Journal of
Marine Science and Engineering

MDPI

Article

Vulnerability of Clean-Label Poisoning Attack for Object Detection in Maritime Autonomous Surface Ships

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

[1] Software Testing Center, 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: Artificial intelligence (AI) will play an important role in realizing maritime autonomous surface ships (MASSs). However, as a double-edged sword, this new technology brings forth new threats. The purpose of this study is to raise awareness among stakeholders regarding the potential security threats posed by AI in MASSs. To achieve this, we propose a hypothetical attack scenario in which a clean-label poisoning attack was executed on an object detection model, which resulted in boats being misclassified as ferries, thus preventing the detection of pirates approaching a boat. We used the poison frog algorithm to generate poisoning instances, and trained a YOLOv5 model with both clean and poisoned data. Despite the high accuracy of the model, it misclassified boats as ferries owing to the poisoning of the target instance. Although the experiment was conducted under limited conditions, we confirmed vulnerabilities in the object detection algorithm. This misclassification could lead to inaccurate AI decision making and accidents. The hypothetical scenario proposed in this study emphasizes the vulnerability of object detection models to clean-label poisoning attacks, and the need for mitigation strategies against security threats posed by AI in the maritime industry.

Keywords: object detection; cyberthreat; risk scenario; clean-label poisoning attack; poison frog

check for updates

Citation: Lee, C.; Lee, S. Vulnerability of Clean-Label Poisoning Attack for Object Detection in Maritime Autonomous Surface Ships. *J. Mar. Sci. Eng.* **2023**, *11*, 1179. https://doi.org/10.3390/jmse11061179

Academic Editors: Kevin Jones and Kimberly Tam

Received: 10 May 2023
Revised: 28 May 2023
Accepted: 3 June 2023
Published: 5 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The introduction of maritime autonomous surface ships (MASSs) and artificial intelligence (AI) in the maritime industry has revolutionized the operation of vessels, promoting considerable benefits regarding safety, efficiency, and cost efficiency. Object detection is a critical area where AI is being utilized in MASSs. Such AI-based object detection systems enable vessels to detect and identify other vessels, objects, and potential hazards in their surroundings. By using deep-learning algorithms to identify objects, AI can help MASS in making informed navigational, speed, and course-correction decisions, thereby reducing the risk of accidents and ensuring safe operations [1–6]. Object detection is critical in MASSs because it enables vessels to detect and identify other vessels, objects, and potential hazards in their surroundings, which is essential for safe navigation, collision avoidance, and compliance with regulations [4,5].

However, in the maritime industry, AI systems are exposed to cyberthreats that arise from both unintentionally created and intentionally exploited vulnerabilities, and can have catastrophic consequences to the safety and security of MASSs and their operators [2–5]. An attacker could also manipulate the data used by a vessel's AI algorithms to cause collisions, damage, or other hazards [2–7] that can result in the vessel being redirected or shut down, thereby endangering its safety.

In response to the rapid advancements in AI, a dedicated subcommittee (SC 42) was established under the Joint Technical Committee (JTC 1) of the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). This subcommittee focuses on standardization efforts pertaining to AI technologies. One of the key outputs of ISO/IEC JTC 1/SC 42 was the ISO/IEC TR 24028 technical report, which provides guidance on the trustworthiness of and security threats to AI systems [8]. The report covers numerous topics related to the trustworthiness and security of AI, including data quality, transparency, accountability, and privacy. However, one of its main foci was the security threats posed by AI systems. The report highlighted the need to address security threats throughout the life cycle of AI systems, from their design and development to their operation and maintenance. The report also recognized that security threats in AI could assume several forms, for instance, data poisoning, adversarial attacks, and cyberattacks.

The cybersecurity threats posed by AI endanger the safe navigation of MASSs, but stakeholders in the maritime industry are not fully aware of the severity of the issue. This is because studies on AI cybersecurity threats tend to focus on the technical aspects of attack algorithms, using generic datasets (such as cats and dogs) that hinder industry stakeholders from identifying the problem. Therefore, this study aims to raise awareness among stakeholders by generating and testing a hypothetical attack scenario based on marine datasets, specifically focusing on data-poisoning attacks, which are a cybersecurity threat posed by AI in the maritime industry. In Section 2, we review previous research and explain the contributions of this paper. In Section 3, we examine the clean-label poisoning attack technique used in this study. In Section 4, we propose a hypothetical attack scenario, and verify it through experimentation in Section 5. Section 6 discusses the results, and Section 7 summarizes the research.

