Bayesian Networks for Object and Intent Classification in Ground-Based Air Defense Systems

Leete Skinner

Johns Hopkins University

EN.605.745 – Reasoning Under Uncertainty – Fall 2022 – Final Project

Abstract

As a new generation of threats take to the skies, the task of effective ground-based air defense has only become more challenging. Highlighted by sophisticated unmanned platforms and the advent of hypersonic missiles, these two new threat classes combined with legacy platforms make for a complex and dynamic battlespace. In this arena, information dominance and situational awareness will be the key defining system features governing success. This paper reviews how Bayesian Networks have been used in the past to address individual elements of the overall air defense problem frame, and proposes a Bayesian Network capable of performing both object and intent classification from a variety of available sensor units.

1. Introduction

As the modern battlespace continues evolving to be ever more reliant on algorithmically produced decisions and determinations, shortcomings of current machine learning techniques like lack of explainability and inability to generalize have lead to widespread concerns over use of AI on the battlefield. Coined "Second Wave AI" by DARPA, current generation techniques are known as Narrow AI and characterized by their ability to meet or exceed human level performance for specific tasks, but are unable to generalize to unseen domains or problem frames [1]. The solution to these issues is known as "Third Wave AI", which promises the fusion of human level performant algorithms with the ability to reason over data and application domains to produce trustworthy and super-human results.

These Third Wave AI systems have the potential to revolutionize warfare and provide critical enhancements to already deployed systems, and systems of the future [2]. One critical application area where AI is poised to have significant impact is in the air defense regime. With new threats such as hypersonic missiles, advanced AI-piloted craft, and improvements in technology meant to evade or confuse standard air defense systems (like stealth technology), the ability to properly identify, track, and classify intent is becoming more and more challenging [3-5]. Narrow AI systems with limited or non-existent means of reasoning or generalization have the potential to be fooled into delivering incorrect results [6], leading to catastrophic overall failures in defense systems.

This paper will review some of the available reasoning techniques, focusing specifically on Bayesian Networks and Dynamic Bayesian Networks, and their application to tracking, identification, and intent classification systems as a subset of the multi-target tracking umbrella. Additionally, a prototype Bayesian Network-driven target identification and intent classification application for air defense will be presented alongside several use cases showcasing its benefits over systems lacking the ability to reason.

2. Background and Related Work

Foundational to any modern and advanced air defense system is the data collected by sensor systems operating in the area. Data streams from multiple sensors are typically fused together at different levels to perform three core tasks: detection, tracking, and classification, which are then used to perform higher level tasks such as intent classification, situational awareness, and impact assessment [5]. In the past, these tasks were performed in the paradigm of X-then-Y (e.g., "track-then-classify") architecture, which was effective, but does a poor job leveraging all available information in the system. Specifically, there exist couplings between the capabilities and intents of different targets and the known kinematic and dynamics of targets, that if backpropagated to the tracking layer, would provide valuable information to improve the tracker.

Recently, the Joint Target Tracking, Classification, and Intent (JTCI) framework was proposed, in which the tasks of tracking, classification, and intent (inference) are performed simultaneously [5]. The JTCI relies on two sources of sensor information, a radar and electronic support measure (ESM), which are then processed within a Bayesian Network and leverages several state-transition models to fuse information from task specific models. While the JTCI is one of the first frameworks to perform multi-task inference simultaneously, it builds on a legacy of using Bayesian Networks to support air defense missions. Several Bayesian approaches have been proposed to perform threat evaluation of aerial targets, which typically focuses on the problem frames of identification and classification of targets with respect to defended assets in the area [7,8]. Many of these approaches have been augmented by Dynamic Bayesian Networks (DBNs) as well [9], in which the temporally accommodating aspects of DBNs are better suited to the continuous time nature of air defense. Common across these approaches is using Bayesian Networks to process and fuse extracted stateful and discretized information about targets, such as distance, altitude, speed, and effective threat range [7-10].

However, this is not the only level of abstraction that Bayesian techniques have been used in air defense. At the lower sensor processing level, Bayesian filters have also been used to perform single and multi-target detection and tracking. Here, Bayesian filters are capable of leveraging information extracted from radar, sonar, and optical sensors and are capable of modeling both linear and non-linear relationships [11]. The ability to handle non-linear relationships makes Bayesian approaches much more appealing than traditional Kalman filtering for cases where a Gaussian distribution of the system cannot be assumed [11,12]. Because of this flexibility, Bayesian techniques have also been applied to numerous other domains requiring detection and tracking functionality, such as air traffic control and runway operations [13,14], ground vehicle tracking [15,16], IoT device tracking [17], and personnel/figure tracking [18], among others.

3. Proposed Bayesian Network

Taking inspiration from and building on the above techniques and approaches, we will now introduce and explain the prototype Bayesian Network-driven air defense object and intent classification system. The system is comprised of three main constructs: the available features (observable and derived), the Bayesian Network structure itself, and the determination of conditional probabilities for each node in the network. Each of these components will be discussed in depth below.

3.1 Features and Network Design

Beginning with the overall problem frame, and final output of the system, is the intent classification of a detected object in the hypothetical air defense area of operation. For our purposes, three possible intent modes were identified: a travel/cruise mode, an attack mode, and an evasive/neutralized mode. The

travel/cruise mode is characterized by objects that do not presently pose a threat and are moving into position execute their mission. Next, the attack mode indicates the object is about to cause or presently causing harm to the defended area. Finally, the evasive/neutralized mode means the object is either currently under engagement or has been successfully engaged and is not likely to become a threat soon.

