Journal of
*Marine Science and Engineering*

*Article*

# Evaluating the Vulnerability of YOLOv5 to Adversarial Attacks for Enhanced Cybersecurity in MASS

**Changui Lee [1] and Seojeong Lee [2,*]**

[1] Korea Conformity Laboratories, Changwon 51395, Republic of Korea; phdculee@gmail.com
[2] Division of Marine System Engineering, Korea Maritime and Ocean University, Busan 49112, Republic of Korea
* Correspondence: sjlee@kmou.ac.kr; Tel.: +82-51-410-4578

**Abstract:** The development of artificial intelligence (AI) technologies, such as machine learning algorithms, computer vision systems, and sensors, has allowed maritime autonomous surface ships (MASS) to navigate, detect and avoid obstacles, and make real-time decisions based on their environment. Despite the benefits of AI in MASS, its potential security threats must be considered. An adversarial attack is a security threat that involves manipulating the training data of a model to compromise its accuracy and reliability. This study focuses on security threats faced by a deep neural network-based object classification algorithm, particularly you only look once version 5 (YOLOv5), which is a model used for object classification. We performed transfer learning on YOLOv5 and tested various adversarial attack methods. We conducted experiments using four types of adversarial attack methods and parameter changes to determine the attacks that could be detrimental to YOLOv5. Through this study, we aim to raise awareness of the vulnerability of AI algorithms for object detection to adversarial attacks and emphasize the need for efforts to overcome them; these efforts can contribute to safe navigation in MASS.

**Keywords:** adversarial attack; perturbed image; YOLO; object classification; MASS

**Citation:** Lee, C.; Lee, S. Evaluating the Vulnerability of YOLOv5 to Adversarial Attacks for Enhanced Cybersecurity in MASS. *J. Mar. Sci. Eng.* **2023**, *11*, 947. https://doi.org/10.3390/jmse11050947

Academic Editors: Marko Perkovic, Lucjan Gucma and Sebastian Feuerstack

Received: 30 March 2023
Revised: 26 April 2023
Accepted: 26 April 2023
Published: 28 April 2023

## 1. Introduction

The impressive advancements made in object detection and classification algorithms that use deep neural networks (DNNs) have significantly affected numerous industries, including the maritime industry, particularly with the development of maritime autonomous surface ships (MASS). By incorporating artificial intelligence (AI) technology, object detection and classification algorithms can detect obstacles in real time, assist in human decision-making during ship navigation [1], and ultimately enable autonomous navigation. However, despite the advancements in object detection and classification algorithms, these systems present vulnerabilities [2]. Adversarial attacks represent one of the most significant security threats to AI systems. The attacks occur during the training phase of deep neural-network algorithms using a training dataset. An attacker adds small and carefully crafted perturbations to the input data that are difficult to detect by humans [3]. Consequently, the model trained with the perturbed data will have significantly degraded performance in its output.

The introduction of MASS has resulted in a new type of threat to the maritime industry in the form of AI-related threats, such as adversarial attacks. As these attacks can compromise the safe operation of MASS, awareness of, and interest in, these threats are crucial for maritime industry stakeholders. In this study, we aimed to emphasize the vulnerability of object detection and classification algorithms to adversarial attacks and the importance of developing strategies to overcome them. To achieve this, we created perturbed images by modifying the settings of four different types of adversarial attack methods. These images were thereafter used as training data for you only look once version 5 (YOLOv5), an object detection and classification algorithm, to simulate adversarial attacks. By evaluating the accuracy of the model trained on these perturbed images, we confirmed the potential catastrophic impacts of adversarial attacks on object detection and classification algorithms. Ultimately, our goal is to increase awareness of the vulnerability of AI algorithms for object detection to adversarial attacks and develop strategies to mitigate their effects, thereby contributing to safe navigation in MASS.

## 2. Background

### *2.1. DNN Algorithms for Object Classification*

Object detection and classification are crucial tasks in computer vision that have seen remarkable progress recently due to advancements in deep learning algorithms. Several popular object detection algorithms have emerged, each with unique strengths and limitations [4]. Recent object detection and classification algorithms, such as region-based convolutional neural networks (R-CNN), mask R-CNN [5], you only look once (YOLO) [6], and single shot detector (SSD) [7], have numerous characteristics in common. Accuracy is among the most significant characteristics of any object detection or classification algorithm, and recent algorithms aim for high accuracy in detecting and localizing objects within an image or video stream. Speed and efficiency are crucial features, as many applications require the real-time processing of considerable data. Algorithms such as YOLOv5 and EfficientDet [8] have demonstrated high processing speeds and resource efficiency, without compromising accuracy. Moreover, mask R-CNN and YOLOv5 use instance segmentation, thus providing visualizations of detected objects with bounding boxes and confidence scores. These can be used to manage occluded items and simplify the interpretation of the output.

YOLOv5 is a powerful and versatile object detection algorithm that has garnered significant attention recently, due to its high accuracy, real-time processing speed, and computational efficiency [6,9]. The model is designed to utilize resources efficiently, and hence is suitable for real-time processing on devices with limited computational resources. It can manage object types, sizes, and orientations, and uses transfer learning to improve generalization to new datasets. YOLOv5 can also visualize detected objects with bounding boxes and confidence scores, making it easy to interpret and understand its output. Among the studies using YOLOv5, Nader Al-Qubaydhi et al. proposed a method [10] for detecting unauthorized unmanned aerial vehicles (UAVs) using YOLOv5 and transfer learning. Their study employed transfer learning to adapt the YOLOv5 framework to the Kaggle drone dataset. They fine-tuned the last three YOLOv5 and convolutional layers to match the number of classes in the dataset and introduced data augmentation techniques to enhance the dataset and improve training. The trained model was evaluated by constructing a number-of-iterations-versus-mAP curve at different points, and the results demonstrated high accuracy in detecting unauthorized UAVs.

### *2.2. AI-Specific Security Threats*

AI systems are becoming increasingly popular across industries for their ability to automate decision-making, improve efficiency, and drive innovation. However, because of increased Internet connectivity and usage, these systems have recently become more vulnerable to cybersecurity threats [11,12].