2. Background

2.1. Literature Review

Studies on object detection are being conducted as an important challenge in the implementation of MASSs. Rekavandi et al. offered a guide on image- and video-based small-object detection using deep learning with a focus on maritime surveillance [9]. The study provided an overview of deep-learning-based approaches for detecting small objects, evaluated various techniques, and explored challenges and potential solutions. Shao et al. proposed a multiscale object detection model for autonomous ship navigation in a maritime environment [10]. The study focused on the challenges of detecting small objects, such as buoys and boats, and proposed a multiscale detection approach that combined global and local information to improve accuracy. Yao et al. proposed a simultaneous LiDAR-based multiobject tracking and static mapping method for nearshore scenarios [11]. The study focused on the challenges of tracking multiple moving objects in nearshore environments, and proposed a method that combined LiDAR data and static mapping to improve tracking accuracy. Yang et al. proposed a modified version of the YOLOv5 object-detection algorithm known as FC-YOLOv5 for use in unmanned surface vehicles [12]. The proposed FC-YOLOv5 algorithm was tested on datasets, compared with other object-detection algorithms, and demonstrated improved performance concerning detection accuracy and computational efficiency.

Studies are being conducted to identify and assess risks associated with MASSs and to explore cybersecurity. Wróbel et al. investigated the use of leading safety indicators in the maritime industry and their feasibility for MASSs [13]. The study provided an overview of leading safety indicators and their use in the maritime industry, discussed the challenges and opportunities of applying them to MASSs, and proposed a framework for selecting and implementing leading safety indicators for MASSs. Li et al. proposed a network analysis approach for identifying operational risks in MASSs [14]. The study focused on the challenges of identifying and analyzing operational risks in MASSs, and proposed a method that uses network analysis to model the relationships between risk factors and identify critical risk factors. Akpan et al. discussed the cybersecurity challenges facing the maritime sector [15]. The study provided an overview of cyberthreats and vulnerabilities in the maritime industry, explored the potential impact of cyberattacks on maritime operations, and discussed the current state of cybersecurity practices and regulations in the industry. Ben Farah et al. performed a systematic survey on recent advances and future trends in cybersecurity for the maritime industry [16]. The study provided an overview of cybersecurity threats and challenges in the maritime sector, discussed the current state of cybersecurity measures and technologies, and explored future trends and opportunities for improving cybersecurity in the industry.

As the adoption of MASSs continues, technical studies on AI-based object detection and cybersecurity are being actively conducted. However, it is challenging to find specifically related studies to AI security threats in the maritime sector. Fortunately, recently, there have been some studies on adversarial attacks against AI. Walter et al. proposed adversarial AI test cases for evaluating the robustness of maritime autonomous systems (MASs) [17]. The study focused on the challenges of developing effective test cases for evaluating the security and safety of MASs, and proposed a method that uses adversarial AI techniques to generate realistic attack scenarios. The authors evaluated the proposed approach using the case study of a MAS, and demonstrated its effectiveness in identifying vulnerabilities and improving MAS security.

Attacks on AI are divided into black- and white-box attacks depending on whether the attacker has knowledge of the target model. Black-box attacks involve attacking an AI model without knowledge of its internal workings, whereas white-box attacks involve full knowledge of the internal workings of the model. Adversarial and data-poisoning attacks are two of the most common methods used against AI. Adversarial attacks involve manipulating inputs to cause incorrect predictions, whereas data-poisoning attacks involve feeding an AI model with corrupted or biased data during training. Several studies are developing algorithms against these attack methods (adversarial and data-poisoning attacks) depending on whether they are black- or white-box attack methods [18,19].

Various algorithms have been studied for adversarial attacks, such as the fast gradient sign method (FGSM), iterative FGSM method (I-FGSM), momentum iterative FGSM (MI-FGSM), and projected gradient descent (PGD). FGSM is a type of one-step adversarial attack by which an attacker perturbs the input data by adding noise on the basis of the direction of the gradient of the loss function with respect to the input data [20]. The added noise is limited to avoid being noticeable, but is sufficient to cause incorrect predictions. I-FGSM is an iterative version of FGSM in which the attacker repeatedly perturbs the input data in small incremental steps until the desired level of misclassification has been achieved [21]. MI-FGSM further improves on I-FGSM by adding a momentum term to the iterative process, which helps the attacker in escaping from the local minima and finding more effective attack directions [22]. PGD is a stronger and more complex adversarial attack algorithm that iteratively perturbs the input data in the direction of the gradient while constraining the perturbations to be within a certain range [23].