To make this intent classification, there are several useful features we can define to condition the likelihood of each class on. First and foremost is the object/platform type, which is the secondary problem frame addressed here (and its determination will be discussed in the next section). In the hypothetical scenario this system addresses, it is expected that four platform types will be operating in the area: missiles, fighter jets, bombers, and UAVs. Given that each of these platforms poses a different kind of threat, and will position themselves differently for their preferred attack method, this is vital information for ultimately determining the objects intent.

However, knowing that a fighter or a bomber is in the area is not enough information alone, so three other features are used in conjunction with it. The first two are considered low-fidelity general context readings: the range and altitude of the detected object. Range and altitude are useful features here because further away and higher altitude objects are less threatening, and each of the possible platforms operating in the area will tend to be observed at different altitudes and ranges depending on their intent. Range can be observed as “far”, “medium”, or “near”, and altitude can be observed as “high”, “medium”, or “low”. In addition to these two, the speed of the object can also be determined, but in contrast, this is considered a high-fidelity measurement which is informed by two sensors in the system. The determination of speed will be discussed below, but for the purposes of intent classification each platform will operate at an expected speed depending on its intent mode and mission.

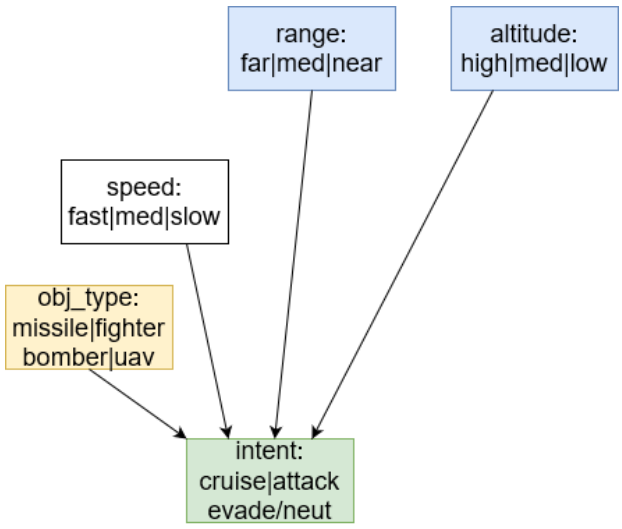

Figure 1 – Intent input nodes and states

Returning to the object/platform type determination, like the intent determination, this is conditioned on multiple input variables. In this case three parameters are used: an altitude observation, a speed measurement, and a size measurement. Like the speed measurement, the size measurement is another high-fidelity measurement also informed by two sensors in the system. Size can be resolved to “large”, “medium”, and “small”, and speed can be resolved to “fast”, “medium”, and “slow”.

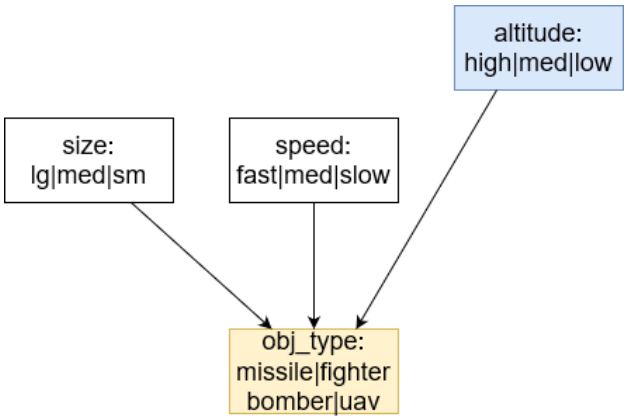

Figure 2 – Object/Platform Type input nodes and states

To properly explain the speed and size measurements, we will now go to the input side of the network and discuss the sensor systems it depends on. For this prototype, hypothetical readings from three different sensor systems will be used: an optical camera, an infrared (IR) sensor, and a radar system. These sensor systems were selected because of the variety of ranges, altitudes, and weather conditions they are most effective (or ineffective) in, and to help compensate for the shortcomings of the other sensors. Here, the camera is capable of observing object size, the IR sensor can observe the object speed, and the radar system can observe both size and speed. However, given the deceptive nature of warfare, we cannot always trust the observations of a sensor system. Taking stealth fighters for instance, the radar profile of an advanced stealth fighter may lead to an incorrect radar observation of a very small or undetectable object, but that same stealth fighter is fully visible to an optical camera system. Because of this, the raw observations from all sensors are combined with the observation from another sensor to make a final measurement. This is done for both size and speed.

