2023-04

Adversarial AI Testcases for Maritime Autonomous Systems

Tam, K

10.5772/ACRT.15
AI, Computer Science and Robotics Technology
IntechOpen

Adversarial AI Testcases for Maritime Autonomous Systems

Mathew J. Walter*, Aaron Barrett*, David J. Walker*, Kimberly Tam*
*University of Plymouth, UK
{mathew.walter, aaron.barrett, david.walker, kimberly.tam}@plymouth.ac.uk

Abstract

Contemporary maritime operations such as shipping are a vital component constituting global trade and defence. The evolution towards maritime autonomous systems, often providing significant benefits (e.g., cost, physical safety), requires the utilisation of artificial intelligence (AI) to automate the function of a conventional crew. However, unsecured AI systems can be plagued with vulnerabilities naturally inherent within complex AI models. The adversarial AI threat, primarily only evaluated in a laboratory environment, increases the likelihood of strategic adversarial exploitation and attacks on mission-critical AI, including maritime autonomous systems. This work evaluates AI threats to maritime autonomous systems in situ. The results show that multiple attacks can be used against real-world maritime autonomous systems with a range of lethality. However, the effects of AI attacks vary in a dynamic and complex environment from that proposed in lower entropy laboratory environments. We propose a set of adversarial test examples and demonstrate their use specifically in the marine environment. The results of this paper highlight security risks and deliver a set of principles to mitigate threats to AI, throughout the AI lifecycle, in an evolving threat landscape.

Index Terms

Maritime Cyber Security, Adversarial AI, Maritime Autonomous Systems

I. Introduction

In recent years artificial intelligence (AI) has been utilised to automate many operations and processes abound in academia and industry. One globally crucial industry is maritime, which recognises the plethora of benefits automation could bring over contemporary vessels; these include reduced crew requirements, ease and optimisation of processes, increased crew safety, the possibility of significant operational cost reduction, and emission reduction [19], [39], [56], [58], [65], [78], [89]. Therefore, it is seemingly indisputable that greater automation, and hence the utilisation of maritime autonomous systems (MAS), will play a significant role in the maritime industry in future years. Furthermore, the development of these systems has already been initiated; for example, the work of [68] developed a reduced-crew autonomous ocean-travelling ship. Other work, such as the Mayflower autonomous ship, intends to be fully automated [4]. The Yara Birkeland project, based in Norway, has also seen successes with automated coastal hopping but with open questions around the cost of insurance, cyber security, and contingencies [89]. The Royal Navy is also making great use of uncrewed surface vessels (USV)[1].

Much of the advanced automation will be the product of AI given its proven success, particularly in optimisation, clustering, classification and regression. However, whilst AI has great potential for significant benefits, the nature of subsymbolic AI makes the process of understanding the solution generation mechanism difficult to interpret, yielding a black-box nature, particularly so for deep neural networks relying on billions of parameters, in addition to the high-dimensional phenomena. Therefore, AI has been documented to be a security risk with the term adversarial AI (AAI) coined for the misuse of AI. The fields of AAI and eXplainable AI (XAI) have shown the dangers and safety risks poor development of mission-critical AI can exhibit, in some cases leading to the possibility of fatalities [12]. AI can be used to automate attacks on other technologies, as well as being a technology that can be vulnerable to attack. AI can be attacked with multiple opportunities through the whole AI development process to the deployment of AI technology. What is more concerning is the lack of existing literature which considers AAI in the risk assessments and security of MAS, all whilst we are seeing increasing examples of adversarial AI [38], [67].

In this paper, the authors consider the threat of AI over a multi-domain literature analysis to parameterise AAI specific tests designed to strengthen MAS development. First, we consider the threats of today and the future that AI in the maritime environment may face. Given that AI for both land-based and air-based operations overlap but are not identical, this indicates that the challenges brought on by a marine environment (e.g., water distortion and reflections) will also have its own unique subset of challenges. By examining each class of MAS AI for vulnerabilities to AAI during its life cycle, the authors are able to theorise a set of test cases. These can then be used to increase safe, reliable, and trustworthy AI solutions for maritime operations. Finally, we propose best practices to secure MAS within maritime environments. Ultimately, this research aims to highlight the fast-approaching dangers of ubiquitous AI/automation in MAS and motivate the inclusion of AAI in MAS risk assessment to mitigate against the dangers to all MAS stakeholders.

This paper offers the following novel contributions:

[1]This includes the Buauza USV, an experimental platform, currently operated by the University of Plymouth

1) A start-to-end life cycle evaluation of AAI threats against maritime autonomous systems in situ;
2) A comprehensive review and evaluation of AI threats to MAS considering the AAI and MAS literature;
3) Case examples to support high-fidelity real-world tests over laboratory environments for AAI;
4) Principles to better secure MAS against AAI and ways to enhance MAS AI security across its life cycle.

The paper is structured as follows: Section II critically reviews existing literature across multiple domains to understand the current state of the art. We consider AI in autonomous maritime systems and then examine the threat of adversarial AI in that context. Section III considers the types of AI used in MAS and the existing threats to these systems. Section IV contains an evaluation and analysis of general adversarial principles to MAS. Finally, we conclude and provide further work in Section V.

II. Background

A. AI in Maritime Autonomous Systems

1) Sensors and Instruments: Shipping is a crucial part of global trade, accounting for around 90% of international trade [32]. Waterborne vessels are also critical for human transport, naval defence, and scientific exploration and monitoring of the seas and inland bodies of water. The successful automation of shipping and other maritime operation and services could bring significant advantages over contemporary vessels. Many of these advantages include reducing costs and increasing safety. For example, having no crew aboard ameliorates human factor errors, safety from dangerous working conditions and adverse weather, captivity/attacks from pirates or criminals, more socially supportive conditions and even a reduction in the transmission of some pathogens. Other advantages include cost savings via not having to employ crew, more storage capacity, cheaper development of vessels (crew facilities and living spaces not required), and more economical and environmentally friendly vessels [6]. Some of these benefits, especially crew physical safety, can be obtained, to a reduced degree, with remote unmanned vessels. However, higher tiers of autonomy are needed to maximise these benefits. Furthermore, remote systems would be susceptible to many cyber security attacks, such as jamming and hacking the remote communication [18].

Maritime autonomy can be categorised into different levels, similar to the way autonomous cars and advanced air mobility (AAM), i.e., drones and aircraft, are defined. For example, the Maritime Safety Committee (MSC) of the International Maritime Organisation (IMO) categorises autonomy into four levels. Level 1 contains vessels with autonomous components which support the vessel's crew. Level 2 vessels also have crew aboard to support operations, but a remote control centre operates the vessel. Level 3 vessels are remotely controlled and unmanned. Level 4 vessels are unmanned and fully autonomous. In this work, we will consider only level 4 systems. Other organisations also provide different autonomous level systems, e.g., [68].

As the work of [77] suggests, autonomous systems consist of perception and control elements. Perception elements can be considered the sensors or systems that collect information to be used by the control elements that control the vessel's actions. Contemporary vessels usually have a number of sensors on board to support crew decisions, therefore, creating an existing framework for autonomous systems (although some sensors may require adaptation in MAS). These sensors and systems include; RADAR (radio detection and ranging) to find, usually large, objects with radio waves. The velocity of the object can be determined with doppler RADARs; object detection can be done with other sectors of the electromagnetic spectrum, such as light detection and ranging (LiDAR), which uses infrared light from lasers - these small wavelengths can detect smaller objects and more accurately detect features but at shorter ranges. Echo sounders can be used in a similar way to RADAR and LiDAR; however, echo sounders use sound pulses to detect underwater objects, such as the depth of the water. Echo sounders can be forwards (or laterally) looking as well as vertical to assist in collision avoidance. Multibeam exchosounders can give a 3D point cloud which can be geolocated with millimetric accuracy using RTK GNSS. Measuring echo return backscatter can give useful data about detected objects that go beyond purely the range and baring, giving details about the nature of the detected surface. CCTV/IR/multispectral cameras can be used to detect close-range objects, such as coastal landmarks or objects in the water, akin to LiDAR; furthermore, multiple cameras can be used to triangulate and detect the range of objects; objects can also be captured in colour at high resolution. An array of microphones can be used to detect audio cues on a vessel but may be disrupted by a lot of audio noise, such as the sound of waves, wind and other vessels. Directional microphone arrays are now available that can indicate the range and baring of remote sounds. These will be essential in the future for autonomous vessel to perceive the direction of sound signals. Automatic identification system (AIS) uses very high frequency (VHF) radio to transmit and receive vessel locations and vessel data. Global navigation satellite systems (GNSS) (such as GPS or Galileo) can support dynamic positioning (DP) systems and location services. Electronic chart display and information system (ECDIS) renders charting information. Vessels can contain weather sensors (barometer, temperature, windspeed etc.). Vessels often contain systems for broadband and 3G/4G/5G, as well as VHF for communication. Cargo supervision systems often host an array of sensors such as internal temperature, humidity, smoke detectors. When considering shipping, sensors for monitoring cargo (e.g., food, gas, oil, passengers) are often specific to the specific maintenance needs of that cargo. Vessels often also contain fault diagnosis and voyage data recorder (VDR) systems that store sensor data for post-incident analyses.