AI-specific security refers to the unique security risks and challenges that AI systems encounter, and the specific security measures that organizations should implement to protect their systems against such threats. The ISO/IEC TR 24028 technical report [12] provides guidelines for managing the security and privacy of AI systems, thus promoting their safe and responsible use. The report highlights several AI-specific security threats, including adversarial attacks, data poisoning, model stealing, and evasion, privacy, integrity, and denial of service attacks.

Adversarial attacks involve intentionally modifying the input data of a machine-learning model to cause the model to make incorrect predictions. These attacks may vary according to the target machine-learning application, including image classification, speech recognition, and autonomous vehicles. Data poisoning involves deliberately inputting misleading or inaccurate data into an AI system with to manipulate its output. Attackers may introduce inaccurate or biased training data into an AI system to influence its decision-making. Data poisoning can result in AI systems that make biased or discriminatory decisions. Model stealing is a technique by which an attacker attempts to acquire an AI model by analyzing its output and using reverse engineering techniques. This aspect can enable the attacker to replicate the model and use it for malicious purposes, such as creating deepfake images or videos. Evasion attacks are techniques used to evade detection by security mechanisms, such as malware or intrusion detection systems. Privacy attacks compromise the privacy of individuals and organizations by exploiting vulnerabilities in AI systems. Attackers may use these vulnerabilities to access sensitive information or to execute espionage and other malicious activities. Integrity attacks compromise data integrity or AI models by modifying or deleting data. These attacks can result in AI systems that produce inaccurate or unreliable results. Denial of service attacks disrupt the availability of AI systems by overwhelming them with requests or other forms of traffic. These attacks can result in significant disruptions to AI systems and financial losses.

Among security threats, adversarial attacks are particularly significant due to their potential to compromise the accuracy and reliability of AI systems [13]. These attacks are intentional modifications of input data designed to compromise the accuracy and reliability of the output of the AI system. The primary objective of an adversarial attack is to add the minimal perturbation to the input data that can result in the desired misclassification. These attacks pose a severe security threat to critical systems such as those used for medical diagnosis or in autonomous vehicles [14,15]. Adversarial attacks can be categorized into different types based on the purpose of the attack and prior knowledge of the attacker. For instance, based on our assumptions on the knowledge of the attacker, these can be classified as white- or black-box attacks. In a white-box attack, the attacker has complete knowledge and access to the model, including architecture, inputs, outputs, and weights [14]. In contrast, in a black-box attack, the attacker only has access to the inputs and outputs of the model and is unaware of the underlying architecture or weights [16].

The basic principle of an adversarial attack is to generate perturbed data by synthesizing specific noise, indistinguishable from a conventional image [4,14], as depicted in Figure 1. When these perturbed data are input into a trained learning model, they appear as a ship to the human eye; however, the deep learning model will classify them as a lighthouse. Therefore, the purpose of an adversarial attack is to lower the accuracy of the model by causing misclassification.

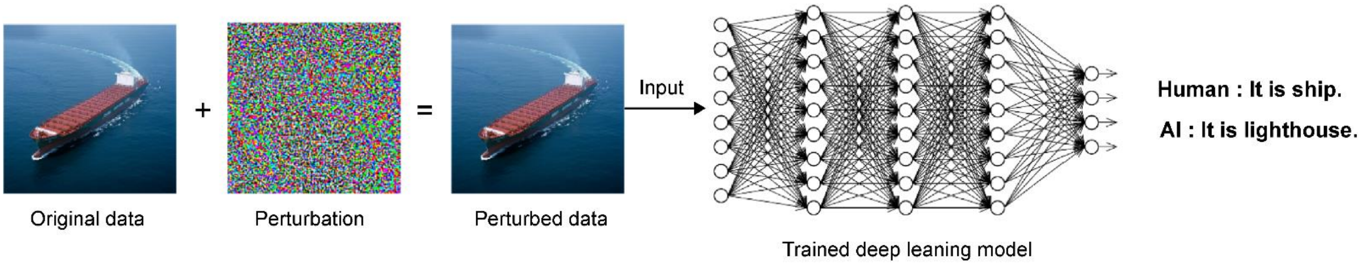

**Figure 1.** Principle of an adversarial attack.

### 2.3. Methods for Generating Adversarial Examples

An adversarial example indicates a perturbed data input specifically designed to induce inaccurate classifications by DNN-based algorithms. Adversarial attacks are a critical issue in machine learning security, and numerous methods are available for generating adversarial examples.

Goodfellow et al. proposed the fast gradient sign method (FGSM) [13]. This is a simple and efficient method in which the gradient of the input data loss function and a small amount of noise in the direction of the gradient are summed to generate perturbed inputs that can be easily misclassified by the AI model. Even though the FGSM is easy to implement, it may not produce the most robust adversarial examples.

Kurakin et al. introduced a powerful variant of the FGSM known as the iterative fast gradient sign method (I-FGSM) [17]. This method applies the attack iteratively with a smaller step size, resulting in more robust adversarial examples. However, the increased complexity and number of iterations may make this method more computationally expensive than the FGSM.

Dong et al. proposed the momentum iterative fast gradient sign method (MI-FGSM) [18] as a further extension of the I-FGSM in which a momentum term was added to the perturbation update rule. This aspect can prevent oscillations and accelerate convergence. However, the increased complexity and number of hyperparameters can make the tuning process more challenging compared with the I-FGSM.

Madry et al. proposed the projected gradient descent (PGD) [19] algorithm as a variant of the I-FGSM in which random perturbations are added to the input at each iteration. This modification can prevent the attack from stabilizing around a local minimum, thus improving the generalization of the adversarial examples. However, as in the previous method, the tuning of the PGD requires more computational resources compared with the I-FGSM.

### 2.3.1. Fast Gradient Sign Method

The FGSM [13] algorithm is a simple and effective method for generating adversarial examples in deep learning models. By adding a minimal perturbation to the input data in the direction of the gradient of the loss function of the input data, the algorithm can cause the model to misclassify the data, even though the perturbed image may appear similar to the original one to a human observer [20]. The algorithm of the FGSM is summarized in Table 1.