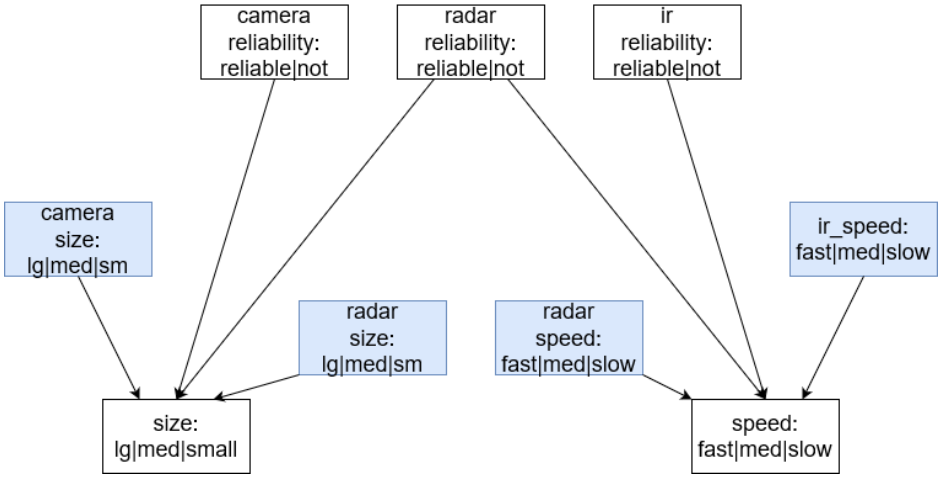

Figure 3 – Size and Speed input nodes and states

Additionally, each of these sensors will perform differently at different ranges and altitudes, and inclement weather in the area will also degrade the performance of each sensor differently. So in addition to combining the observations of each sensor, the reliability or trustworthiness of each sensor is also used to condition the final measurement for size and speed. For this scenario, the camera can be most trusted at near and medium ranges and altitudes. The IR sensor can be most trusted at high altitudes (where there is less atmosphere, so temperature differentials are greater). And the radar system is highly reliable for all

ranges and altitudes, with slight degradation at far range. Under inclement weather conditions however, the camera is severely handicapped to short range and low altitudes only. The IR sensor gains improvement at medium altitudes and the radar system receives a blanket degradation at all ranges and altitudes.

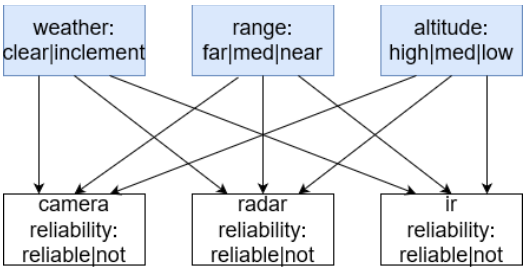

Figure 4 – Sensor Reliability input nodes and states

With each of the functional components of the Bayesian Network defined, as well as the edges connecting each network node/feature, the Figure 5 below shows the full network structure.