Vessels may also contain specialist sensors unique to the vessel type, which determines its size (e.g., gross tonnage), area of operation (e.g., Arctic), and cargo type (e.g., fertilizer). Vessels could also use AI in airborne, surface or subsurface drones to extend sensor range. Studies also show the benefits of multi-sensor perception systems [77], which increase the accuracy of the available data by using multi-systems, e.g., a small object may not be detected by RADAR but instead detected by the

camera through cross-validation of system data. Multiple sensors can also support an element of redundancy. Additionally, the challenges of detecting objects in their natural environment, for example, both on the surface of the water and subsurface, yields difficulties. The water itself can cause confusing distortions but also presents an arduous environment for the physical devices (e.g., salt corrosion).

2) Artificial Intelligence: In fully autonomous maritime systems, AI is used to support, supplement, or replace crews in automating the operations of the vessel. The AI takes sensor data as input and makes decisions to automate the vessel's processes. Different AI is required for different systems because of the range of tasks required by a fully autonomous marine system. The sensors previously discussed can be used as input features to support AI systems to safely navigate the ocean and maintain the functionality of the vessel. However, there exists an overlap in technology, e.g., in order to avoid objects, one requires a degree of situational awareness. We next consider AI for autonomous systems, as categorised in [68], which consists of several AI technologies connected to a DP that controls the vessel. These technologies include:

- **Situational awareness** (SA) - the SA component is required to determine the vessel's real-time location and the vessel's environment (for example, the detection and range of objects). SA modules may also use natural language processing (NLP) to interpret incoming communications. This AI system can use a number of different types of methods and algorithms, such as convolutional neural networks (CNN), region proposal networks (RPN) and sensors to detect landmark objects and navigational cues such as coastal features or buoys [60]. In addition, the AI system could be supported by other non-AI systems, such as GNSS, to cross-validate the vessel's location.
- **Collision avoidance** - uses SA information and prevents the vessel from colliding with objects. These systems use computer vision object recognition to detect objects (SA module output) and feed into the local autonomous route planning modules to change the vessel's trajectory to avoid a collision. Some AI technologies used are CNNs [8] to locate objects and support vector machines (SVMs) which have previously been used to output a new trajectory to prevent collisions [81].
- **Global autonomous optimal planning modules** - ensure the vessel is travelling along the optimal route; an optimal global route may depend on many objectives such as the quickest route, most fuel-efficient, cheapest route and safest route (e.g., weather, global tensions, piracy). The common algorithms utilised are evolutionary algorithms (EAs) which can evolve optimal high-dimensional solutions, particle swarm optimisation (PSO) and ant colony optimisation (ACO), which use the emergence properties of nature to find optimal solutions [35], [72], [88].

The vessel may also include AI to support specialist tasks such as auto berthing/mooring and engine condition maintenance which assist with the general functionality of the vessel. Other examples include Gaussian processes, neural networks, Bayesian modelling, and active learning that can be used for anomaly detection in autonomous vessels to detect deviations and unexpected events [60].

B. Adversarial AI

The advancement and ever-increasing size of neural networks increase the complexity of applications AI can support. However, as the complexity of the model increases, explainability and hence the interpretability of the model decrease [5]. The lack of explainability for complex models, combined with high-dimensional phenomena and poor security principles, can give rise to adversarial AI. The work of [75] was some of the earliest to recognise that neural networks yield properties that can be vulnerable to adversarial attacks, and a 2023 survey paper identified 32 offensive AI capabilities [54]. Governments globally have begun to recognise the threat; notably, the 2021 U.S. National Security Commission on AI stated, "the U.S. government is not prepared to defend the United States in the coming artificial intelligence (AI) era.". Many other countries are preparing for an Adversarial AI (AAI) wave by developing frameworks which attempt to secure AI systems [11], [59]. Furthermore, academic authors [38] have highlighted that "the number of adversarial attacks will continue to increase in the future as the economic benefits trend". As of now, adversarial AI has been demonstrated in a number of applications to support social engineering/spearfishing [71], biometric spoofing [15], computer vision object recognition [2], [17], [82], malware development avoiding network detection [3], [37], [45], [52], NLP [57], [79], and attacks on cloud APIs [31] to name a few.

We recognise AI can be, and has been, used as an attacking tool, e.g., the automation of conventional cyber attacks, side-channel analysis, creation of deepfake media, OSINT collection and analysis. However, these more active AI threats are outside the scope of this paper. Instead, we focus on the inherent vulnerabilities within AI systems processes (in particular, threats to maritime autonomous systems), and how AAI tests can reveal those vulnerabilities to the developer. The primary adversarial goals of AAI are to attack the confidentiality, integrity and accessibility (CIA) triad for ML processes of AI systems.

1) **Confidentiality** - sensitive data can be used during the training phase of the model and has been shown to be extractable from the model [31], [80]. This is of particular concern to a model which uses sensitive data (such as governments) and personal data (privacy concern). Furthermore, data is one of the most valuable resources in modern times [34] and developing AI can be a long and expensive process which could be bypassed with large financial gain by stealing the intellectual property (IP) of the model. As MAS is a new area of growth globally, competition is significantly high.
2) **Integrity** - often involving the attacker aiming to get the AI system to misclassify an input to a specific target or any other false target, usually, so the system carries out an intended adversarial action such as allowing malicious traffic to

pass through a network AI-based IDS [70]. This is a concern with physical object evasion in mission-critical AI, such as naval mine detection, for which an attack could damage the integrity of the AI.

3) **Accessibility** - this adversarial goal is similar to denial of service type attacks where the attacker usually intends to cause high numbers of misclassification to deem the AI inoperable or cause a serious misclassification such as changing the interpretation of a perturbed stop sign; this should prevent the use and access of the AI [23].

These terms can overlap with the risk and threat commensurate with the application of the system, e.g., attacks on mission-critical AI posing the greatest threat. Before we consider the different types of existing attacks on AI, we introduce some key terms, namely, black-box algorithms and white-box algorithms. We note particular confusion with the term black-box algorithm - in the general AI literature, black-box often refers to the poor interpretability of AI models - that is, given the model architecture, hyperparameters and even raw weight values, the combination of interpreting billions of weight values, makes the algorithm difficult to interpret the inner workings (how a prediction is being made by some instance passed through the model). However, the use of the term black-box in the adversarial AI literature appears to take a slightly different meaning which is that the attackers of the AI systems do not know any of the model's architecture, hyperparameters and weights but merely have access to only a model API input and the final result. In this and future sections, black-box AI will refer to the attacker only having access to the model inputs and outputs. In contrast, white-box will refer to having access to the inputs, outputs and also all the model's parameters and architecture. AAI survey papers use a range of nomenclature to classify attacks.

We now consider some of the most prominent adversarial AI methods disclosed in the AAI literature. We note that a large proportion of these methods are relevant to computer vision, and this is thought of as one of the primary AI concerns of the near future [54]. Adversarial attacks are not limited to the post-deployment stage and can be attacked in earlier development stages of the ML pipeline. We categorise this literature into attacks that are performed pre-deployment and deployment.

