem to offer lower performance than the previous task, the BN is capable of learning the probabilities for these decisions, but it is desirable for any form of AI to be deterministic rather than stochastic (i.e. the AI will perform the same action given the same inputs). As such, the accuracy is reduced in by operating in a deterministic manner.

For task 3, initially the action node has a higher accuracy, but at a greater number of data instances the move node is higher. This is due to the more uncertain nature of the move decision which, as mentioned previously, results in a reduced accuracy due to the deterministic manner of the AI.

V. Conclusions

In this paper, the use of a Bayesian network for imitating a radar operator carrying out a maritime surveillance mission has been outlined. The network's probabilities were learned from operator decision data with the used of a real-time radar simulation for three common surveillance tasks. The network was then tested within the simulation against the operator's decisions given the same observational information. The results show the high accuracy and low error margin of the BN relative to the operator indicating its usefulness in imitating a human operator and also its usefulness in being able to carry out the missions autonomously.

Acknowledgment

The authors would like to thank Leonardo for their assistance in this work.

References

[1] S. Sivashanmugam and C. Tsatsoulis, "A Bayesian Network for Autonomous Sensor Control during Polar Ice Sheet Measurements," in *IEEE International Geoscience and Remote Sensing Symposium*, 2004, pp. 101–104.

[2] M. Krüger, J. Ziegler, and K. Heller, "A Generic Bayesian Network for Identification and Assessment of Objects in Maritime Surveillance," in *15th International Conference on Information Fusion*, 2012, ISBN: 978-0-9824438-5-9.

[3] R. N. Carvalho, R. Haberlin, P. C. G. Costa, *et al.*, "Modeling a Probabilistic Ontology for Maritime Domain Awareness," in *14th International Conference on Information Fusion*, 2011, ISBN: 978-0-9824438-2-8.

[4] P. C. Costa, K. Laskey, K. C. Chang, *et al.*, "High-level information Fusion with Bayesian Semantics," in *CEUR Workshop Proceedings*, 2012.

[5] Y. Fischer and J. Beyerer, "Modeling of Expert Knowledge for Maritime Situation Assessment," *International Journal on Advances in Systems and Measurements*, vol. 6, no. 3&4, pp. 245–259, 2013.

[6] Y. Fischer, A. Reiswich, and J. Beyerer, "Modeling and Recognizing Situations of Interest in Surveillance Applications," in *IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2014, pp. 209–215.

[7] J. J. Dabrowski, J. P. de Villiers, and C. Beyers, "Context-based behaviour modelling and classification of marine vessels in an abalone poaching situation," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 95–111, 2017. DOI: 10.1016/j.engappai.2017.06.005.

[8] F. Fooladvandi, C. Brax, P. Gustavsson, and M. Fredin, "Signature-based activity detection based on Bayesian networks acquired from expert knowledge," *12th International Conference on Information Fusion*, 2009.

[9] R. Parra and L. Garrido, "Bayesian Networks for Micromanagement Decision Imitation in the RTS Game Starcraft," in *Advances in Computational Intelligence*, 2012, pp. 433–443, ISBN: 978-3-642-37797-6. DOI: 10.1007/978-3-642-37798-3_38.

[10] R. E. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, 2003, ISBN: 978-0130125347.

[11] A. Darwiche, *Modeling and reasoning with Bayesian networks*. Cambridge University Press, 2009, ISBN: 978-0-521-88438-9.

[12] D. Anderson, *MAVERIC: Modelling of Autonomous Vehicles using Robust, Intelligent Computing*. [Online]. Available: https://www.gla.ac.uk/media/media%7B%5C_%7D480053%7B%5C_%7Den.pdf (visited on 06/21/2018).