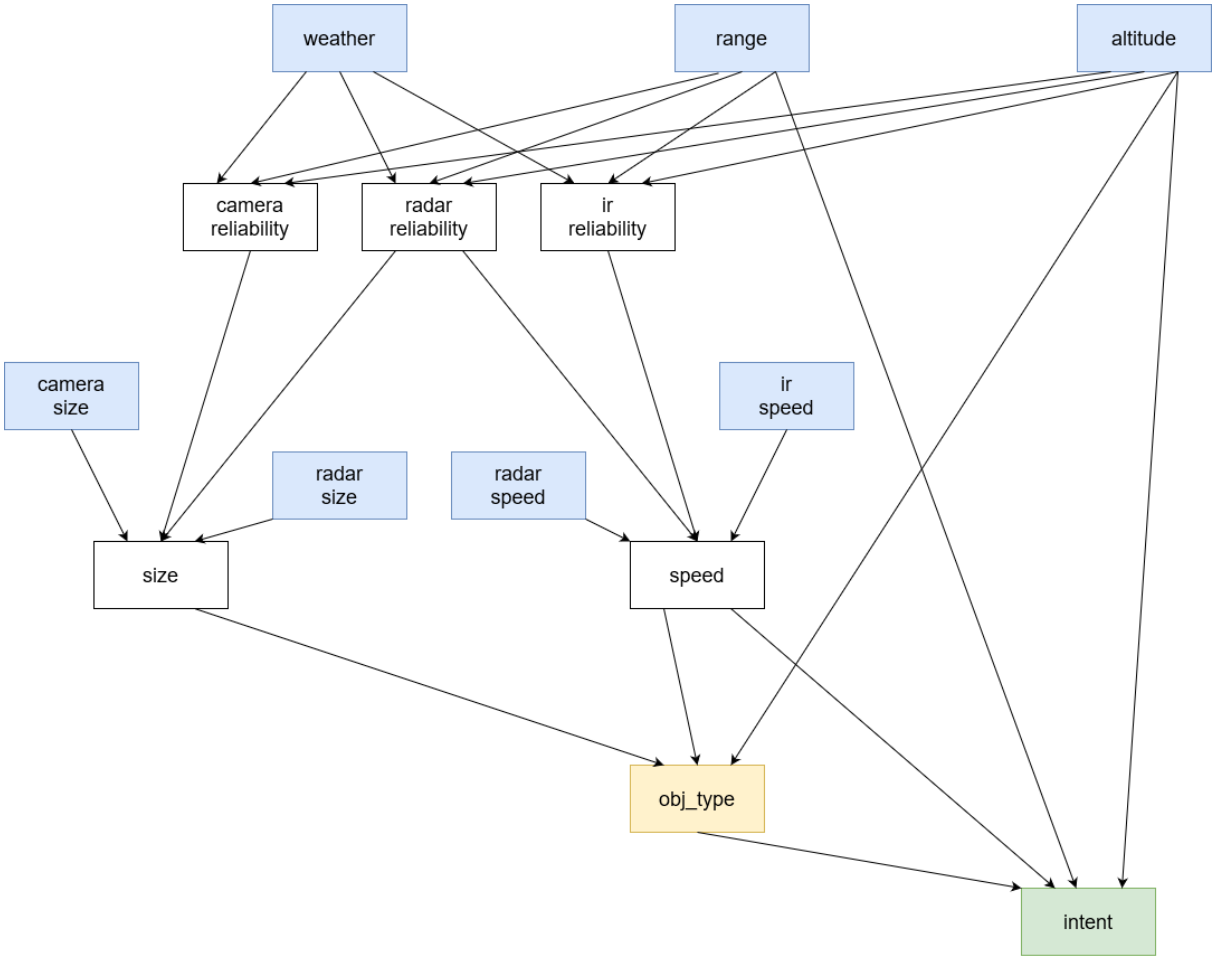

Figure 5 – Full Bayesian Network architecture

3.2 Conditional Probability Distributions

Several different approaches were taken to assign priors or conditional probability values to each node in the network. Beginning first with the observable nodes: weather, range, and altitude, it is expected that these will all be observed and in the test cases to follow. Because of this, their conditional probability distribution (CPD) values are largely non-consequential, however some logic was used to fill them out. The weather node was filled out an environment where 75% of the time the weather is clear, and 25% it is inclement, but this could easily be filled out with heuristic values. For range, it is unlikely for an object to just "appear", so it is reasonable to assign a higher probability to the "far" class as new objects will likely be flying in from outside of range, and "medium" less likely than "far", and "near" less likely than "medium". Altitude is assigned an equal probability to each class.

Next, the raw sensor observation node CPDs were filled out. For size class observations with possible values of "large", "medium", and "small", these were filled out by logic of the opposing force platform distribution. In this case, the majority of the platforms are of size "medium", followed by "small", making "large" vehicles the lowest likelihood reflecting the smallest portion of force. This logic is applied to both the camera observation of size, and the radar observation of size, and these values could easily be updated by heuristic or real world data.

After this, the sensor reliability nodes were filled out in accordance with the optimal sensor operation conditions. To reiterate, the camera is most reliable at near range and low altitude, and under clear conditions. The radar is slightly negatively impacted by range, but performs generally well over all altitudes, and inclement weather reduces reliability by 15% across all ranges and altitudes. The IR sensor is positively impacted by altitude, regardless of range, and inclement weather improves its reliability for objects at medium altitudes.

For the class of size and speed measurements, which are both conditioned by the sensor reliability nodes and the raw observation nodes, the logic for filling these CPDs out is as follows. If the sensor is reliable, we trust its measurement more than a not-reliable measurement. Additionally, if both sensor observations agree, we have the highest possible confidence in the agreed classification. If the sensors disagree, but have adjacent classes (e.g. large and medium, or medium and small) the probability mass is split across the two, favoring the more reliable sensors observation. For disagreements of non-adjacent classes (large and small), one or both sensors is clearly wrong, so the probability mass is spread evenly while slightly favoring the more reliable sensors observation.

For the object/platform type CPD, an algorithmic approach was taken to produce the probabilities. Here, for each possible platform type for each conditioning node/feature was marked as having a possible reading for that feature. Taking the fighter for example, it can be observed at all altitudes, its size will either be small or medium, and its speed will be fast or medium. Knowing these possible operating conditions, a routine was developed to assign probabilistic mass for each conditional scenario.

The same logic was applied to generate the final intent classification node CPD, with an additional feature class control loop for the platform type. It is also important to note here that this method of CPD generation could easily be augmented with a first order rule base. Such a rule base could be created and maintained on a database of past engagements allowing for a regularly updated and re-generated CPDs. Please refer to the appendix for full CPD table values.

4. Example Test Cases

To showcase the utility of this proposed network, a series of test cases were devised, and their conditions and results will be discussed in depth below. Generally speaking, each use case will hold constant all but one or two observable features to highlight how changes in the independent features will result in varying outputs of the system at key decision boundaries between one or more possible output intent. Furthermore, because the complexity of the network, we will focus on specific cases revolving around a hypothetical object operating in the environment, and then show how the network resolves inconsistencies in observations.

4.0 Network Calibration – No Observations

Before diving into specific use cases and scenarios, we will first review outputs of the network when no evidence has been supplied, as shown in Table 0 below. Here, we see that of the four possible objects that can be classified, it is effectively equal probability between each of the options. However, for intent classification, the cruise intent is notably higher than attack and evade/neutralized. Given the lack of observations, this assessment is reasonable as objects operating in the area will spend a disproportionate amount of time moving into or out of position for an attack. This rationale holds consistent for all object and intent combinations, except for missiles which have attack as the highest likelihood. This is also not surprising because while missiles will cruise into position, they will not idlily cruise without an attack being the final objective, unlike the other traditional aircraft options.

Inferred - Object Type	
missile	0.257
fighter	0.257
bomber	0.239
uav	0.246

Inferred - Intent	
cruise	0.439
attack	0.296
evade/neut	0.265

Inferred - Intent for Object Type		
missile	attack	0.111
missile	cruise	0.088
missile	evade	0.058
fighter	cruise	0.112
fighter	attack	0.073
fighter	evade	0.073
bomber	cruise	0.106
bomber	attack	0.066
bomber	evade	0.067
uav	cruise	0.111
uav	attack	0.068
uav	evade	0.067

Table 0 – Results when no Observations present

4.1 Test Case 1 - Missile in Cruise

In the first true use case scenario, we will define a relatively low-threat and low-uncertainty situation. Here, the true values of each observable node are specified in Table 1 below. For now, the weather conditions will stay as clear, and the object was spotted at a far range and high altitude. Important to note as well is that both the camera and radar size observations agree, as well as the radar and IR speed observations.