Research on data-poisoning attacks has been underway since the early days of machine-learning studies. Turner, Tsipras, and Madry introduced a novel approach to attacking machine-learning models known as clean-label backdoor attacks [24]. These attacks involve the insertion of trojan examples into the training dataset, carefully labeled to appear benign and avoid suspicion, thus triggering significant misclassifications during testing. The authors underscored the vulnerability of deep-learning models to subtle manipulations and the importance of developing robust defenses against such attacks. Similarly, Saha, Subramanya, and Pirsiavash proposed an innovative attack technique involving the embedding of backdoor triggers within neural network models [25]. By modifying a small fraction of training samples, they ensured that these triggers remained dormant until a specific pattern was present in the input, thereby activating the backdoor behavior. The authors also stressed the challenges associated with detecting and mitigating hidden trigger backdoor attacks. Expanding the scope of clean-label backdoor attacks, Zhao et al. devised a method specifically targeting video recognition models [26]. They developed a strategy to insert hidden triggers into the training data while preserving its performance on clean-label samples. These triggers activate backdoor behavior only when a specific pattern is present in the input video, causing misclassification. The researchers demonstrated the effectiveness of their attack method on various video recognition tasks and evaluated the robustness of different defense mechanisms against such an attack.

Algorithms such as the poison frog, convex polytope, and bullseye polytope have been studied for data-poisoning attacks. The poison frog algorithm injects small amounts of malicious data into the training data to influence the AI model's decision making [27]. The attack is known as "poison frog" because it mimics the behavior of the poison dart frog, which uses a small amount to incapacitate prey or threats by causing paralysis. Convex polytope is a data-poisoning attack that modifies the training data by adding a set of points that lie within a convex polytope, a geometrical shape with straight sides and flat faces [28]. These points are carefully selected to maximize the effect of the attack on the AI model's decision making. The bullseye polytope is a variant of the convex polytope attack [29]. In this attack, malicious points are selected to lie within a smaller subset of the original convex polytope, creating a "bullseye" pattern that increases the effectiveness of the attack.

2.2. Contribution of This Paper

Although technical studies on object recognition and cybersecurity in the maritime sector are active, those on AI security threats are only just beginning. In contrast, the AI science field involves studying algorithms for adversarial and data-poisoning attacks on AI, focusing on improving the attack performance, often using general datasets such as Canadian Institute for Advanced Research 100 classes (CIFAR-100). To raise awareness on AI threats among stakeholders in the maritime industry and promote further studies, it is necessary to conduct experiments using datasets and scenarios to which stakeholders can relate. To achieve this, this study proposes a plausible hypothetical attack scenario using maritime datasets, and verifies the scenario by performing a clean-label poisoning attack on the YOLOv5 object recognition model. The results of these experiments increase stakeholders' awareness of the risks of clean-label poisoning attacks, and highlight the need for studies and defense methods against attacks.

3. Theory: Clean-Label Poisoning Attack

Data-poisoning attacks by which attackers exploit the vulnerabilities of machine-learning models by corrupting their training data have been a well-known threat to machine-learning systems for several years. These examples could be used to manipulate the behavior of a model, leading to incorrect predictions or decisions. However, as machine-learning technology has advanced, traditional data-poisoning attacks are more easily detected. Consequently, attackers have shifted their focus to clean-label poisoning attacks, which can be more difficult to detect [24,25,27,30,31].

Clean-label poisoning attacks are a type of data-poisoning attack in which an adversary manipulates the training data without changing the associated labels. The aim is to introduce subtle, hard-to-detect changes that degrade the performance of the model or cause it to produce specific errors. The general procedure for a clean-label poisoning attack is as follows [24,25,27,30–35]:

1. Data collection: An attacker first gathers information about the target model and its training dataset. This can be achieved using public datasets or datasets with distributions similar to those of the target model.
2. Poison sample selection: An attacker selects a subset of data points to modify or from which to create new instances. The choice of sample depends on the attacker's goal, such as targeting a specific class or introducing a specific type of error.
3. Data manipulation: An attacker subtly manipulates the selected data points to create poisoned instances. These manipulations can include adding, removing, or modifying features to make the instances appear legitimate while still affecting the learning process of the model.
4. Injection of poisoned data: The attacker injects manipulated data points into the target model's training dataset. This can be achieved through various ways, for instance, by compromising the data collection process, infiltrating the data storage system, or

leveraging insider access. An attacker can deploy poisoned data or move to the next step and deploy a poisoned model. As shown in Figure 1, the algorithm generates a poisoned instance by extracting the features of the target cat image and applying them to the base dog image. In the input domain, which is visible to the human eye, the instance appears to be a dog, but in the feature domain, which is perceived by AI, the instance appears to be a cat.