**Table 1.** FGSM algorithm.

| | |
|---|---|
| **Input** | An input image x, a target class y, and the size of perturbation $\epsilon$. |
| **Output** | An adversarial example $x'$, misclassified by the deep learning model, with a perturbation size satisfying $kx' - k\infty x \leq \epsilon$, where k is a scalar and $k\infty$ is its L-infinity norm, and $\epsilon$ = $kx' - k\infty x$. |
| **Algorithm** | 1. Calculate the gradient of the loss function J for the input image x:<br>gradient = $\nabla$x J($\theta$, x, y),<br>where $\theta$ represents the parameters of the deep learning model. |
| | 2. Calculate the perturbation by scaling the sign of the gradient with a small $\epsilon$ value:<br>perturbation = sign(gradient) * $\epsilon$,<br>where sign() denotes the sign function, and * represents element-wise multiplication. |
| | 3. Add the perturbation to the input image to obtain the adversaria example:<br>$x'$ = x + perturbation. |
| | 4. Clip the pixel values of the adversarial example to ensure that they remain within the valid range:<br>$x'$ = clip($x'$, 0, 1). |
| | 5. Return the adversarial example $x'$. |

### 2.3.2. Iterative FGSM

The I-FGSM [17] is an extension of the FGSM that generates adversarial examples by applying multiple iterations of the FGSM at a small step size $\alpha$. The algorithm clips the perturbation at each iteration to ensure that its $L\infty$ norm does not exceed the specified size $\epsilon$. The I-FGSM can generate more effective adversarial examples than those generated by the FGSM, particularly when combined with other techniques such as momentum or randomization [20,21]. The algorithm of the I-FGSM is presented in Table 2.

**Table 2.** I-FGSM algorithm.

| | |
|---|---|
| **Input** | An input image x, a target class y, the size of perturbation $\epsilon$, the number of iterations T, and the step size $\alpha$. |
| **Output** | An adversarial example $x'$, misclassified by the deep learning model, with a perturbation size satisfying $kx' - k\infty x \leq \epsilon$, where k is a scalar and $k\infty$ is its L-infinity norm, and $\epsilon = kx' - k\infty x$. |
| **Algorithm** | 1. Initialize the perturbation $\delta$ to zero. |
| | 2. For t = 1 to T:<br>a. Calculate the gradient of the loss function J for the input image x:<br>gradient = $\nabla x\ J(\theta, x + \delta, y)$,<br>where $\theta$ represents the parameters of the deep learning model.<br>b. Add a scaled version of the sign of the gradient to the perturbation:<br>$\delta \leftarrow \delta + \alpha$ sign(gradient).<br>c. Clip the perturbation $\delta$ so that its $L\infty$ norm is at most $\epsilon$:<br>$\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$.<br>d. Update the adversarial example by adding the perturbation to the input image:<br>$x' \leftarrow x + \delta$.<br>e. Clip the pixel values of the adversarial example to ensure that they remain within the valid range:<br>$x' \leftarrow \text{clip}(x', 0, 1)$. |
| | 3. Return the adversarial example $x'$. |

### 2.3.3. Momentum Iterative FGSM

The MI-FGSM [18] is an extension of the I-FGSM that generates adversarial examples by adding a momentum term to the update rule. This term prevents oscillations, resulting in a faster convergence to the adversarial example compared to the previous models. The momentum term is computed using the average of the previous gradients; it is scaled by a factor $\alpha$ to control its contribution to the update. The MI-FGSM algorithm can generate more effective adversarial examples than the I-FGSM, particularly when combined with other techniques such as randomization or ensemble methods [20,22]. The algorithm of the MI-FGSM is summarized in Table 3.

**Table 3.** MI-FGSM algorithm.

| | |
|---|---|
| **Input** | An input image x, a target class y, the size of perturbation $\epsilon$, the number of iterations T, and the decay factor $\mu$. |
| **Output** | An adversarial example $x'$, misclassified by the deep learning model, with a perturbation size satisfying $kx' - k\infty x \leq \epsilon$, where k is a scalar and $k\infty$ is its L-infinity norm, and $\epsilon = kx' - k\infty x$. |
| **Algorithm** | 1. Initialize the perturbation $\delta$ to zero. |
| | 2. For t = 1 to T:<br>a. Calculate the gradient of the loss function J for the input image x:<br>gradient = $\nabla x\ J(\theta, x + \delta, y)$,<br>where $\theta$ represents the parameters of the deep learning model.<br>b. Add a scaled version of the sign of the gradient to the perturbation:<br>$\delta \leftarrow \mu\delta + (\epsilon/T)$ sign(gradient).<br>c. Clip the perturbation $\delta$ so that its $L\infty$ norm is at most $\epsilon$:<br>$\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$.<br>d. Update the adversarial example by adding the perturbation to the input image:<br>$x' \leftarrow x + \delta$.<br>e. Clip the pixel values of the adversarial example to ensure that they remain within the valid range:<br>$x' \leftarrow \text{clip}(x', 0, 1)$. |
| | 3. Return the adversarial example $x'$. |

#### 2.3.4. Projected Gradient Descent

The PGD [19] algorithm is a variant of the I-FGSM that generates adversarial examples by adding random perturbations to the input at each iteration. These perturbations prevent the model from converging to local optima, thus resulting in more diverse adversarial examples. The PGD algorithm clips the perturbation at each iteration to ensure the L∞ norm does not exceed the specified size $\epsilon$. The randomness factor $\delta$ controls the magnitude of the random perturbations and can be adjusted to modify the exploration–exploitation balance. The PGD algorithm can generate more robust adversarial examples than those obtained using the I-FGSM, particularly when combined with other techniques such as ensemble or regularization methods [21,22]. The algorithm of the PGD is presented in Table 4.