Observations		Inferred - Object Type		Inferred - Intent for Object Type		
Conditions	clear	**missile**	0.249	**missile**	**cruise**	0.236
Range	far	**fighter**	0.249	**uav**	**cruise**	0.168
Altitude	high	**bomber**	0.219	**fighter**	**cruise**	0.156
		uav	0.283	**bomber**	**cruise**	0.112
Size - Camera	small			**uav**	**evade/neut**	0.105
Size - Radar	small			**bomber**	**attack**	0.062
		Inferred - Intent		**fighter**	**attack**	0.049
Speed - Radar	medium	**cruise**	0.671	**bomber**	**evade/neut**	0.045
Speed - IR	medium	**attack**	0.130	**fighter**	**evade/neut**	0.045
		evade/neut	0.200	**uav**	**attack**	0.010
				missile	**attack**	0.009
				missile	**evade/neut**	0.005

Table 1 – Test Case 1 Observations and Results

Looking first at the object type classifications, we see that the UAV is actually the highest likelihood object, and the bomber is the lowest likelihood object. However, in reviewing the inferred aggregate intent that very clearly the object is in a cruise mode and has a low likelihood of being in attack mode. When inferring intent on a per object type basis, we see that a missile in cruise is by far the highest likelihood scenario which is ~30% stronger indicator than the next highest combination of UAV in cruise. Interestingly, we also see that the two other operating modes of a missile are the lowest likelihood platform-intent combinations.

4.2 Test Case 2 – Bomber in Attack

The next use case showcases the identification of a different object with a different intent. In this case, the observed features are in line with the kinds of behavior we would expect to see a bomber operating with when conducting an attack run. The size and speed raw observations also agree for this case, but we have changed the size observations to be large.

Observations		Inferred - Object Type		Inferred - Intent for Object Type		
Conditions	clear	**missile**	0.203	**bomber**	**attack**	0.201
Range	near	**fighter**	0.203	**uav**	**attack**	0.100
Altitude	medium	**bomber**	0.348	**fighter**	**attack**	0.091
		uav	0.245	**uav**	**evade/neut**	0.084
Size - Camera	large			**bomber**	**evade/neut**	0.079
Size - Radar	large			**missile**	**attack**	0.075
		Inferred - Intent		**missile**	**evade/neut**	0.070
Speed - Radar	medium	**cruise**	0.233	**bomber**	**cruise**	0.068
Speed - IR	medium	**attack**	0.466	**fighter**	**evade/neut**	0.067
		evade/neut	0.301	**uav**	**cruise**	0.061
				missile	**cruise**	0.058
				fighter	**cruise**	0.045

Table 2 – Test Case 2 Observations and Results

Here we see that unlike in Test Case 1, the bomber class stands out as being much more likely than any of the other platform options. Both the missile and fighter remain similar though. However, we do see less confidence in any single class, with the probability of pure intent being more spread out despite still being the highest. When inferring intent on a per-object basis though, we see bomber performing an attack run as decidedly higher than any other object-intent combination, at 20% of the overall probability mass which is 2x as likely as the next highest class.

4.3 Test Case 3 – Bomber with Stealth under Inclement

Building on Test Case 2 from above, we introduce two sources of uncertainty: a disagreement in the size of the object, which can be attributed to an aircraft using stealth technology resulting in a smaller radar profile than what would be optically observed in the same object. Additionally, we introduce inclement weather conditions which alters the reliability of sensor readings at different ranges and altitudes.

Observations	
Conditions	inclement
Range	near
Altitude	medium
Size - Camera	large
Size - Radar	small
Speed - Radar	medium
Speed - IR	medium

Inferred - Object Type	
missile	0.234
fighter	0.234
bomber	0.261
uav	0.271

Inferred - Intent	
cruise	0.235
attack	0.452
evade/neut	0.313

Inferred - Intent for Object Type		
bomber	**attack**	0.149
uav	**attack**	0.111
fighter	**attack**	0.105
uav	**evade/neut**	0.093
missile	**attack**	0.087
missile	**evade/neut**	0.082
fighter	**evade/neut**	0.078
uav	**cruise**	0.067
missile	**cruise**	0.065
bomber	**evade/neut**	0.060
fighter	**cruise**	0.052
bomber	**cruise**	0.051

Table 3 – Test Case 3 Observations and Results

Here we see that even in inclement weather and with disagreeing sensors, that the network is still able to resolve the correct object and intent despite there being more ambiguity in the object type. While not as confident as in Test Case 2, only having ~15% of the probability mass instead of 20%, the network is still able to reason through the sensor obstructions and ambiguity and identify the object and intent properly.

4.4 Test Case 4 – Camera and Radar Offline

In addition to the networks ability to reason through disagreements in sensor observations and compensating for environmental weather conditions, the network is also capable of handling one or more sensor system being taken offline during operation. In this use case, we simulate the camera and radar station being offline an unable to supply measurements. As shown in Table UC4 below, we see that the only observable raw sensor is the IR sensor.

Observations	
Conditions	inclement
Range	near
Altitude	med
Size - Camera	OFFLINE
Size - Radar	OFFLINE
Speed - Radar	OFFLINE
Speed - IR	med

Inferred - Object Type	
missile	0.238
fighter	0.238
bomber	0.259
uav	0.266

Inferred - Intent	
cruise	0.206
attack	0.466
evade/neut	0.328

Inferred - Intent for Object Type		
bomber	**attack**	0.141
uav	**attack**	0.121
missile	**attack**	0.105
fighter	**attack**	0.099
missile	**evade/neut**	0.089
fighter	**evade/neut**	0.088
uav	**evade/neut**	0.081
bomber	**evade/neut**	0.069
uav	**cruise**	0.065
fighter	**cruise**	0.050
bomber	**cruise**	0.049
missile	**cruise**	0.043

Table 4 – Test Case 4 Observations and Results

One of the most interesting results of this trial is that unlike previous trials, the highest likelihood object-intent pair for evade/neutralized is lower than the lowest attack intent, and the highest cruise is below the lowest evade/neut. This is an expression of relying less on the object type to inform the intent and instead using the other available features. It also shows how the reduction of information will not dramatically change classifications despite a reduction in the confidence of said classifications.