In the pre-deployment stage, an attacker is concerned with altering the development of the AI model, known as poisoning attacks. These attacks can have a lasting effect on the model through the rest of the model's lifespan. Some key pre-deployment attacks include manipulating training data; this can be done by poisoning the training dataset. The motives for this attack could be the misclassification for evasion and misclassification to lower the classification rate. This can be done by changing the distribution of the training dataset by modifying feature values or injecting new training samples [7], as well as changing the training labels [9]. Other forms of poising included neural injection to create a back door in the neural network so that, given a specific input, the model achieves the desired nefarious output [23].

The post-deployment methods are more concerned with evasion and inversion attacks. Model extraction and model inversion attacks aim to acquire information about the model. Given the time-consuming and expensive training to collect data, preprocess and train a model, the information could include sensitive data points or information about the model's architecture to steal IP or sensitive data [21], [61]. Furthermore, if one is able to recreate an accurate surrogate model, then one could use the model to create adversarial examples in an offline environment where one is more stealthy and able to avoid the actions being logged.

The most well-documented AAI literature is the evasion methods post-deployment, so we discuss these in detail. One of the earliest methods was the work of [75], which proposed perturbing samples to obtain a misclassification by the ML model during the deployment stage. The change to the sample was a minimisation optimisation problem which intended to make a minimal change (so that it was undetectable by the human eye) but enough to cause a misclassification. In order to optimise this problem, one needs to know the direction of sensitivity (e.g., positive/negative perturbations, perturbations of which features) and the magnitude to perturb (usually as small as possible). The work introduces the L-BFGS algorithm (the acronym is the author's initials), which was a method to solve the problem formulated as,

$$\min \|\delta X\| : f(X + \delta X) \neq Y, \tag{1}$$

where f is the model's cost function, and Y is the true label of the instance X. The work also considers the disputed reasons for adversarial examples. They note that the cause of adversarial samples is the result of linear behaviour in high dimensional space [22]. The authors interestingly note a small perturbation to many features of an instance can be far greater than a larger single feature perturbation (such as the one-pixel attack [74]), which uses differential evolution to evolve solutions which intend to change one single pixel in the image). Although this method was fast to compute, the samples generated were often non-functional. Later, three variants of FGSM were developed; namely, one-step target class, Basic Iterative Method (BIM), and Iterative Least-Likely Class Method (ILCM) [41]. The one-step method changes the Y value in Equation 1 from the true label to the desired class label and sets the equation equal to Y. Therefore the algorithm considered perturbing toward a specific class rather than just away from the true class. The BIM method considers Equation 1, but instead iterates the algorithm over small step sizes which can produce numerous adversarial examples. Finally, ILCM also considers an iterative version of Equation 1 but considers perturbing toward the class with the lowest recognition probability. DeepFool [55] is another tool which uses an iterative process to generate adversarial examples to create samples of an image which eventually crosses the class decision boundary.

Instead of gradient descent methods, Jacobian Based Saliency Maps (JSMA) [63] consider the Jacobian of a function matrix (forward derivative), i.e., how does a feature (pixel) change affect the change of a class probability? The saliency map is commonly used in the XAI literature to detect which pixels are making the most significant contributions to the model's

prediction and hence which input features should be perturbed for a desired effect on model output - this map is usually utilised to craft adversarial samples or can be superimposed over the original image as a heatmap. JSMA allows for source-to-target class adversarial sample creation. Using the Jacobian of the model can allow one to determine the sensitivity of a model for specific inputs, i.e., greater sensitivity means perturbations have larger effects on the model's prediction of a class. To use JSMA, one must calculate the forward derivative matrix, also denoted the Jacobian j of the learned function F. This is defined in the original work as:

$$J_{\mathbf{F}}(\mathbf{X}) = \left[\frac{\partial \mathbf{F}(\mathbf{X})}{\partial x_1}, \frac{\partial \mathbf{F}(\mathbf{X})}{\partial x_2} \right], \tag{2}$$

for inputs x_1, x_2 and $\mathbf{F}(\mathbf{X})$ providing the model output. A salience map $S(\mathbf{X}, t)[i]$ can then be computed with the formulae:

$$S(\mathbf{X}, t)[i] = \begin{cases} 0 \text{ if } J_{it}(\mathbf{X}) < 0 \text{ or } \sum_{j \neq t} J_{ij}(\mathbf{X}) > 0 \\ J_{it}(\mathbf{X}) \left| \sum_{j \neq t} J_{ij}(\mathbf{X}) \right| \text{ otherwise.} \end{cases} \tag{3}$$

This limits the Jacobian to be positive (positive effect of the pixel on classification) to decide if input feature i should be perturbed for an adversarial effect on the model. The work showed not all areas of the search space are equally difficult for crafting adversarial samples and that certain source-to-target class pairs are easier to craft than others. For this attack, only the model's output and inputs (black-box) are required to calculate the Jacobian and create a saliency map. Other attacks include using EAs and PSO to optimise the problem [13] and generative adversarial networks GANs, such as AdvGAN [85], which is used to generate adversarial examples. Furthermore, there exists a range of open-source tools to enact these perturbation attacks.

In this work, we also consider patch-based attacks [10] which are a prevalent evasion attack within the literature. These attacks involve generating a highly salient digital or physical patch that can be applied to the image or physical environment to avoid object detection models [43], [49], [73], [84], [86] and classification models [10] by being more salient than other objects in the image and refocusing the attention of the model to the patch [10]. Cyber-physical patch attacks are often formulated as an optimisation problem akin to perturbation attacks. The problem can be formulated as [49],

$$\arg\max_{P} \mathbb{E}_{x,t,l}[\log \Pr(\hat{y} \mid A(P, x, l, t))], \tag{4}$$

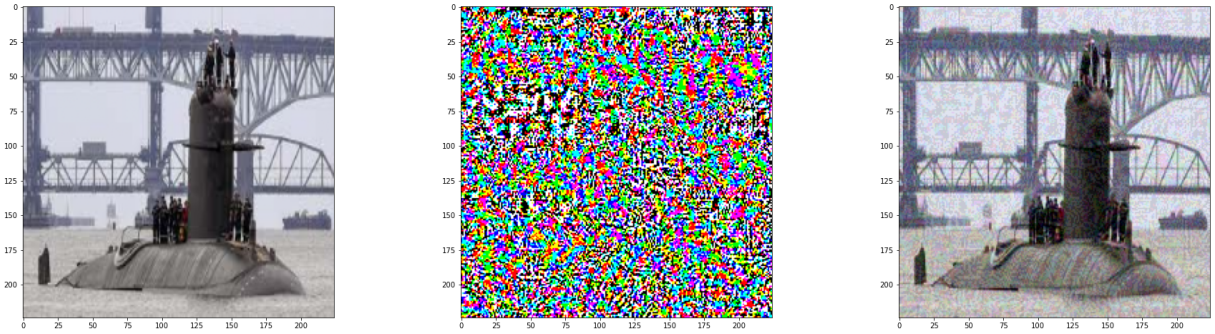
where we aim to generate a patch P where $A(P, x, l, t)$ is the input taking a transformation function A which applies the patch using the original image x, location l and rotation/scaling transformation t. We aim to maximise the loss function of the probability of classifying the input A to true classification label $\hat{y}$. The resulting optimisation formulates an adversarial patch which is superimposed onto the original image and inputted into the model. Common patch-based attacks include DPatch, which is a black-box adversarial patch attack for object detection models, and the work of [27] considers dynamic patches (video), to name a few.

III. Adversarial AI in Maritime Autonomous Systems

We now consider evaluating the threat to ML systems utilised to operate autonomous systems in the maritime environment. Most of the adversarial attacks have been evaluated only in a limited laboratory environment; we aim to evaluate these attacks in the real world where the effects are unknown yet have the greatest potential for impact. This is highlighted as a primary challenge of adversarial AI by adversarial AI authors, including in the literature survey of [46] "[the] need to verify the attack effect in real physical scenarios" and "the current defence technology research lacks the practice in the real world". We demonstrate these attacks in the MAS environment and provide the results and analysis to highlight the effects and practicability of these attacks in the real world. Where appropriate, we visually show some of these attacks in this work. While these threats consider adversarial attacks on AI, conventional cyber security attacks are just as pertinent. Furthermore, some AAI attacks require one to employ AAI and conventional cyber security attacks in unison. Also, conventional security, such as unpatched software and the jamming/spoofing of sensors, can affect both conventional cyber security and AAI-based security. The focus of this work only considers the nascent domain of AAI in MAS, which comprises very limited literature. Notably, in the literature review to date, we found a single publication [87] considering a few theoretical AAI attack possibilities against MAS.