5. Model retraining: The target model is retrained with the poisoned dataset by incorporating poisoned instances into its learning process. This typically results in degraded performance or specific errors depending on the attacker's goals. As shown in Figure 2, the victim collects the poisoned instance, and because there are no apparent anomalies in the image, it is unsuspectingly labeled as a dog and trains the model. In this example, the poisoned instance is exaggerated for the sake of understanding; however, in reality, it is nearly indistinguishable from the base instance.
6. Exploiting the compromised model: Once the model has been retrained, the attacker exploits the compromised model for their purposes. This can involve causing misclassifications, bypassing security measures, or altering the model's behavior in other malicious ways. As shown in Figure 3, when an attacker inputs the target instance into the poisoned model, it is classified as the same class as the poisoned instance because they share similar features. Because the label of the poisoned instance is "dog," the target instance is also classified as "dog," resulting in misclassification.

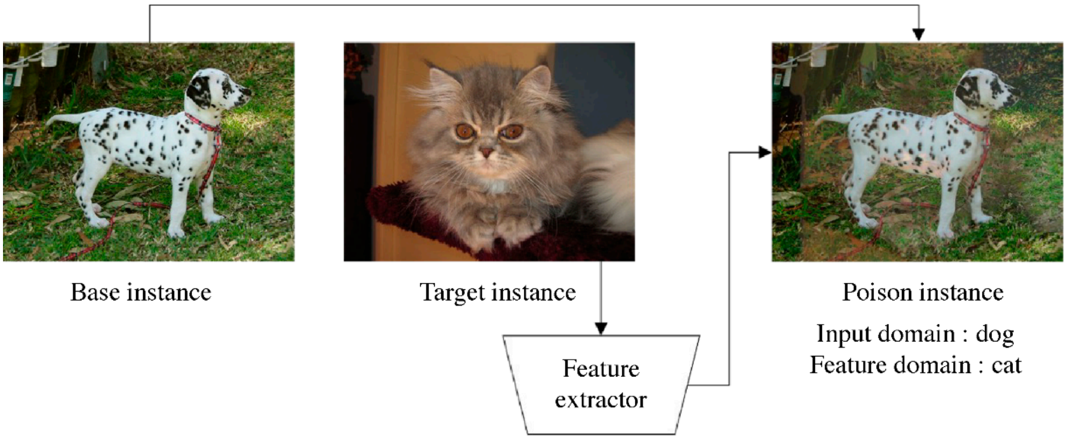

Figure 1. Injecting poisoned data.

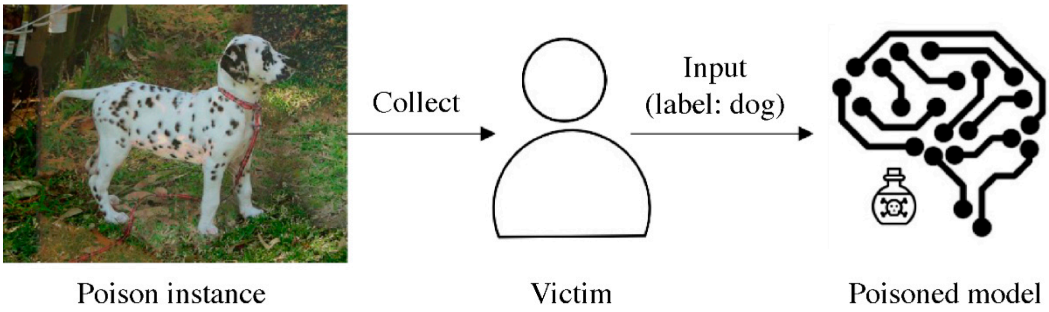

Figure 2. Model retraining.

A clean-label poisoning attack is particularly challenging to defend against because the poisoned datapoints retain their original labels, rendering them more difficult to detect. Although various algorithms have been studied in relation to clean-label poisoning attacks, in this study, we introduce a relatively simple poison frog algorithm, aiming to verify vulnerabilities in object detection of MASSs using this approach.