**Table 4.** PGD algorithm.

| | |
|---|---|
| **Input** | An input image x, a target class y, the size of perturbation $\epsilon$, the number of iterations T, and the step size $\alpha$. |
| **Output** | An adversarial example x′, misclassified by the deep learning model, with a perturbation size satisfying $kx' - k\infty x \leq \epsilon$, where k is a scalar and k∞ is its L-infinity norm, and $\epsilon = kx' - k\infty x$. |
| **Algorithm** | 1. Initialize the perturbation $\delta$ to zero. |
| | 2. For t = 1 to T:<br>a. Calculate the gradient of the loss function J concerning the input image x:<br>gradient = $\nabla x\, J(\theta, x + \delta, y)$,<br>where $\theta$ represents the parameters of the deep learning model.<br>b. Add a scaled version of the gradient to the perturbation:<br>$\delta \leftarrow \delta + \alpha$ sign(gradient).<br>c. Project the perturbation onto the L∞ ball of radius $\epsilon$:<br>$\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$.<br>d. Update the adversarial example by adding the perturbation to the input image:<br>$x' \leftarrow x + \delta$.<br>e. Clip the pixel values of the adversarial example to ensure that they remain within the valid range:<br>$x' \leftarrow \text{clip}(x', 0, 1)$. |
| | 3. Return the adversarial example x′. |

## 3. Materials and Methods

### *3.1. Problem Setup*

In this study, we aimed to simulate adversarial attacks that manipulate the training dataset to degrade the performance of object detection and classification models. A critical aspect of training AI for industrial use is securing suitable datasets. Although datasets for objects such as dogs and cats are common for object detection and classification, it is difficult to obtain datasets suitable for the maritime industry, such as boats, ferries, and buoys. Attackers take advantage of this and distribute perturbed images that are difficult for humans to detect but contain feature information that causes misclassification. At first glance, the perturbed image may appear to be slightly noisy, but this can be mistaken for optical noise. However, the perturbed image contains feature information that causes misclassification and cannot be removed by methods such as blurring, which are used to remove optical noise. Therefore, the system developer (victim) unknowingly uses the perturbed images as part of their training dataset because they appear normal to the developer. This means that an attack can occur without the system developer even noticing it. If the system developer attempts to remove noise from the training dataset, the feature information that causes misclassification will remain, thus compromising the model's performance. Moreover, because of the plateaued performance, the system developer may mistakenly assume that the model's performance is optimal and use it in real-world scenarios where failures to detect objects would cause misclassifications and precipitate serious accidents.

In AI, research into adversarial attack methods is ongoing, e.g., being investigated as a significant threat to using AI for diagnosing disease [3]. In the maritime industry, such as in the use of MASS, research is being conducted on systems that apply AI to detect and classify objects. However, in the maritime industry, the potential risks are not well-known because such adversarial attacks have not yet been experienced. To address this gap, we conducted experiments to determine the adversarial attack methods that could be most detrimental to object detection and classification models.

### 3.2. Experimental Scenario

In the experiments, we generated perturbed images using various adversarial attack methods and parameter settings and evaluated the accuracy of the models that were trained with these perturbed images. Thereby, we aimed to determine the most effective adversarial attack methods and raise awareness of their potential risks in the maritime industry.

The experimental scenario is divided into four phases, as depicted in Figure 2. In the first phase, the modified Singapore maritime dataset (SMD-Plus) proposed by Kim et al. [9] is pre-processed such that images and annotations are suitable for the YOLOv5 model. In the second phase, an attacker generates the perturbed images. This study assumed that the attacker could generate perturbed images using various methods. Therefore, six pre-trained DNN algorithms and four adversarial attack methods were used to generate the perturbed dataset using Python's PyTorch open-source framework. In the third phase, the system developer collects data for model training and trains the YOLOv5 model [23,24]. During this phase, the attacker deploys a perturbed dataset, and the system developer collects and verifies the data. The slight noise in the perturbed image is considered to be optical noise and passes the verification process. This assumption is fundamental to adversarial attacks [12,13,17–19], and this is the reason for the development of new algorithms to render it more difficult for detection by humans. Therefore, in this scenario, the perturbed image is assumed to pass the verification process without any issues and is included in the training dataset. The adversarial attack occurs during this phase without the explicit intervention of the attacker. In the fourth phase, the trained model is tested on conventional benchmark and perturbed datasets to examine the impact of adversarial attacks [23–27].