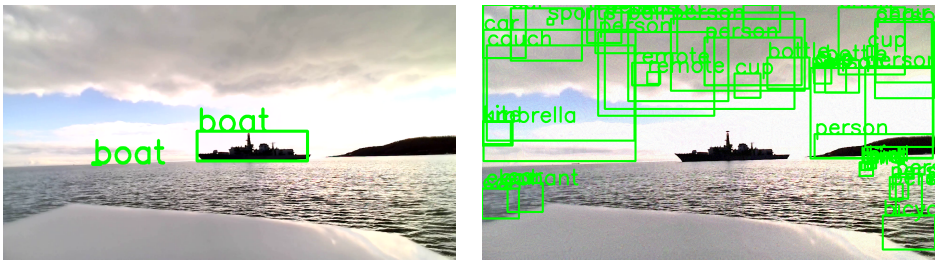
In this publication, we will use Microsoft's failure modes in the ML framework [40] to comprehensively evaluate the type of threats to MAS and provide context to the maritime environment. Microsoft's failure modes in the ML framework categorise AAI attacks into a possible 11 classifications; we list these below. We provide several proof-of-concepts with this list. These are not intended to be an exhaustive set but to demonstrate the usefulness and feasibility of MAS AAI.

Class 1: Model inversion - Even if one is able to secure and protect the knowledge the ML uses to make an output, such as the features used during prediction, one may be able to query the model to determine the model's prerequisite features in a model inversion attack. Whilst this does not threaten the model's functionality, it could be used in the form of reconnaissance to support a future attack. Therefore, the attack is an abuse of the confidentiality of the system.



(a) Original Input (b) Perturb Map (c) Adversarial Sample

Fig. 1: Adversarial perturbation attack sample was generated for a pre-trained MobileNetV2 image classification model. The input sample, a submarine image, is predicted as a submarine by the model, providing a confidence value of 80.15%. The adversarial sample is predicted as a llama by the model, providing a confidence value of 10.74%. The FGSM attack with $\epsilon = 0.1$ was set.



(a) Original Input (b) Adversarial Image

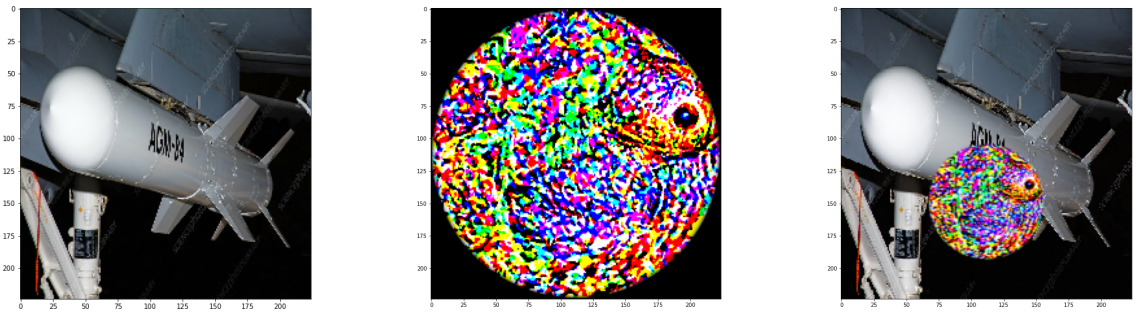
Fig. 2: Adversarial perturbation attack sample was generated using real-world data on a pre-trained FasterRCNN object detection model. The input sample, a warship, is predicted as a vessel by the model, providing a confidence value of 98.87%. The adversarial sample is predicted as multiple incorrect objects by the model with the greatest prediction of person, providing a confidence value of 99.33%. The projected gradient descent attack with the maximum allowable pixel-wise difference between the original image and the adversarial image set at 32. Both the FasterRCNN and YOLOV3 model used in this work was trained on the COCO dataset labelling all ships, marine vessels and boats under the classification of 'boat'.

Class 2: Perturbation attack - In a perturbation attack, the attacker crafts a query which is submitted to the ML algorithm and the ML algorithm actions the attacker's desired response. For example, an attacker could evolve an adversarial data example with an EA, possibly in some underrepresented area/tail of the probability occurrence distribution or a near boundary instance, which causes the collision avoidance system to output a false negative in a busy port and not make an appropriate collision avoidance maneuver. This threat has high consequences; it could be relatively simple to achieve the attack but requires access to modify (or create adversarial inputs and block the legitimate traffic) the input to the ML model - conventional cyber attacks could be leveraged to support this. An example of this attack can be seen in Figure 1 and Figure 2, it should be noted that the accuracy of the adversarial sample appears correlated with the quality of the original image. Therefore using low-resolution cameras may be more susceptible to attack as well as reducing the model classification accuracy.

Class 3: Membership inference - The attacker may be able to infer whether a data instance is a constituent of the training data used to train a model, potentially a breach of privacy.

Class 4: Model stealing - Through querying the model, the attacker may be able to determine information about the model parameters and architecture. With this information, the attack could recreate the model and essentially steal the model/IP. This could save the attacker time and money having to develop the model themselves; this also could be used to turn a black-box model into a white-box model for use with other attack methods. An adversary who steals a MAS model could recreate the model and perform offline attacks (non-logged events) for greater stealth and create more efficient and accurate attacks before applying it to the real online model.

Class 5: Adversarial example in the physical domain - An adversarial example in the physical domain is akin to a perturbation attack. It considers modifying physical properties; for example, an attacker could spoof certain sensor inputs



(a) Original Input (b) Patch (c) Patched Input

Fig. 3: Adversarial patch attack sample generated to attack the common pre-trained ResNet50 image classification model. The input sample, an image of an anti-shipping missile, is predicted as a missile by the model, providing a confidence value of 61.12%. An adversarial patch (b) is provided physically (e.g., as a sticker) or to the input image to change the classification prediction to a spotlight by the model, providing a confidence value of 20.74%. The patch attack [10] method was used. The patch size and shape can be optimised to increase the model classification (although out of scope for this research paper).

(a) Original Input (b) Patch (c) Patched Input

Fig. 4: A real-world adversarial patch attack sample [43] generated to attack the common pre-trained YOLOv3 object detection model. The input sample, an image of two vessels, is predicted as a vessel by the model, providing a confidence value of 99% and 92%. An adversarial patch (b) is provided digitally or physically (e.g., as a sticker) to change the detection prediction to a zebra by the model. Notice the patch can interfere with the prediction and hide nearby objects, such as the second 'boat', i.e. vessel, on its right in the photo. The patch does not need to be significantly large but can be closer to the camera than the evading objects to produce the desired effect.

to confuse the MAS vessel and cause a change in the vessel's trajectory or one could paint a signature on a hostile ship that the searching MAS CNN recognises as a benign object. Example patch attacks can be seen in Figures 3 and 4.

Class 6: Malicious ML provider recovering training data - Akin to a membership inference attack, the attacker may be able to infer the training data used to train a model. The difference from this attack is that the attacker can use queries to derive training data which could potentially be a breach of privacy. The data could contain sensitive information, breach confidentiality, and support model/IP theft.

Class 7: Attacking the ML supply chain - In this attack, the attacker could interfere with elements of the ML lifecycle. For example, capturing training/testing data and retraining new models can be resource-heavy (time and cost); therefore, engineers optimise time by reusing models (transfer learning) and existing datasets. This creates a vector for attackers to manipulate data and models. For example, a model intended to be shared and reused for developing navigational AI could have neurons injected, which cause the vessel to change course given specific spoofed input signals.