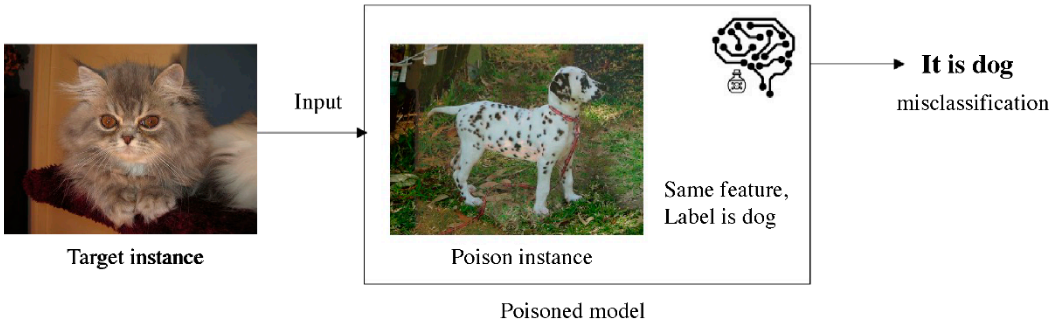

Figure 3. Exploiting a compromised model.

Shafahi et al. proposed a targeted clean-label poisoning attack on neural networks. The poison frog algorithm aims to generate poisoned data points that can cause a model to misclassify a specific target instance while maintaining a subtle and hard-to-detect appearance. The algorithm is based on solving a bilevel optimization problem, and involves iteratively updating the poisoned datapoints using the gradient information from the loss function of the target model. Table 1 summarizes the poison frog algorithm [27].

Table 1. Poison frog algorithm.

Input	Original data points x and their true labels y; target model with parameters θ, loss function $L(\theta, x, y)$
Output	Poisoned datapoints x^*
Algorithm	1. Find poisoned datapoints x^* that minimize the distance to the original datapoints while maximizing the model's loss on the target test point: $min\_x^*\ D(x, x^*)$ subject to $min\_\theta\ L(\theta, x^*, y^*)$ where $D(x, x^*)$ denotes a distance metric between the original datapoints x and the poisoned datapoints x^*. 2. Compute the gradient of the inner optimization problem with respect to the poisoned datapoints using the implicit function theorem: $\nabla\_x^*\ L(\theta^*, x^*, y^*) = -H(\theta^*, x^*, y^*)^{-1} * J(\theta^*, x^*, y^*)$ where θ^* denotes the optimal model parameters, $H(\theta^*, x^*, y^*)$ denotes the Hessian matrix of the loss function, and $J(\theta^*, x^*, y^*)$ denotes the Jacobian matrix of the loss function with respect to the poisoned datapoints. 3. Iteratively update the poisoned datapoints using the computed gradient: $x^*\hat{}(t + 1) = x^*\hat{}(t) - \alpha * \nabla\_x^*\ L(\theta^*, x^*\hat{}(t), y^*)$ where $x^*\hat{}(t)$ denotes the poisoned datapoints at iteration t, α denotes the learning rate, and $\nabla\_x^*\ L(\theta^*, x^*\hat{}(t), y^*)$ denotes the computed gradient. 4. Repeat steps 2 and 3 until convergence or a predefined number of iterations. 5. Inject the generated poisoned datapoints x^* into the training dataset and retrain the target model.

4. Methodology

4.1. Proposed Hypothetical Scenario

In this study, we propose a hypothetical scenario where an object detection model misclassifies a boat as a ferry owing to a data-poisoning attack, and validate it through experimentation. The scenario we devised is as follows:

1. An attacker captures scenes of a ferry approaching the target vessels for piracy purposes.
2. A poisoning image is generated using the clean-label poisoning algorithm with the base image of a boat for each frame of the captured video.
3. The attacker uploads the dataset pretending that it is a new trustworthy dataset for object detection.
4. The collision avoidance system developer (victim) unknowingly trains their model using the poisoned dataset, which appears normal to the human eye and produces high accuracy during training.

Steps 1 to 4 are illustrated in Figure 4.