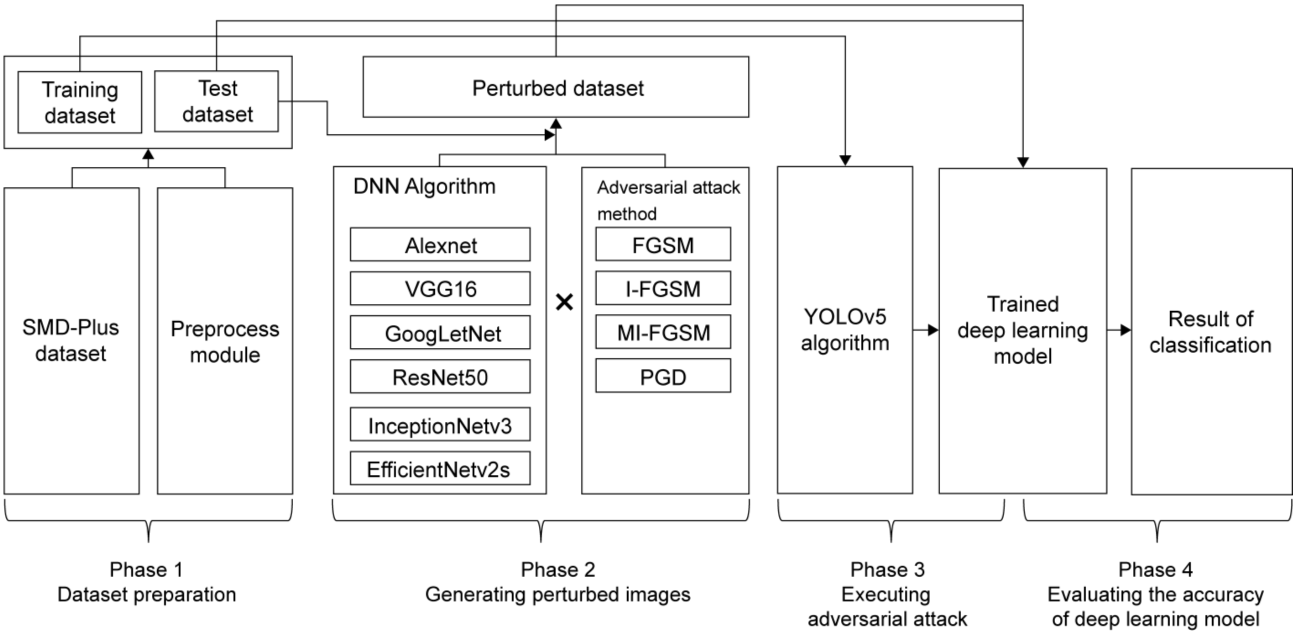

**Figure 2.** Diagram of the proposed experiment.

## 4. Results: Experiment on Adversarial Attacks against YOLOv5

### *4.1. Dataset Preparation*

In domain-specific DNN applications, obtaining a proper dataset for training can be challenging, particularly for maritime environments, due to the scarcity of publicly available datasets and the cost of collecting and annotating images. The SMD dataset provides high-quality videos with labelled bounding boxes for ten types of objects in marine environments. The SMD dataset consists of high-definition videos captured at a resolution of 1920 × 1080 pixels. The dataset is divided into two parts: on-shore videos, consisting of 40 video clips, and on-board videos, consisting of 11 video clips. Additionally, each frame in the video dataset is labeled with bounding boxes and labels. However, this dataset presents some label errors and imprecise bounding boxes; therefore, it is not ideal as a benchmark dataset for object classification. To address this issue, the SMD-Plus dataset [9] was developed to improve the accuracy of bounding box annotations for small maritime objects. Moreover, in the SMD-Plus dataset, visually similar classes were merged to provide more training data for object recognition. Table 5 presents the classification of the training classes in the SMD-Plus dataset.

**Table 5.** Details of SMD-Plus dataset.

| Class | Class Identifier | Number of Objects |
|---|---|---|
| Boat | 1 | 14,021 |
| Vessel/Ship | 2 | 125,872 |
| Ferry | 3 | 3431 |
| Kayak | 4 | 3798 |
| Buoy | 5 | 3657 |
| Sailboat | 6 | 1926 |
| Others | 7 | 24,993 |

Because the SMD-Plus dataset comprises videos and annotations, we split the videos into frame-by-frame images to train YOLOv5. The provided annotations include object classes and the locations of bounding boxes for each video frame. These annotations were converted from the file format developed with the MATLAB ImageLabeler tool into an annotation format suitable for YOLOv5 [9,23]. Notably, 80% of the samples were used as the training dataset; the remaining 20% were used as the test dataset.

### *4.2. Generating Perturbed Images*

Generating perturbed images requires both a DNN algorithm and an adversarial attack method. To assess the effect of adversarial attack methods on deep learning models, we developed perturbed datasets by implementing four different adversarial attack algorithms using PyTorch open-source code and a test dataset as input for six pre-trained models, namely, AlexNet, VGG16, GoogLeNet, ResNet50, InceptionNetv3, and EfficientNetv2s [28]. The objective of this experiment was to investigate the combinations of DNN algorithms, adversarial attack methods, and changes in the $\epsilon$ value that are most detrimental to YOLOv5 by generating perturbed images for adversarial attacks. The DNN algorithm, adversarial attack method, and hyperparameters (including the $\epsilon$ value) are determined by the attackers at the moment they generate the perturbed image. As presented in Table 6, we considered $\epsilon$ as the independent variable varying from 0.01 to 0.3; the other variables were kept constant as control variables. We generated 120 perturbed datasets, one for each combination of the pre-trained model and epsilon value.

**Table 6.** Parameters of the methods used for the generation of adversarial examples.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| Hyperparameters | $\epsilon$ = 0.01, 0.05, 0.1, 0.2, 0.3 | $\epsilon$ = 0.01, 0.05, 0.1, 0.2, 0.3<br>T = 20<br>$\alpha$ = 0.01 | $\epsilon$ = 0.01, 0.05, 0.1, 0.2, 0.3<br>T = 20<br>$\mu$ = 0.001 | $\epsilon$ = 0.01, 0.05, 0.1, 0.2, 0.3<br>T = 20<br>$\alpha$ = 0.01 |

We used an AMD Ryzen 9 5950X processor with 64 GB of main memory and an NVIDIA GeForce RTX 3080 Ti to generate the perturbed images. The runtime of the different methods is reported in Table 7. The execution time was not affected by the $\epsilon$ value.

**Table 7.** Execution time of pre-trained DNN algorithms for the generation of perturbed image.

| | Execution Time (s) | | | |
|---|---|---|---|---|
| | FGSM | I-FGSM | MI-FGSM | PGD |
| AlexNet | 19 | 43 | 22 | 31 |
| VGG16 | 27 | 291 | 132 | 190 |
| GoogLeNet | 21 | 132 | 97 | 128 |
| ResNet50 | 21 | 187 | 102 | 137 |
| InceptionNetv3 | 25 | 207 | 145 | 199 |
| EfficientNetv2s | 30 | 327 | 231 | 312 |

Figures 3 and 4 depict the images generated using the FGSM and PGD with AlexNet with different $\epsilon$ values. When the $\epsilon$ value is small, changes made to the image are not easily distinguishable. However, as the $\epsilon$ value increases, changes become more noticeable. Nevertheless, it is challenging to distinguish whether this is caused by an adversarial attack or simple optical noise.

The results of the changes in the $\epsilon$ value affect not only the addition of noise that can be discerned by the human eye but also the performance degradation of the targeted model. In the following section, we simulate the performance degradation of the targeted model according to the $\epsilon$ value.