4.5 Test Case 5 – Externally Resolved Object

Finally, we consider a scenario where the object type being tracked is observable – whether by a high confidence classification from an external ATR system, or from domain knowledge of that specific object type being the only platform operating in the area. Because the object type is observed, we are now only performing intent classification which relies on the speed sensor readings but not size. Additionally, we specify deliberately vague observations to highlight how the observed object impacts intent classification.

Observations	
Conditions	clear
Range	near
Altitude	med
Speed - Radar	med
Speed - IR	med

Inferred - Intent			
Observed Object	**Cruise**	**Attack**	**Evade/Neut**
missile	0.532	0.443	0.025
fighter	0.515	0.333	0.152
bomber	0.435	0.535	0.030
uav	0.403	0.349	0.248

Table 5 – Test Case 5 Observations and Results

Here we see that despite observing the same non-object parameters, that observing the object has a dramatic impact on the intent of the object. Most interesting is the very low likelihoods of evade/neutralized for both the missile and bomber, and that the cruise probability for missile is effectively the same as the attack likelihood for the bomber.

6. Conclusions

Despite the advances made in other areas of artificial intelligence and machine learning, namely deep neural networks, core probabilistic frameworks such as Bayesian Networks are still strong techniques that should be used to solve the most challenging problems of our time. The ability to encode and reason over human-level domain knowledge, and combining that with other advanced systems will make for more impactful systems than any single approach.

Furthermore, as we have seen by the prototype Bayesian Network for Object and Intent classification, it is possible to fuse together numerous sensor sources and create a robust system with high fault tolerance of any subsystem. Additionally, such a network allows for the integration and interoperability with any number of other advanced decision making models making Bayesian Networks a go to framework to perform last-mile reasoning over any number of non-explainable “black-box” type approaches.

A specific area of future research would be to draw from the field of control theory and enter the headspace of subsystem frequencies, in an algorithmic context. Here, different sensor systems and different decision making algorithms in the system are likely capable of producing measurements or decisions at different computational rates. Knowing this, it could be possible to design a Bayesian Network made up of several “inner” and “outer” control loops where as information becomes available from faster subsystems, these are used to update the priors of outer loops. And, as the higher fidelity outer loops complete their execution, feed these outputs back as priors into the inner loops. Such an architecture would facilitate critical information about the state of the environment being made useful as soon as it comes available. This directly addresses any artificial computational restrictions that effectively relegate a system to becoming a batch processing routine that blocks until all new measurements become available.

References

[1] "DARPA Perspective on AI," Nov 2022. https://www.darpa.mil/about-us/darpa-perspective-on-ai

[2] J. Jones, R. Kress, W. Newmeyer, and A. Rahman. "Leveraging Artificial Intelligence (AI) for Air and Missile Defense (AMD): An Outcome-Oriented Decision Aid," Naval Postgraduate School, 2020. https://apps.dtic.mil/sti/citations/AD1126470

[3] J. Johnson. "The Fast and the Furious: Drone Swarming and Hypersonic Weapons," Artificial Intelligence and the Future of Warfare, 2021. https://doi.org/10.7765/9781526145062.00015

[4] W. He and L. Zhang. "Dim Target Detection in Infrared Image Sequences Using Accumulated Information," Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, 2007. https://doi.org/10.1007/978-1-4020-6266-7_89

[5] W. Zhang, F. Yang and Y. Liang, "A Bayesian Framework for Joint Target Tracking, Classification, and Intent Inference," in IEEE Access, vol. 7, pp. 66148-66156, 2019. https://ieeexplore.ieee.org/document/8717680

[6] S. Qiu, Q. Liu, S. Zhou, and C. Wu. "Review of Artificial Intelligence Adversarial Attack and Defense Technologies," Applied Sciences, 2019. https://doi.org/10.3390/app9050909

[7] J. F. Basso Brancalion and K. H. Kienitz, "Threat Evaluation of Aerial Targets in an Air Defense System Using Bayesian Networks," 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 2017. https://ieeexplore.ieee.org/document/8328495

[8] F. Johansson and G. Falkman, "A Bayesian network approach to threat evaluation with application to an air defense scenario," 2008 11th International Conference on Information Fusion, 2008. https://ieeexplore.ieee.org/abstract/document/4632368

[9] Y. Wang, Y. Sun, J. -Y. Li and S. -T. Xia, "Air defense threat assessment based on dynamic Bayesian network," 2012 International Conference on Systems and Informatics, 2012. https://ieeexplore.ieee.org/document/6223112

[10] X. Wang, J. Zuo, R. Yang, Z. Zhang, L. Yue, and H. Liu, "Target Threat Assessment Based on Dynamic Bayesian Network," Journal of Physics, 2019. https://iopscience.iop.org/article/10.1088/1742-6596/1302/4/042023/pdf

[11] Kristine Bell; Thomas Corwin; Lawrence Stone; Roy Streit, Bayesian Multiple Target Tracking, Second Edition, Artech, 2013. https://ieeexplore.ieee.org/document/9101112

[12] A. Alin, M. V. Butz and J. Fritsch, "Tracking moving vehicles using an advanced grid-based Bayesian filter approach," 2011 IEEE Intelligent Vehicles Symposium (IV), 2011. https://ieeexplore.ieee.org/document/5940471

[13] X. Ye, G. Kamath and L. A. Osadciw, "Using Bayesian inference for sensor management of air traffic control systems," 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM), 2009. https://ieeexplore.ieee.org/document/4938824