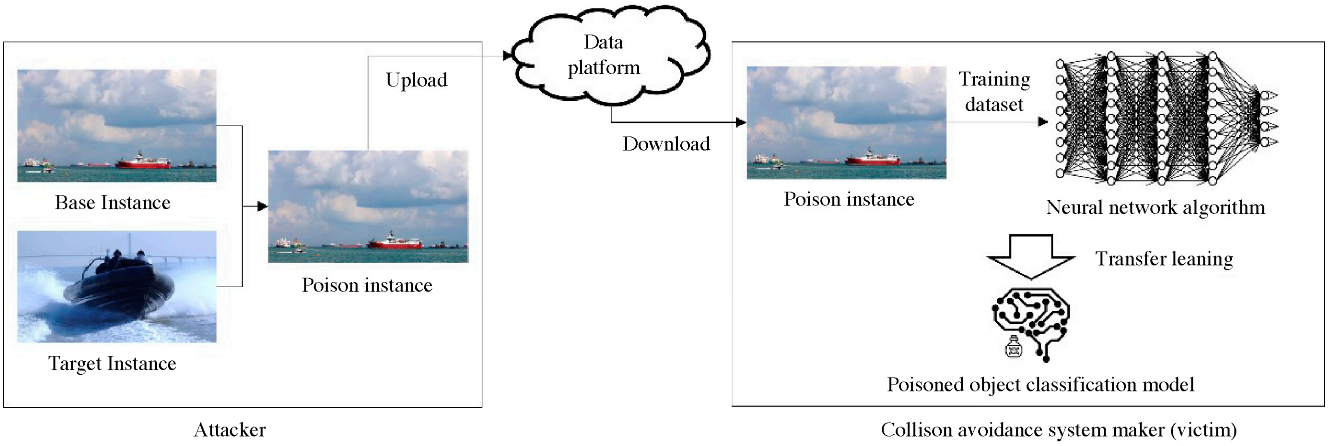

Figure 4. Clean-label poisoning attack scenario by pirates: training phase.

5. The collision avoidance system, which includes a poisoned-object detection model, is installed on a ship and operates normally.
6. The pirate uses a boat to approach the target vessel in a real, similar location to the captured scenes.
7. The poisoned-object detection model is triggered by the poison image and misclassifies the approaching boat as a ferry, thereby failing to detect a boat.
8. Through this clean-label poisoning attack, the collision avoidance system fails to detect the approaching boat, potentially rendering the target vessel vulnerable to piracy.

Steps 5 to 8 are illustrated in Figure 5.

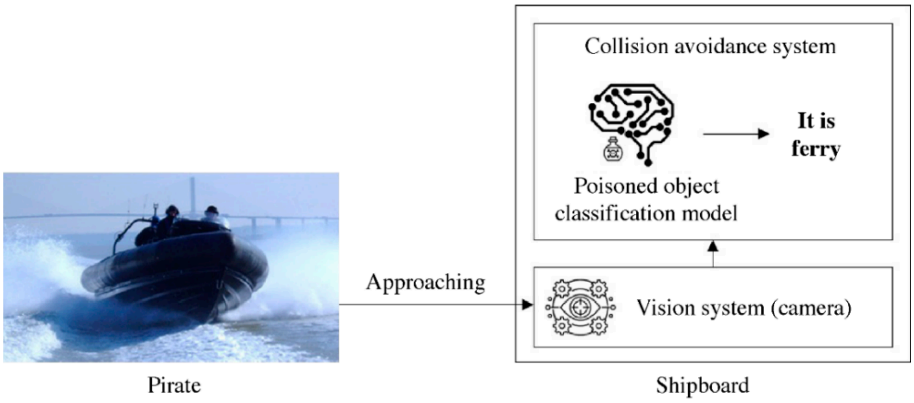
Collision avoidance system
It is ferry
Poisoned object classification model
Approaching
Vision system (camera)
Pirate
Shipboard

Figure 5. Clean-label poisoning attack scenario by pirates: attack phase.

4.2. Validation of the Proposed Scenario

To verify the validity of this scenario as a potential security threat and gain a comprehensive understanding of its occurrence, we devised an experimental procedure that aligned with the generated scenario, and conducted tests accordingly [24–27]. While real-world vision systems utilize videos as inputs for object detection in deep-learning models, our experimental procedure involved extracting video clips frame by frame, treating them as individual still images, and feeding them to the deep-learning model. Although this assumption deviates from real-world conditions, it allowed for us to control variables that may influence the experiment, thereby enabling the effective evaluation of the impact of clean-label poisoning attacks.

The experimental procedure comprised four phases, as shown in Figure 6. In the first phase, we preprocessed the SMD-Plus dataset to meet the requirements of YOLOv5, and split the dataset into a malicious dataset to be used for the attack, training dataset to train the YOLOv5s model, and test dataset to evaluate the model's performance. In the second phase, we assumed the role of an attacker and generated poisonous instances using the poison frog algorithm with the malicious dataset. In the third phase, we assumed the role of the victim, and trained the model with poisoned instances and a clean training dataset. The poisoned instances appeared to be normal at first glance, so they were used for model training without suspicion. During this phase, the object detection model was poisoned due to the poisoned instances. Lastly, in the fourth phase, we evaluated the poisoned model's performance by inputting the test dataset to confirm that it operated normally, and executed the attack by inputting the target instance.