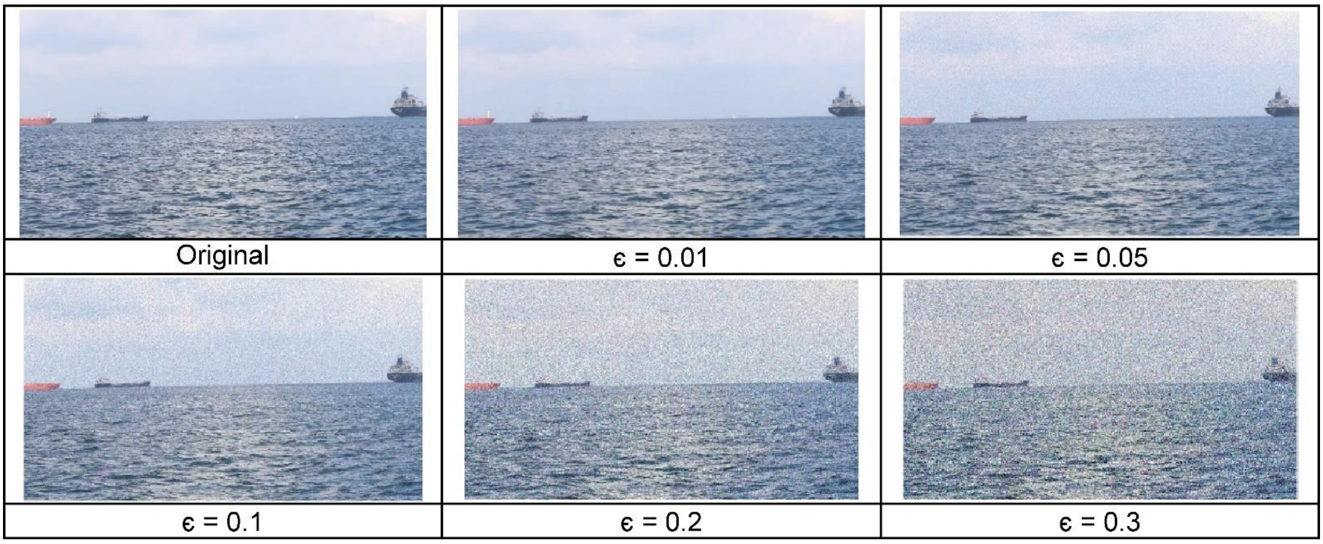

**Figure 3.** Perturbed images using the FGSM method with AlexNet.

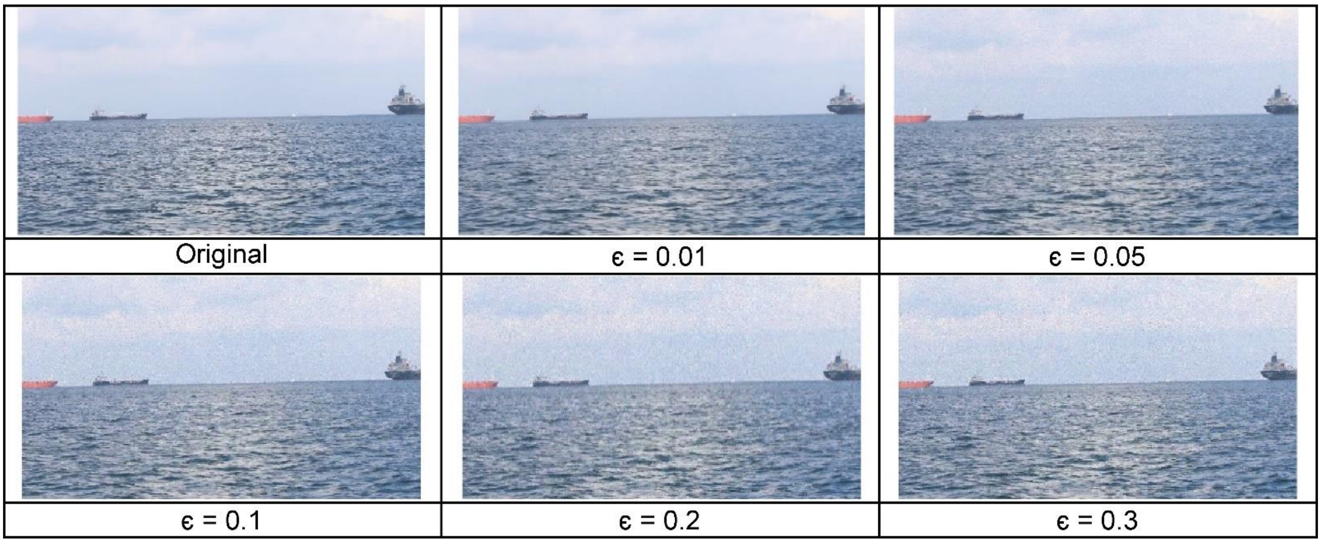

**Figure 4.** Perturbed images obtained using the PGD method with AlexNet.

### *4.3. Simulating an Adversarial Attack on a Deep Learning Model*

We employed transfer learning using the YOLOv5s model as the base model to improve the training speed and accuracy of the object detection and classification model. The YOLOv5 model has four structures, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, according to speed and accuracy. We assumed that YOLOv5s would be more vulnerable to adversarial attacks because of its low number of neural network layers and lower model accuracy. Transfer learning is a common learning method that is used to improve the performance of a model on relatively small datasets. The method uses a pre-trained model trained on a large amount of data as the base model. We used the fine-tuning method, which involves re-training the entire neural network, and adjusted the last convolutional layers of YOLOv5 to match the number of classes in the dataset, increasing them from 3 to 8. We set the training parameters to 100 epochs, learning rate of 0.12, and a batch size of 16 [10,19,22].

To evaluate the robustness of the YOLOv5-based deep learning model, we performed a comprehensive analysis by inputting a single original test dataset and the 120 perturbed datasets generated using various adversarial attack methods. By measuring the change in accuracy under these different scenarios, we aimed to assess the ability of the model to resist adversarial attacks and its overall performance in detecting objects in images.

We used the original test dataset to perform object classification. The YOLOv5s model trained using the SMD-Plus dataset achieved an accuracy of 0.896. The experiment also tested the accuracy of the model for each of the six different pre-trained DNN algorithms using the four adversarial attack methods and five $\epsilon$ values. The results are summarized in Tables 8–13.