Class 8: Backdoor ML - One could create a backdoor utilising the innate poor interpretability of an extensive neural network. For example, one could inject specific neurons or alter existing neuron weights to minimise the noise in the object detection CNN model but render a backdoor so that, given a specific input (a hostile vessel), a model predicts a desired output (misclassify). This could damage both the integrity and confidentiality of the ML model.

Class 9: Exploit software dependencies - This considers the conventional attack surface of software more generally. This could require an attacker to corrupt ML libraries or exploit buffer overflow attacks in the developing software (e.g., labelling application).

Class 10: Reprogramming the ML system - In this attack, the attacker takes the existing ML model and uses it to perform a nefarious task.

Time Stamp	MMSI	Latitude	Longitude	Speed	IMO	Name	Destination
05/01/2021 01:34	209504011	27.09833	-79.88783	15.15792	9517411	CONTSHIP ICE	USMIA
05/01/2021 01:50	209504011	27.05333	-79.88583	15.15792	9517411	CONTSHIP ICE	USMIA
05/01/2021 02:10	209504011	26.99783	-79.88383	15.15792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:04	209504011	26.33928	-79.92229	14.75792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:10	209504011	26.32417	-79.92317	14.65792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:15	209504011	26.31133	-79.92383	14.85792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:20	209504011	26.29667	-79.92483	14.65792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:25	209504011	26.28433	-79.92533	14.75792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:30	209504011	26.269	-79.926	14.85792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:35	209504011	26.256	-79.92667	14.85792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:40	209504011	26.24117	-79.92717	14.85792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:45	209504011	26.227	-79.92767	14.65792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:50	209504011	26.21433	-79.92817	14.55792	9517411	CONTSHIP ICE	USMIA
05/01/2021 06:55	209504011	26.19967	-79.92883	14.75792	9517411	CONTSHIP ICE	USMIA
05/01/2021 07:00	209504011	26.09457	-79.99955	14.65792	9517411	CONTSHIP ICE	USMIA

TABLE I: A sample of poisoned AIS data. The Maritime Mobile Service Identity (MMSI) number was altered (the last two values were modified), and the vessel velocity was modified using the velocity standard deviations. The GPS coordinates can be replaced with other vessels' coordinates. Poisoned data can be used to poison the model during training or spoof existing situational awareness AI.

Class 11: Poisoning attack - Poisoning attacks involve manipulating the training data of the ML model. One could manipulate by injecting new values, new samples, or modifying the feature values or/and labels of the training data. This could be executed to reduce the integrity and availability of the ML model. For example, changing the distribution of the training data or injecting chaff data creates a high misclassification rate and hence reduces the integrity of the system; this could render a denial of service type attack. A MAS-related example could be that for a MAS search vessel, the images or acoustic signals of a certain hostile ship are incorrectly classified. This attack requires access to the training phase of ML development and so is more difficult to achieve than other attacks. It is also more likely that an attack to cause large misclassification for many classes would be noticed during the testing phase of the model's development. An example of AIS poisoning can be seen in Table I.

A. Experimental Setup and Findings

In order to test the proof-of-concept adversarial perturbation and patch attacks, we examined and collected the relevant data from Plymouth sound (UK territorial waters). The vessel used four Omega 1080p cameras which can collect video media that can be fed into an object detection computer vision model at either the vessel or remote centre side. The camera array was mounted on a manned vessel, but in a way that it would have the same view of its immediate surroundings as a USV would. Refer to the figure captions for the specific models used and parameters to generate relevant attacks.

The primary findings of the study showed that the lab-developed attack methods worked well in a controlled environment. However, when performing these same attacks in a complex and dynamic environment like the sea, the effectiveness of the attacks varied more significantly. Just as the type of AI most effective depends on the environment and application of the model, so does the type of attack. The quality of the onboard cameras made object detection and hence evasion more difficult. The object's range from the camera influences the model's effectiveness and evasion. At sea, water distortion, water on the lens, and vibrations of the vessel also added to this effect. Lighting was another important variable where the position of the natural light could cause difficulties. One can observe in Figure 2a and Figure 4 different effects of light taken within an hour window on both the model accuracy and attack accuracy. We also consider the possible application of these attacks, for example, perturbation attacks, which would require the precise distortion of the input for misclassification - this would require access to the model input, which could be challenging in a marine environment. Furthermore, the generation time of the perturbation map for many attacks would generate significant delays to the input image - making it an unlikely vector of attack until more sophisticated and faster methods can be developed. This additional complication means one can not be certain of the effects and limitations of the AAI without evaluating the AI in the natural environment and application of the AI.

Other attacks, such as the patch attack seemed far more likely to be used in a real-world attack. For example, the patch could be physically placed on or near an object and surrounding objects (even objects not covered by the patch) would appear hidden to the model. This is potentially a way for attackers to evade and hide from object detection models. The patches also have a degree of transferability between models. However, from experimentation, the size, and placement (camera-angle-distance relationship) alters the effect of the patch attack. The strength of the patch detection and distance from other objects also affects this hiding/evasion property of nearby objects. Patch attack of an image classifier is easier to achieve than of an object

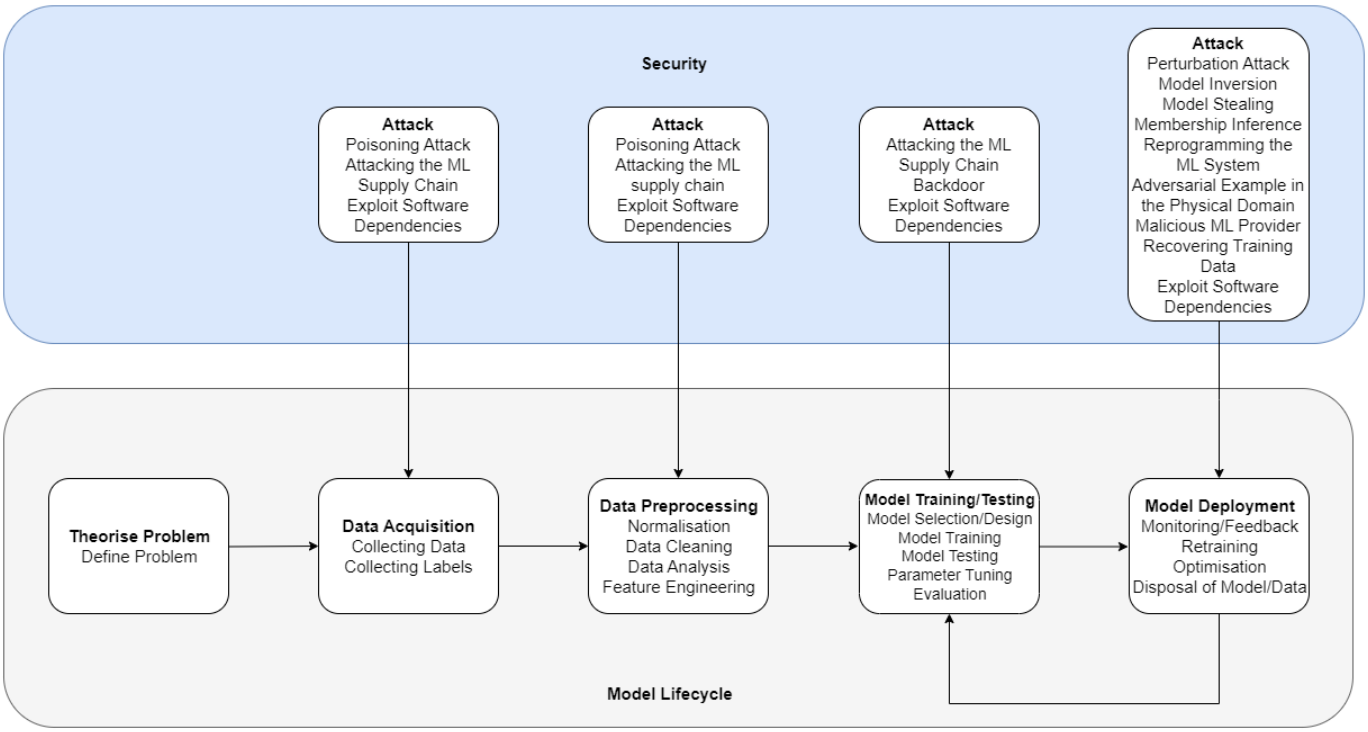


Fig. 5: The machine learning lifecycle and adversarial AI attack types.

detection model but less practical as MAS are more likely to use object detection (for detecting multiple objects in frame) than image classification. For many of these reasons, we strongly advocate the testing of these methods and future novel methods in a more dynamic environment to gauge the real-world impacts of such attacks.