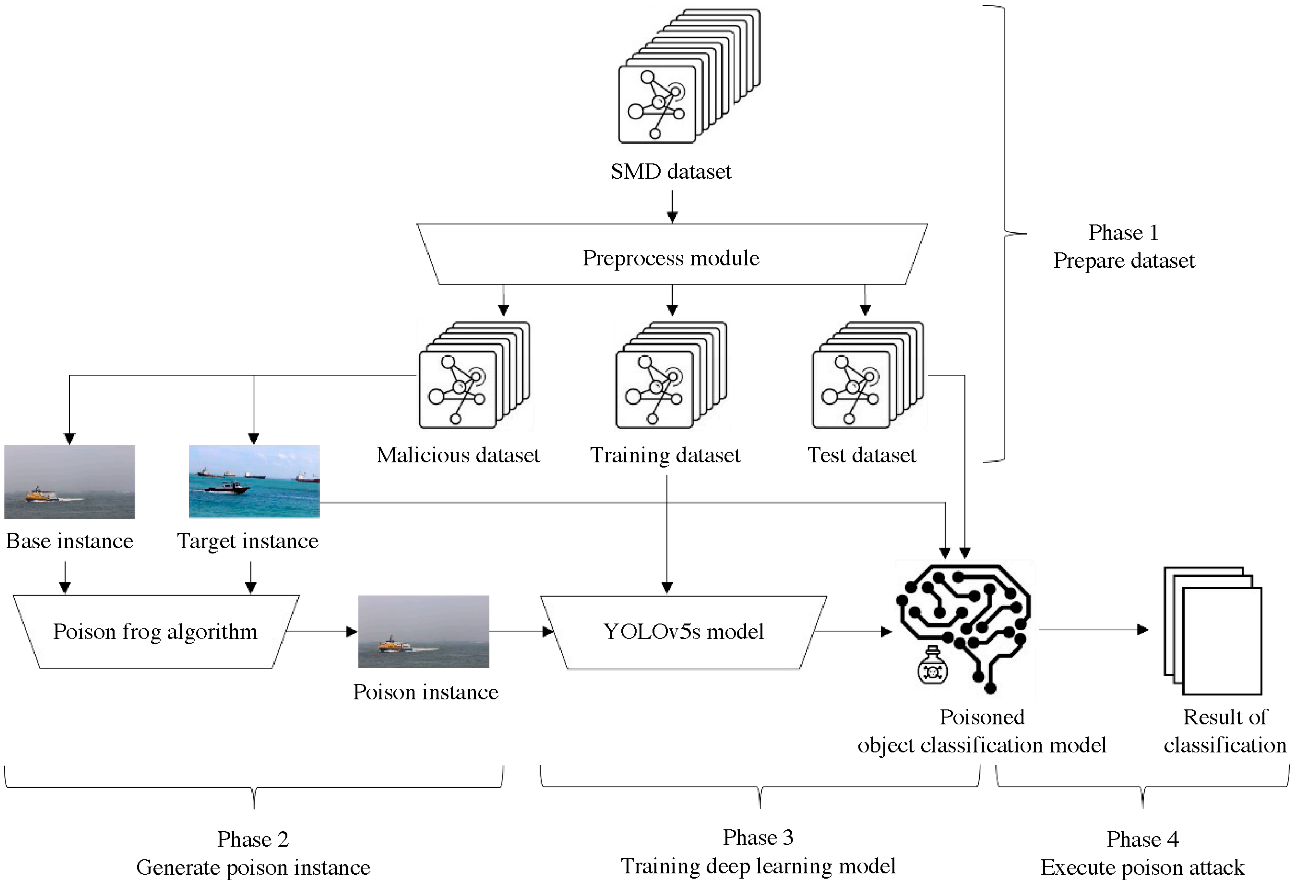

Figure 6. Procedure for validating the proposed hypothetical scenario.

5. Results: Experimental Clean-Label Poisoning Attack

5.1. Dataset Preparation

The acquisition of datasets for the maritime industry can be challenging. The Maritime and Port Authority of Singapore (MPA) and Singapore Management University (SMU) have publicly released the Singapore Maritime Dataset (SMD) for research and development purposes. This dataset includes more than 2 million automatic identification system vessel movements, allowing for researchers to analyze vessel behavior, and develop new algorithms and systems for maritime operations [36]. The SMD is continuously updated, and is available to researchers and developers worldwide to promote innovation in the maritime industry. However, it contains several labeling errors and imprecise bounding boxes, hindering using it as a benchmark dataset for object classification. Consequently, Kim et al. proposed SMD-Plus to improve annotations, particularly the precise bounding box annotations for small maritime objects [37]. SMD-Plus also combines classes with indistinguishable labels to provide additional training data for object recognition. Table 2 lists the classification of training classes in SMD-Plus.