[14] E. S. Ayra, D. R. Insua and J. Cano, "Bayesian Network for Managing Runway Overruns in Aviation Safety," Journal of Aerospace Information Systems, 2019. https://arc.aiaa.org/doi/full/10.2514/1.I010726

[15] F. Dellaert and C. Thorpe, “Robust Car Tracking using Kalman filtering and Bayesian templates,” Proceedings of SPIE Intelligent Transportation Systems, 1998. https://www.ri.cmu.edu/publications/robust-car-tracking-using-kalman-filtering-and-bayesian-templates/

[16] J. Elfring, R. Appeldoorn and M. Kwakkernaat, “Multisensor simultaneous vehicle tracking and shape estimation,” 2016 IEEE Intelligent Vehicles Symposium (IV), 2016. https://ieeexplore.ieee.org/document/7535453

[17] W. Zhang, J. Zhang, M. Bao, X. -P. Zhang and X. Li, “Multitarget Tracking Based on Dynamic Bayesian Network With Reparameterized Approximate Variational Inference,” in IEEE Internet of Things Journal, 2022. https://ieeexplore.ieee.org/document/9628185

[18] V. Pavlovic, J. M. Rehg, Tat-Jen Cham and K. P. Murphy, “A dynamic Bayesian network approach to figure tracking using learned dynamic models,” Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999. https://ieeexplore.ieee.org/document/791203

Appendix

A note on the tables: the network was implemented with the python library pgmpy (https://pgmpy.org/) and these tables are structured to meet their API interface

Weather		Range		Altitude	
clear	0.75	far	0.60	high	0.33
inclement	0.25	med	0.30	med	0.34
		near	0.10	low	0.33

Table A.1 – Low-fidelity observable features CPD values

		conditions	clear	clear	clear	clear	clear	clear	clear	clear	clear	incl	incl	incl	incl	incl	incl	incl	incl	incl
		range	far	far	far	med	med	med	near	near	near	far	far	far	med	med	med	near	near	near
	Camera	alt	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low
		reliable	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	0.00	0.00	0.10	0.00	0.10	0.60	0.10	0.50	0.90
		not	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10	1.00	1.00	0.90	1.00	0.90	0.40	0.90	0.50	0.10
		conditions	clear	clear	clear	clear	clear	clear	clear	clear	clear	incl	incl	incl	incl	incl	incl	incl	incl	incl
		range	far	far	far	med	med	med	near	near	near	far	far	far	med	med	med	near	near	near
Reliability	Radar	alt	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low
		reliable	0.75	0.75	0.75	0.85	0.85	0.85	0.90	0.90	0.90	0.60	0.60	0.60	0.70	0.70	0.70	0.75	0.75	0.75
		not	0.25	0.25	0.25	0.15	0.15	0.15	0.10	0.10	0.10	0.40	0.40	0.40	0.30	0.30	0.30	0.25	0.25	0.25
		conditions	clear	clear	clear	clear	clear	clear	clear	clear	clear	incl	incl	incl	incl	incl	incl	incl	incl	incl
		range	far	far	far	med	med	med	near	near	near	far	far	far	med	med	med	near	near	near
	IR	alt	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low	high	med	low
		reliable	0.90	0.50	0.10	0.90	0.50	0.10	0.90	0.50	0.10	0.60	0.50	0.40	0.60	0.50	0.40	0.60	0.50	0.40
		not	0.10	0.50	0.90	0.10	0.50	0.90	0.10	0.50	0.90	0.40	0.50	0.60	0.40	0.50	0.60	0.40	0.50	0.60

Table A.2 – Sensor Reliability CPD Values

Camera Size		Radar Size	
lg	0.15	lg	0.15
med	0.45	med	0.45
sm	0.40	sm	0.40
force distribution		force distribution	

	cam_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel
	rad_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
	cam_size	large	large	large	med	med	med	sm	sm	sm	large	large	large	med	med	med	sm	sm	sm	
	rad_size	large	med	sm	large	med	sm	large	med	sm	large	med	sm	large	med	sm	large	med	sm	
	lg	0.90	0.45	0.33	0.45	0.05	0.10	0.33	0.45	0.05	0.80	0.80	0.80	0.10	0.10	0.10	0.10	0.10	0.10	
	med	0.05	0.45	0.34	0.45	0.90	0.45	0.34	0.45	0.05	0.10	0.10	0.10	0.80	0.80	0.80	0.10	0.10	0.10	
	sm	0.05	0.10	0.33	0.10	0.05	0.45	0.33	0.10	0.90	0.10	0.10	0.10	0.10	0.10	0.10	0.80	0.80	0.80	
Size																				
	cam_rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
	rad_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
	cam_size	large	large	large	med	med	med	sm	sm	sm	large	large	large	med	med	med	sm	sm	sm	
	rad_size	large	med	sm	large	med	sm	large	med	sm	large	med	sm	large	med	sm	large	med	sm	
	lg	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.60	0.40	0.33	0.40	0.20	0.20	0.33	0.20	0.20	
	med	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.20	0.40	0.34	0.40	0.60	0.40	0.34	0.40	0.20	
	sm	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.20	0.20	0.33	0.20	0.20	0.40	0.33	0.40	0.60	

Table A.3 – Size Raw Observation Priors, and resolved Size CPD values

IR Speed		Radar Speed	
lg	0.30	lg	0.30
med	0.40	med	0.40
sm	0.30	sm	0.30
quasi normal dist		quasi normal dist	