**Table 8.** Accuracy of transfer learned model of YOLOv5s for different $\epsilon$ values using AlexNet.

| | **FGSM** | **I-FGSM** | **MI-FGSM** | **PGD** |
|---|---|---|---|---|
| $\epsilon$ = 0.01 | 0.873 | 0.861 | 0.841 | 0.831 |
| $\epsilon$ = 0.05 | 0.810 | 0.791 | 0.837 | 0.776 |
| $\epsilon$ = 0.1 | 0.612 | 0.740 | 0.768 | 0.681 |
| $\epsilon$ = 0.2 | 0.417 | 0.681 | 0.633 | 0.631 |
| $\epsilon$ = 0.3 | 0.132 | 0.671 | 0.491 | 0.614 |

**Table 9.** Accuracy of transfer learned model of YOLOv5s for different ϵ values using VGG16.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| ϵ = 0.01 | 0.831 | 0.822 | 0.811 | 0.818 |
| ϵ = 0.05 | 0.618 | 0.751 | 0.801 | 0.716 |
| ϵ = 0.1 | 0.437 | 0.693 | 0.715 | 0.679 |
| ϵ = 0.2 | 0.105 | 0.651 | 0.631 | 0.678 |
| ϵ = 0.3 | 0.063 | 0.643 | 0.551 | 0.653 |

**Table 10.** Accuracy of transfer learned model of YOLOv5s for different ϵ values using GoogLeNet.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| ϵ = 0.01 | 0.827 | 0.858 | 0.841 | 0.829 |
| ϵ = 0.05 | 0.766 | 0.765 | 0.818 | 0.771 |
| ϵ = 0.1 | 0.551 | 0.750 | 0.761 | 0.728 |
| ϵ = 0.2 | 0.266 | 0.721 | 0.731 | 0.731 |
| ϵ = 0.3 | 0.115 | 0.711 | 0.565 | 0.710 |

**Table 11.** Accuracy of transfer learned model of YOLOv5s for different ϵ values using ResNet50.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| ϵ = 0.01 | 0.841 | 0.849 | 0.833 | 0.838 |
| ϵ = 0.05 | 0.633 | 0.788 | 0.761 | 0.712 |
| ϵ = 0.1 | 0.410 | 0.711 | 0.763 | 0.653 |
| ϵ = 0.2 | 0.160 | 0.667 | 0.622 | 0.614 |
| ϵ = 0.3 | 0.061 | 0.651 | 0.531 | 0.609 |

**Table 12.** Accuracy of transfer learned model of YOLOv5s for different ϵ values using InceptionNetv3.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| ϵ = 0.01 | 0.827 | 0.832 | 0.832 | 0.819 |
| ϵ = 0.05 | 0.568 | 0.736 | 0.776 | 0.718 |
| ϵ = 0.1 | 0.355 | 0.649 | 0.737 | 0.608 |
| ϵ = 0.2 | 0.037 | 0.615 | 0.619 | 0.600 |
| ϵ = 0.3 | 0.011 | 0.601 | 0.456 | 0.587 |

**Table 13.** Accuracy of YOLOv5s for different ϵ values using EfficientNetv2s.

| | FGSM | I-FGSM | MI-FGSM | PGD |
|---|---|---|---|---|
| ϵ = 0.01 | 0.830 | 0.841 | 0.845 | 0.809 |
| ϵ = 0.05 | 0.568 | 0.776 | 0.767 | 0.711 |
| ϵ = 0.1 | 0.437 | 0.677 | 0.691 | 0.638 |
| ϵ = 0.2 | 0.055 | 0.663 | 0.611 | 0.627 |
| ϵ = 0.3 | 0.009 | 0.661 | 0.431 | 0.614 |

## 5. Discussion

The experiment performed in this study clarifies the vulnerability of object classification algorithms, specifically those using deep neural networks, to adversarial attacks. The results demonstrate that all algorithms and adversarial attack methods result in a significant decrease in accuracy when the ϵ value exceeds 0.2. This outcome highlights the significance of selecting a suitable ϵ value to develop effective defense strategies against these attacks.

Although the FGSM has the advantages of a higher success rate and faster generation time for perturbed images compared with other methods, the resulting image may contain a large amount of noise. Additionally, our results indicated that AlexNet generates perturbed images significantly faster than the other DNN algorithms, making it an ideal choice when reducing generation time is crucial. This may be because AlexNet has a simpler layer configuration than other DNN algorithms.

A crucial aspect of adversarial attacks is adding perturbations that are not easily detectable by humans. Consequently, the FGSM may not be an effective adversarial attack because it adds high-level noise to the image, thus making it more probable for humans to detect the attack. On the contrary, the PGD method consistently demonstrated a high success rate for attacks across all algorithms. Due to the FGSM adding noise of epsilon only once to the original image, humans can easily detect the noise. However, PGD gradually adds noise several times. Furthermore, perturbations generated with an $\epsilon$ value up to 0.1 were not easily detectable by humans for all DNN algorithms due to the difficulty in distinguishing them from optical noise. Figure 5 depicts the accuracy of each DNN algorithm, and the time required to generate perturbed images using the PGD method with an $\epsilon$ value of 0.1. As the $\epsilon$ value increases, it becomes easier for humans to detect the noise.

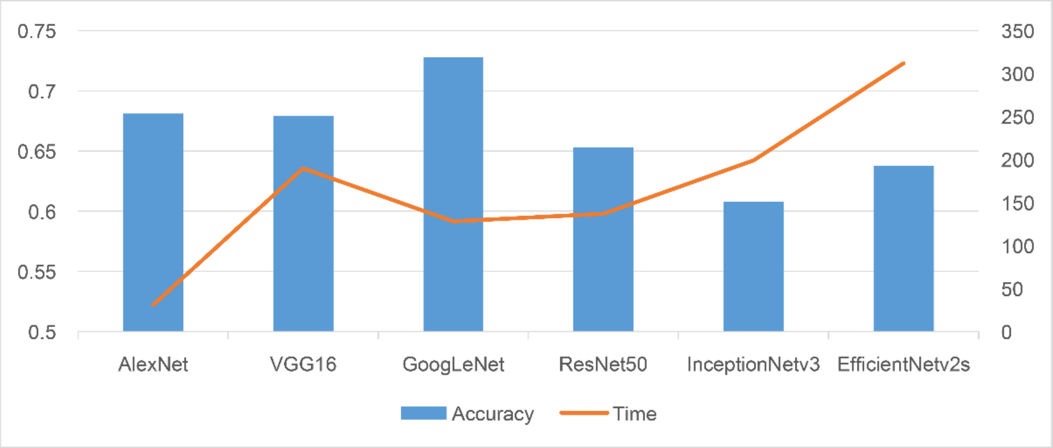

**Figure 5.** Accuracy of each DNN algorithm and time required to generate perturbed images using the PGD method for $\epsilon = 0.1$.