B. AI Lifecycle

The ML development process has a whole lifecycle, denoted MLOps (analogous to DevOps shorting of development operations), commonly understood as defining the problem, data acquisition, data preprocessing, model training, model testing and post-deployment ops. Different attacks can occur at different stages in the development of the model, so each stage and transfer between stages are vulnerable. One could corrupt the environment during the early stages of data collection, manipulate the data preprocessing functions, exploit the hardware (particularly GPUs and CPUs) during training, obfuscate backdoors in transfer learning models and craft adversarial samples during deployment. In Figure 5, we illustrate the process of the ML pipeline and the possible known types of attacks discussed previously.

The attacks are considered at the various common stages of ML development; however, it is possible for other attacks to be performed in the interim stages of the ML cycle and for vulnerabilities to exist during the ML stage transition. For example, a data poisoning attack could occur when recalled by the training software immediately before model training. We can also see from the diagram that the majority of attacks are focused on the predeployment stage of the model's lifecycle.

IV. AI Security Principles for MAS

This section considers existing AAI attacks in maritime autonomous systems (Section III) to create principles to secure maritime autonomous systems. Based on the eleven attack categories, we propose seven secure MAS principles, each with the objective of mitigating the respective AAI attack threat.

There is no one size fits all method of adversarial defence and risk assessments should be considered at the beginning of the ML development process to determine the types of risks, threats, data and the use case of the AI system. It should also be noted that whilst following the suggested principles for maritime autonomous systems provides a degree of security, this will not completely secure the systems from all possible attacks; one reason for this is that a model would need to produce a safe mapping from output to all possible inputs which is an NP-hard problem. Furthermore, the adversarial AI threat is a fast-evolving landscape, and it is likely the case that novel threats will be detected over the coming years. These AAI principles have been generated by the authors for MAS AI based on the findings of this work and aim to reduce the real threat surface this industry faces. These principles for secure AI in maritime autonomous systems are as follows:

1) **Enforce strong conventional cyber security principles** - In addition to strong adversarial AI defences, conventional security methods can complement AI security. This is why the first principle is to create strong conventional cyber

defences. At this early stage in AI development, attacks utilising poor conventional security practices (e.g., unpatched ML libraries) are the most likely vectors of attack. Good countermeasures include blocking bad Internet Protocol (IP) addresses, using CAPTCHA before inputs can be made and throttling/limiting queries to the model. Log inputs and events. Ensure ML libraries and systems are patched and up to date. Limit user access to the model and implement least privilege authority practices. Secure acquisition/storage/transport/decommissioning of model and data can prevent transfer learning/data/model poisoning-based attacks. These strong conventional security measures can protect against Reprogramming ML, attacking the supply chain, and exploiting software dependencies threats.

2) **Develop risk assessment/security assessment before starting ML projects** - Whilst this is a property of enforcing strong conventional cyber security, it was too important not to have its own principle. Consider the application of the ML, the types of ML and their associated vulnerabilities and the generation process to develop a risk assessment. Consider who, why and how an attack might benefit from attacking the application. Mission-critical and security-sensitive AI would likely require a more secure approach to AI development. Furthermore, some applications may further increase risk by utilising processes such as continual learning, which prove additional attack vectors for attackers [59]. A real-world example of this attack was the Microsoft Tay Twitter-trained bot in the first few hours of deployment [83]. Moreover, whilst convenient, the reuse of models (transfer learning/AI re-purposing) and data increases the opportunities for exploitation.
3) **Maximise the model's robustness** - Maximising model robustness reduces the attacking space available to prevent perturbation and adversarial examples. The exploitation of model robustness is one of the simplest of attacks to implement given the scalability of the attack (many models are not robust against all possible adversarial space. It is also possible to use the attack to cause a failure (sometimes not so obvious) of mission-critical systems. Maximising the model's robustness should also provide the additional benefits of protecting against accidental adversarial attacks/errors. Furthermore, the work of [59] also considers forcing model architecture and capacity proportionate to training data to improve robustness and reduce unnecessary feature space whilst covering the distribution of training data.
4) **Maximise explainability and insight for trusted developers and minimise for untrusted users** - Explainability should play a significant role in supporting the development of adversarial AI defences in the coming years. Having greater explainability provides many benefits, but in the context of security, having a better understanding of the ML system's decision processes can support; locating poisoned models, the system limitations, transferability, robustness, and trustworthiness to enhance the security of the system. However, this knowledge could also be used to find and exploit weaknesses, such as locating adversarial space. Therefore one should limit the explainability outputted by the model to untrusted users as well as the technical details of the model, e.g., parameter values and model architecture.
5) **Regulate the input and output of the model** - This principle ties in with revealing too much information about the model, which could be used for nefarious activities against the model, e.g., one could avoid revealing exact probabilities of detection for a classification model to prevent some gradient-based attacks. Regulating the input of the model can prevent adversarial queries from successfully triggering backdoors or exploiting the model [59].
6) **Recognise the exploitation of the model and understand the risks of exploitation** - Having indicators of compromise for the model will not stop adversarial attacks from happening but could allow one to identify and isolate threats. Understanding the effects of a compromised system will allow one to understand the risks and develop effective tailored security approaches in depth.
7) **Sensor redundancy/harmonisation and data correlation** - Utilising multiple fused sensor inputs can be used to bring assurance to situational awareness modules. For example, relying exclusively on a single sensor to deduce the presence or absence of objects is likely to increase the ease of attack. However, sensor fusion of the CV, LiDAR, forward-looking sonar, AIS, and RADAR, all feeding into a navigational AI system, would increase the overall robustness of the system. The requirement of fooling multiple sensors would be exponentially more challenging to mount a successful attack than that of a single sensor, and anomalous behaviour becomes more apparent if not all sensors are fooled, which could lead to a lower confidence weighting provided to data which significantly differs/appears adversarial from other sensors' results.

A. Countermeasures

This section details possible countermeasures which could be used to implement the six principles proposed above to protect MAS AI against AAI; these are summarised in Table II. We first consider adversarial training. Adversarial training requires synthesising adversarial samples from a model and using those adversarial samples as training data to train the model to produce an element of model robustness against adversarial attacks [22], [75]. The samples can be iteratively generated by retraining the model and regenerating adversarial training data [36]. By creating new samples, we increase the distribution of the dataset for the robustness and accuracy of the model. The first instance of adversarial training was [75], which used FSGM to create and then inject samples into the training data. Many variations and improvements of adversarial training exist, such as using GANs [33], [42], [51]. DNN verification tools can be used to locate adversarial samples [66]. It is worth noting to search all the sample space is an NP-hard problem. The three-step null label method blocks the transferability of attacks between models. The method adds a new null label to a one-hot encoding, and the network is trained with some adversarial samples which are labelled as null, therefore if the model input is classified highly as null, this indicates an adversarial input [28] and reduces adversarial attacks happening between models.

AAI Attack	Defences
Perturbation Attack	Adversarial training, Regularisation, Ensemble methods, Input validation and manipulation/preprocessing, Gradient masking, Model distillation, Adversarial sample detection, Explainability
Poisoning Attack	Regularisation, Ensemble methods, Input validation and manipulation/preprocessing, Explainability
Model Inversion	Input validation and manipulation/preprocessing, Adversarial sample detection, Explainability, Preventing information loss
Membership Inference	Input validation and manipulation/preprocessing, Adversarial sample detection, Explainability, Preventing information loss
Model Stealing	Preventing information loss
Reprogramming the ML	Regularisation, Ensemble methods, Gradient masking, Model distillation, Explainability, Preventing information loss
Adversarial Example in Physical Domain	Adversarial training, Regularisation, Ensemble methods, Input validation and manipulation/preprocessing, Gradient masking, Model distillation, Adversarial sample detection, Explainability
ML Provider Recovering Training Data	Regularisation, Ensemble methods, Input validation and manipulation/preprocessing, Gradient masking, Model distillation, Explainability, Preventing information loss
Attacking the ML Supply Chain	Strong conventional cybersecurity practices
Backdoor ML	Regularisation, Ensemble methods, Input validation and manipulation/preprocessing, Model distillation, Explainability, Preventing
Exploit Software Dependencies	Strong conventional cybersecurity practices

TABLE II: The associated defensive countermeasures to prevent exploitation from each adversarial attack. There is much overlap between the defences resulting in the effect of the sum of multiple defensive measures being greater than its individual constituents.