Speed

cam_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	rel
rad_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
ir_speed	fast	fast	fast	med	med	med	slow	slow	slow	fast	fast	fast	med	med	med	slow	slow	slow
rad_speed	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow
fast	0.90	0.45	0.33	0.45	0.05	0.10	0.33	0.45	0.05	0.80	0.80	0.80	0.10	0.10	0.10	0.10	0.10	0.10
med	0.05	0.45	0.34	0.45	0.90	0.45	0.34	0.45	0.05	0.10	0.10	0.10	0.80	0.80	0.80	0.10	0.10	0.10
slow	0.05	0.10	0.33	0.10	0.05	0.45	0.33	0.10	0.90	0.10	0.10	0.10	0.10	0.10	0.10	0.80	0.80	0.80

cam_rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
rad_rel	rel	rel	rel	rel	rel	rel	rel	rel	rel	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r	not_r
ir_speed	fast	fast	fast	med	med	med	slow	slow	slow	fast	fast	fast	med	med	med	slow	slow	slow
rad_speed	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow
fast	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.60	0.40	0.33	0.40	0.20	0.20	0.33	0.20	0.20
med	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.20	0.40	0.34	0.40	0.60	0.40	0.34	0.40	0.20
slow	0.10	0.10	0.80	0.10	0.10	0.80	0.10	0.10	0.80	0.20	0.20	0.33	0.20	0.20	0.40	0.33	0.40	0.60

Table A.4 – Speed Raw Observation priors, and resolved Speed CPD values

Object Type

alt	high	high	high	high	high	high	high	high	high	med	med	med	med	med	med
size	lg	lg	lg	med	med	med	sm	sm	sm	lg	lg	lg	med	med	med
speed	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow
missile	0.26	0.20	0.11	0.29	0.23	0.18	0.32	0.26	0.20	0.26	0.20	0.10	0.29	0.23	0.18
fighter	0.26	0.20	0.10	0.29	0.23	0.18	0.32	0.26	0.20	0.26	0.20	0.10	0.29	0.23	0.18
bomber	0.32	0.36	0.47	0.21	0.27	0.32	0.12	0.19	0.24	0.32	0.36	0.48	0.21	0.27	0.32
uav	0.16	0.24	0.32	0.21	0.27	0.32	0.24	0.29	0.36	0.16	0.24	0.32	0.21	0.27	0.32

alt	med	med	med	low	low	low	low	low	low	low	low	low
size	sm	sm	sm	lg	lg	lg	med	med	med	sm	sm	sm
speed	fast	med	slow	fast	med	slow	fast	med	slow	fast	med	slow
missile	0.26	0.20	0.11	0.29	0.23	0.18	0.32	0.26	0.20	0.26	0.20	0.10
fighter	0.26	0.20	0.10	0.29	0.23	0.18	0.32	0.26	0.20	0.26	0.20	0.10
bomber	0.32	0.36	0.47	0.21	0.27	0.32	0.12	0.19	0.24	0.32	0.36	0.48
uav	0.16	0.24	0.32	0.21	0.27	0.32	0.24	0.29	0.36	0.16	0.24	0.32

Table A.5 – Generated Object/Platform Type CPD via rule-based routine

Intent - Object: Fighter

platform	range	alt	speed	cruise	attack	evade/neut
fighter	far	high	fast	0.53	0.00	0.47
fighter	far	high	med	0.64	0.21	0.15
fighter	far	high	slow	0.55	0.27	0.18
fighter	far	med	fast	0.45	0.15	0.40
fighter	far	med	med	0.53	0.35	0.12
fighter	far	med	slow	0.43	0.43	0.14
fighter	far	low	fast	0.35	0.18	0.47
fighter	far	low	med	0.43	0.43	0.14
fighter	far	low	slow	0.27	0.55	0.18
fighter	med	high	fast	0.53	0.00	0.47
fighter	med	high	med	0.65	0.21	0.14
fighter	med	high	slow	0.55	0.27	0.18
fighter	med	med	fast	0.45	0.15	0.40
fighter	med	med	med	0.53	0.35	0.12
fighter	med	med	slow	0.43	0.43	0.14
fighter	med	low	fast	0.35	0.18	0.47
fighter	med	low	med	0.43	0.43	0.14
fighter	med	low	slow	0.27	0.55	0.18
fighter	near	high	fast	0.23	0.23	0.54
fighter	near	high	med	0.26	0.39	0.35
fighter	near	high	slow	0.15	0.45	0.40
fighter	near	med	fast	0.21	0.31	0.48
fighter	near	med	med	0.23	0.46	0.31
fighter	near	med	slow	0.13	0.52	0.35
fighter	near	low	fast	0.12	0.35	0.53
fighter	near	low	med	0.13	0.52	0.35
fighter	near	low	slow	0.00	0.60	0.40

Table A.6 – Snippet from generated Intent Classification CPD via rule-based routine for Platform: Fighter

```
290  def load_fighter_rules():
291      fighter = {
292          'cruise': {
293              'range': [FAR, MED],      # NOT cruise near to target
294              'alt': [HIGH, MED],       # NOT cruise at low alts
295              'speed': [FAST, MED],     # NOT cruise at slow
296          },
297          'attack': {
298              'range': [NEAR],          # Attack only at near
299              'alt': [MED, LOW],        # Worthwhile attacks only at med, low alts
300              'speed': [MED, SLOW],     # Cannot attack while at top speed
301          },
302          'evade/neut': {
303              'range': [NEAR],          # If NEAR and FAST, getting out of dodge
304              'alt': [LOW, MED, HIGH], # can occur at any alt
305              'speed': [FAST],          #
306          }
307      }
308      return fighter
```

Figure A.1 – Example of rule-base for Intents of Platform: Fighter