Based on our results, different approaches should be recommended depending on the priority of the defense resource. For instance, if generation time is a critical factor, it is more appropriate to use PGD with AlexNet and an $\epsilon$ value of 0.1. In contrast, if the success rate represents a high priority, PGD with InceptionNetv3 and an $\epsilon$ value of 0.1 is a more suitable strategy. Finally, if a compromise is necessary, the use of PGD with ResNet50 and an $\epsilon$ value of 0.1 is recommended. However, these recommendations should not be generalized and applied indiscriminately. In fact, the performance of object classification algorithms using YOLOv5 with transfer learning may vary for distinct datasets and adversarial attack methods; thus, it is necessary to consider the specific context when designing defense mechanisms against adversarial attacks. The experiment developed in this study provides a substantial foundation for further research and development in this critical area of cybersecurity.

## 6. Conclusions

### *6.1. Contributions*

AI technologies are essential for enabling the operation of MASS. Object detection and classification algorithms are critical for improving navigation and collision avoidance and in optimizing the performance and efficiency of the vessels. However, the vulnerability of AI systems to adversarial attacks is a significant concern, as these attacks can compromise the accuracy and reliability of these systems and have real-world consequences. Despite the lack of experience with adversarial attacks in the maritime industry, the potential risks of such attacks are significant. Therefore, the SMD-Plus dataset, which includes classes such as ferries, boats, and buoys that MASS may encounter during actual operation, was used to generate adversarial images in various ways. We thereafter performed adversarial attacks on YOLOv5 with transfer learning, an object detection and classification algorithm. The experimental results demonstrated that the time required for generating perturbed images varied depending on the DNN algorithm and adversarial attack method. Moreover, we found that changes in the $\epsilon$ value can affect the vulnerability of the system to adversarial attacks. Experimentally, we determined the adversarial attack methods that are most harmful to object detection and classification models.

In AI, the risk of adversarial attacks has long been recognized, and studies on adversarial attacks and mitigation methods have been ongoing. However, the optimization of these studies for the specific characteristics of each industry and the awareness of their necessity is important. Recently, studies on these attacks have also been conducted in the medical field. In the maritime industry, stakeholders are developing and studying systems that apply AI models to realize MASS. However, they have not fully recognized the risks of adversarial attacks or experienced them. Nevertheless, to ensure the safe operation of MASS, recognizing the risks of adversarial attacks and developing measures for their mitigation are crucial. By presenting a case of adversarial attacks using maritime datasets, this study has contributed to raising awareness among stakeholders on cybersecurity in AI. Moreover, our findings will be used to investigate experimental defense technologies to mitigate vulnerability to adversarial attacks, ultimately contributing to the enhancement of cybersecurity in MASS. The development of effective defense strategies could further improve the security and safety of autonomous ships, rendering them a more reliable transportation mode.

### *6.2. Limitations*

Even though this study does not consider technical advancements in adversarial attack methods, its aim is to provide information about security threats to object detection and classification algorithms through adversarial attack methods. Therefore, we simplified our experiments to help gain empathy for the risks of adversarial attacks. For this, we used known adversarial attack algorithms and limited hyperparameters and assumed that the attacks could not be detected when the perturbed images were acquired and validated. Therefore, the results of this experiment cannot be generalized. In the future, we intend to extend our research on the vulnerability of AI systems considering different object detection and classification algorithms in addition to YOLOv5 and various hyperparameters. Furthermore, we will continue to research methods to identify and mitigate vulnerabilities that could pose even more critical threats to the maritime industry.

**Author Contributions:** Conceptualization, C.L. and S.L.; methodology, C.L.; software, C.L.; validation, C.L. and S.L.; formal analysis, C.L.; investigation, C.L. and S.L.; resources, C.L.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, S.L.; visualization, C.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries, Korea (RS-2023-00256086).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The source code presented in this study is openly available in https://github.com/ChanguiLee/YOLOv5toAdversarialAttacks (accessed on 14 April 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Al-Shatti, A.; Khaksar, W. Artificial intelligence in autonomous maritime navigation: A comprehensive review. *J. Navig.* **2021**, *74*, 756–788.
2. Tomic, T.; Peneva, J. Maritime autonomous surface ships: A review of recent developments and challenges. *J. Navig.* **2020**, *73*, 827–843.

3. Apostolidis, K.D.; Papakostas, G.A. A survey on adversarial deep learning robustness in medical image analysis. *Electronics* **2021**, *10*, 2132. [CrossRef]
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
8. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787. [CrossRef]
9. Kim, J.-H.; Kim, N.; Park, Y.W.; Won, C.S. Object detection and classification based on YOLO-V5 with improved maritime dataset. *J. Mar. Sci. Eng.* **2022**, *10*, 377. [CrossRef]
10. Al-Qubaydhi, N.; Abdulrahman, A.; Turki, A.; Abdulrahman, S.; Naif, A.; Bandar, A.; Munif, A.; Abdul, R.; Abdelaziz, A.; Aziz, A. Detection of unauthorized unmanned aerial vehicles using YOLOv5 and transfer learning. *Electronics* **2022**, *11*, 2669. [CrossRef]
11. Maimunah, A.; Rosadi, R. A review of artificial intelligence application in maritime transportation. *J. Mar. Sci. Eng.* **2019**, *7*, 445.
12. *ISO/IEC TR 24028:2020*; Information Technology—Artificial Intelligence—Overview of Trustworthiness in Artificial Intelligence. ISO: London, UK, 2020.
13. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2015**, arXiv:1412.6572.
14. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Commun. Secur.* **2018**, *84*, 317–331. [CrossRef]
15. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing robust adversarial examples. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
16. Kurakin, A.; Goodfellow, I.; Bengioet, S. Adversarial machine learning at scale. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
17. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.
18. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9185–9193. [CrossRef]
19. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2018**, arXiv:1706.06083.
20. Tan, H.; Wang, L.; Zhang, H.; Zhang, J.; Shafiq, M.; Gu, Z. Adversarial attack and defense strategies of speaker recognition systems: A survey. *Electronics* **2022**, *11*, 2183. [CrossRef]
21. Alotaibi, A.; Rassam, M.A. Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet* **2023**, *15*, 62. [CrossRef]
22. Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers, Inc.: Burlington, MA, USA, 2012; pp. 1097–1105.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
28. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.