Further countermeasures include regularisation. Regularisation is used in ML to prevent overfitting of a model during training (adding a penalty, i.e. regular term, to a cost function); this can reduce the possible adversarial attack space by making the model more robust to small perturbations. Methods of regularisation include feature pruning to prune activations and neurons from a network [16]. Neuron dropout can be used during model training to stochastically remove neurons which can prevent overfitting on small datasets and potentially remove a backdoor [48]. Other methods include adding a layer of random noise to the model after the input layer so that during forward propagation, the noise creates slightly different outcomes to make the model more robust against small permutations [50].

Ensemble methods is a term used to represent the combination of multiple ML models constituting an overall model. Common methods include using gradient boosting such as XGBoost [14]. This method can reduce the likelihood of training data poisoning as the individual models are trained on different datasets; therefore, when combining the models, the good models can reduce the effect of the poisoned models [44]. It is also possible that adversarial samples are fewer with a greater distribution of training data.

Input validation (or sanitation) and manipulation/preprocessing, which can control the data going into the model can help prevent attacks. Input reconstruction can be used to remove the adversarial effect from input data analogous to input sanitisations to prevent Structured Query Language (SQL) attacks. Input reconstruction was suggested in the work of [24], which proposed transforms applied to input images before making model predictions (clipping, JPEG compression, rescaling depth, etc.). Feature compression/data compression, as in ComCNN [30], can be used to reduce the feature depth of the input, e.g., reduce the colour depth of pixels and increase robustness at the cost of reduced input accuracy. Inputs could also be filtered, smoothed, and have random noise applied on input to sanitise the data. Regression analysis can be used to locate data outliers during input

[29]. Image preprocessing (such as random image padding) can be used to prevent a backdoor attack from being triggered. Input denoising works by attempting to remove noise from the input. Tools include (high-level representation guided denoiser HGD [47]). GANs can be used to clean data by recreating a similar image to the input. MagNet [53] and defence-GAN [69] are tools that can also be used to recreate input images with reduced adversarial noise on a more similar manifold to the benign data.

A countermeasure that would be useful against the perturbation attacks shown in this paper, and others, is gradient masking. Gradient masking can reduce the likelihood of an attacker acquiring the model's gradient and hence reduce against gradient-based adversarial attacks [20]. There is no reduction in the adversarial sample size. However, this attack aims to make it more difficult for white-box probing, by masking the useful gradient, at finding these samples. The effect works by creating less smooth (sharper) boundaries for classification. Gradient masking methods include model distillation and dropout.

Model distillation requires training a smaller, less complex model based on the original complex model. This reduction/compression in model complexity, whilst maintaining similar model accuracy, can help prevent adversarial attacks by creating a model with smoother loss and hence less sensitive to small perturbations. [64]. The original output is used as a soft label, and the original label is used as a hard label.

Adversarial sample detection can help protect MAS AI against adversarial AI via input monitoring. Instead of sterilising the input, one could attempt to detect if the input is an adversarial sample before being accepted or rejected by the model. The input can then be determined as adversarial or benign before being decided whether to be entered into the model. For example, these detection models can be created as a binary classifier and determine if the input follows a similar distribution to the training data [76].

Explainability covers a number of terms, such as trustworthiness, causality, transferability, and informativeness which all support the understanding and hence the security of the ML models. Explainability is a hot topic, and many methods exist to support model explainability. Improving AI explainability is critical as this sector develops AI for mission-critical operations.

Preventing information loss considers the threat of model stealing in a few ways. To protect against data stealing, one could use PATE [62], which splits the training data into subsets and trains multiple models on the subsets before the models are combined, and the systems vote on the predicted outcome. Watermarking can also be used to place a unique watermark in the model, which can be evaluated to determine if it was stolen [1].

V. Conclusion

This work has provided an evaluation of AI security in maritime autonomous systems. A literature review revealed the potential vulnerabilities in MAS AI that could be exposed through a set of adversarial AI test cases strategically designed to test AI used in MAS operations. However, this study of the current state-of-the-art MAS security has also highlighted the inherent vulnerabilities of only testing adversarial AI in limited laboratory environments. Given the extreme differences in marine environments based on location, weather, and time of day, it is also clear that any AAI dataset must also be evaluated in a real-world environment to be truly useful and cyber-resilient maritime AI. After the evaluation of these results in situ, we developed a series of secure AI in MAS principles which can be used to mitigate these threats across the AI's lifecycle.

In further work, we recognise the limited preparation and understating of AAI in MAS technologies by developers, security professionals, and marine regulators. Therefore, knowledge could be disseminated by the secure AI principles and AAI employee training. We would also consider the evaluation of other attacks and their associated defences in the maritime environment (in a range of conditions); we would then consider the effects of underwater distortion etc. Further, we aim to evaluate a range of real-world AI (existing commercial and military AI systems) against AAI - this will allow one to gauge the secondary effects of the attack too, e.g., if one interferes with the CV object detection, how would that impact the collision avoidance module of a vessel? As well as evaluate the effectiveness of AAI and defence in a complex and dynamic environment. Furthermore, we would like to consider the probability of each attack in a maritime autonomous environment with some attacks more effective and likely than others in the real-world environment. As we see greater accessibility of AI, we are likely also to see an increase in the misuse of AI (e.g., AI to support clandestine/smuggling operations) as well an increase in the exploitation of AI systems. The importance of the security of AI is increasing with its use in mission-critical systems (e.g., we are seeing increasing use of militaries using maritime autonomous systems [25], [26]). Whilst many of these AAI attacks have not been utilised in the real world, as the potential financial gain of these attacks increase and the increased use of AI in mission-critical systems, adversaries will look to exploit these methods and the requirement to prepare for the fast-evolving AAI threat landscape is today.

VI. Acknowledgments

This work was supported by the Turing's Defence and Security programme through a partnership with the UK government in accordance with the framework agreement between GCHQ & The Alan Turing Institute. The authors would also like to to thank the University of Plymouth for their use of their autonomous fleet in order to collect real world data.

REFERENCES

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1615–1631, 2018.

[2] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[3] Abdullah Al-Dujaili, Alex Huang, Erik Hemberg, and Una-May O'Reilly. Adversarial deep learning for robust detection of binary encoded malware. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 76–82. IEEE, 2018.

[4] Mark Anderson. Bon voyage for the autonomous ship mayflower. *IEEE Spectrum*, 57(1):36–39, 2019.

[5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[6] Humayun Rashid Askari and Mohammad Nazir Hossain. Towards utilising autonomous ships: A viable advance in industry 4.0. *Journal of International Maritime Safety, Environmental Affairs, and Shipping*, 6(1):39–49, 2022.

[7] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, 2006.

[8] Carlos Bentes, Anja Frost, Domenico Velotto, and Bjoern Tings. Ship-iceberg discrimination with convolutional neural networks in high resolution sar images. In *Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, pages 1–4. VDE, 2016.

[9] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pages 97–112. PMLR, 2011.

[10] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[11] B Caroline, B Christian, B Stephan, B Luis, D Giuseppe, E Damiani, H Sven, L Caroline, M Jochen, Duy Cu Nguyen, et al. Securing machine learning algorithms, 2021.

[12] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

[13] Jinyin Chen, Mengmeng Su, Shijing Shen, Hui Xiong, and Haibin Zheng. Poba-ga: Perturbation optimized black-box adversarial attacks via genetic algorithm. *Computers & Security*, 85:89–106, 2019.

[14] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[15] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020.

[16] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[17] Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems*, 31, 2018.

[18] Cunlong Fan, Krzysztof Wróbel, Jakub Montewka, Mateusz Gil, Chengpeng Wan, and Di Zhang. A framework to identify factors influencing navigational risk for maritime autonomous surface ships. *Ocean Engineering*, 202:107188, 2020.

[19] Andrzej Felski and Karolina Zwolak. The ocean-going autonomous ship—challenges and threats. *Journal of Marine Science and Engineering*, 8(1):41, 2020.

[20] Joachim Folz, Sebastian Palacio, Joern Hees, and Andreas Dengel. Adversarial defense based on structure-to-signal autoencoders. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3568–3577. IEEE, 2020.

[21] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[23] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[24] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[25] Aiden Hall. Autonomous minehunter to trial uncrewed operations in the gulf, Feb 2023.

[26] Aiden Hall. Dstl and dasa research underpins royal navy maritime autonomy, Jan 2023.

[27] Shahar Hoory, Tzvika Shapira, Asaf Shabtai, and Yuval Elovici. Dynamic adversarial patch for evading object detection models. *arXiv preprint arXiv:2010.13070*, 2020.

[28] Hossein Hosseini, Yize Chen, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Blocking transferability of adversarial examples in black-box learning systems. *arXiv preprint arXiv:1703.04318*, 2017.

[29] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 19–35. IEEE, 2018.

[30] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6084–6092, 2019.

[31] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527. IEEE, 2019.

[32] Pablo Kaluza, Andrea Kölzsch, Michael T Gastner, and Bernd Blasius. The complex network of global cargo ship movements. *Journal of the Royal Society Interface*, 7(48):1093–1103, 2010.

[33] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[34] Joanna Kessler. Data protection in the wake of the gdpr: California's solution for protecting" the world's most valuable resource". *S. Cal. L. Rev.*, 93:99, 2019.

[35] Heesu Kim, Sang-Hyun Kim, Maro Jeon, JaeHak Kim, Soonseok Song, and Kwang-Jun Paik. A study on path optimisation method of an unmanned surface vehicle under environmental loads using genetic algorithm. *Ocean Engineering*, 142:616–624, 2017.

[36] Sungrae Kim and Hyun Kim. Zero-centered fixed-point quantization with iterative retraining for deep convolutional neural network-based object detectors. *IEEE Access*, 9:20828–20839, 2021.

[37] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. Adversarial malware binaries: Evading deep learning for malware detection in executables. In *2018 26th European signal processing conference (EUSIPCO)*, pages 533–537. IEEE, 2018.

[38] Zixiao Kong, Jingfeng Xue, Yong Wang, Lu Huang, Zequn Niu, and Feng Li. A survey on adversarial attack in the age of artificial intelligence. *Wireless Communications and Mobile Computing*, 2021, 2021.

[39] Lutz Kretschmann, Hans-Christoph Burmeister, and Carlos Jahn. Analyzing the economic benefit of unmanned autonomous ships: An exploratory cost-comparison between an autonomous and a conventional bulk carrier. *Research in transportation business & management*, 25:76–86, 2017.

[40] Ram Shankar Siva Kumar, David O Brien, Kendra Albert, Salomé Viljöen, and Jeffrey Snover. Failure modes in machine learning systems. *arXiv preprint arXiv:1911.11034*, 2019.

[41] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[42] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

[43] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019.

[44] Deqiang Li and Qianmu Li. Adversarial deep ensemble: Evasion attacks and defenses for malware detection. *IEEE Transactions on Information Forensics and Security*, 15:3886–3900, 2020.

[45] Deqiang Li, Qianmu Li, Yanfang Ye, and Shouhuai Xu. Arms race in adversarial malware detection: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–35, 2021.

[46] Hongshuo Liang, Erlu He, Yangyang Zhao, Zhe Jia, and Hao Li. Adversarial attack and defense: A survey. *Electronics*, 11(8):1283, 2022.

[47] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1778–1787, 2018.

[48] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 273–294. Springer, 2018.

[49] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

[50] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[51] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[52] Davide Maiorca, Battista Biggio, and Giorgio Giacinto. Towards adversarial malware detection: Lessons learned from pdf-based attacks. *ACM Computing Surveys (CSUR)*, 52(4):1–36, 2019.

[53] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[54] Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, et al. The threat of offensive ai to organizations. *Computers & Security*, page 103006, 2022.

[55] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[56] D Morris. Worlds first autonomous ship to launch in 2018, 2017.

[57] John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020.

[58] Ziaul Haque Munim. Autonomous ships: a review, innovative applications and future maritime business models. In *Supply Chain Forum: An International Journal*, volume 20, pages 266–279. Taylor & Francis, 2019.

[59] Kate S. NCSC. Introducing our new machine learning security principles, Aug 2022.

[60] Abraham Noel, K Shreyanka, K Gowtham, and K Satya. Autonomous ship navigation methods: a review. In *Proceedings of the Conference Proceedings of ICMET OMAN*, 2019.

[61] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963, 2019.

[62] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[63] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[64] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[65] Thomas Porathe, Johannes Prison, and Yemao Man. Situation awareness in remote control centres for unmanned ships. In *Proceedings of Human Factors in Ship Design & Operation, 26-27 February 2014, London, UK*, page 93, 2014.

[66] YG Qian, Xi-Ming Zhang, Bin Wang, Wei Li, Jian-Hai Chen, WJ Zhou, and Jing-Sheng Lei. Towards robust dnns: a taylor expansion-based method for generating powerful adversarial examples. *CoRR*, 2020.

[67] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019.

[68] Rolls Royce. Remote and autonomous ships. *AAWA Position Paper*, 2016.

[69] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[70] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. Adversarialuscator: An adversarial-drl based obfuscator and metamorphic malware swarm generator. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2021.

[71] John Seymour and Philip Tully. Weaponizing data science for social engineering: Automated e2e spear phishing on twitter. *Black Hat USA*, 37:1–39, 2016.

[72] Chang Hui Song. Global path planning method for usv system based on improved ant colony algorithm. In *Applied Mechanics and Materials*, volume 568, pages 785–788. Trans Tech Publ, 2014.

[73] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018.

[74] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[75] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[76] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[77] Sarang Thombre, Zheng Zhao, Henrik Ramm-Schmidt, José M Vallet García, Tuomo Malkamäki, Sergey Nikolskiy, Toni Hammarberg, Hiski Nuortie, M Zahidul H Bhuiyan, Simo Särkkä, et al. Sensors and ai techniques for situational awareness in autonomous ships: A review. *IEEE transactions on intelligent transportation systems*, 2020.

[78] Anastasia Tsvetkova and Magnus Hellström. Creating value through autonomous shipping: an ecosystem perspective. *Maritime Economics & Logistics*, pages 1–23, 2022.

[79] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

[80] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 36–52. IEEE, 2018.

[81] Jia Wang, Yang Xiao, Tieshan Li, and CL Philip Chen. A survey of technologies for unmanned merchant ships. *Ieee Access*, 8:224461–224486, 2020.
[82] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.
[83] Marty J Wolf, K Miller, and Frances S Grodzinsky. Why we should have seen that coming: comments on microsoft's tay" experiment," and wider implications. *Acm Sigcas Computers and Society*, 47(3):54–64, 2017.
[84] Han Wu, Syed Yunas, Sareh Rowlands, Wenjie Ruan, and Johan Wahlstrom. Adversarial detection: Attacking object detection in real time. *arXiv preprint arXiv:2209.01962*, 2022.
[85] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.
[86] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, pages 681–698. Springer, 2020.
[87] Ji-Woon Yoo, Yong-Hyun Jo, and Young-Kyun Cha. Artificial intelligence for autonomous ship: Potential cyber threats and security. *Journal of the Korea Institute of Information Security & Cryptology*, 32(2):447–463, 2022.
[88] Yong Zhang, Dun-wei Gong, and Jian-hua Zhang. Robot path planning in uncertain environment using multi-objective particle swarm optimisation. *Neurocomputing*, 103:172–185, 2013.
[89] Ewelina Ziajka-Poznańska and Jakub Montewka. Costs and benefits of autonomous shipping—a literature review. *Applied Sciences*, 11(10):4553, 2021.