Table 2. Details of the 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

In this experiment, we extracted a single still image per frame from the available videos in the SMD-Plus dataset and modified the annotation format to ensure compatibility with YOLOv5. This process involved converting the original common objects in context format annotations to the YOLOv5 format, which specifies the coordinates and class of each object within the image. We selected two videos from the SMD-Plus dataset to simulate a malicious attack. For the remaining data, we split the dataset into 80% for training and 20% for testing to train and validate the object detection model.

5.2. Poison Instance Generation

In the malicious dataset obtained from SMD-Plus, we prepared base instances using frame-separated images extracted from videos featuring ferries, and target instances using frame-separated images extracted from videos containing boats. Subsequently, we generated poisoned instances using the poison frog algorithm with the ResNet50 neural network [27,28]. To ensure that the changes to the images were not noticeable to humans, we set the number of iterations to 5000, epsilon to 0.02, and alpha to 0.001. The poisoned instances generated for each frame are shown in Figure 7. Despite natural variations in the objects' positions between frames, no anomalies were discernible to the human eye. Consequently, the victim was likely to label objects as ferries without becoming suspicious.

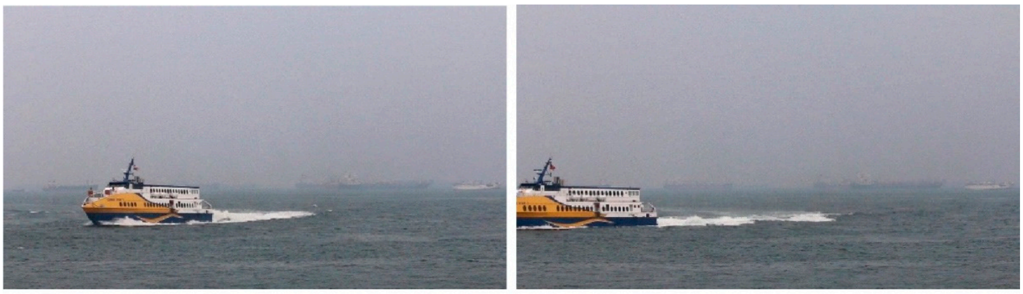

Figure 7. Poisoned instances generated for each frame.

5.3. Deep-Learning Model and Attack Execution

We combined the training dataset generated by splitting the SMD-Plus dataset and the poisoned instances to create a training dataset. We then selected a pretrained model to perform transfer learning using the YOLOv5 algorithm. YOLOv5 offers four models based on speed and accuracy, that is, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. For this experiment, we selected the YOLOv5s model as the base model, the smallest model that was expected to be more vulnerable to data-poisoning attacks, and performed transfer learning by fine-tuning it with a downstream task to improve the training speed and accuracy. The training parameters were set to 100 epochs, and the batch size was 16 [15,17]. Transfer learning is a common training method used to improve the performance of models with relatively small datasets wherein a pretrained model that trained on a large volume of data is used as the base model. However, models generated through transfer learning are more vulnerable to data-poisoning attacks [27].

After completing the training, we verified the accuracy of the object detection with the trained model using the test dataset split from the SMD-Plus dataset. Figure 8 shows the resulting images of object detection for ferries. The left-hand side image in Figure 8 was classified as a ferry with a confidence of 0.82, while the right-hand side image was classified as a boat with a high confidence of 0.91.

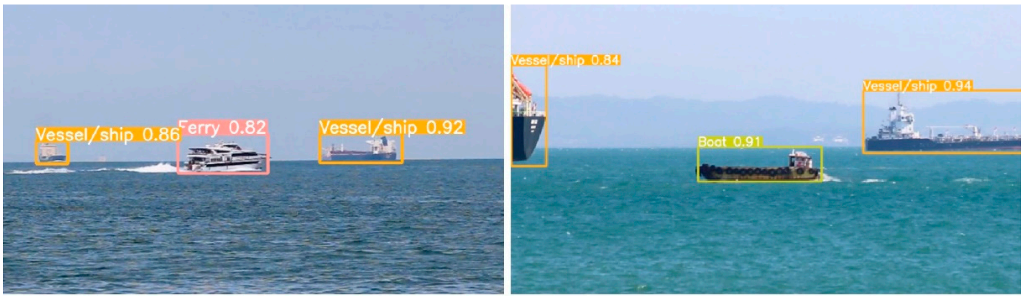

Figure 8. Object detection results for the test dataset.

Table 3 lists the mean average precision (mAP) for each class with a confidence threshold of 0.5. The model exhibited high mAP values for all classes, including boats and ferries. Furthermore, even when considering the mAP of all classes simultaneously, the model achieved a high level of object detection performance with a score of 0.858. This implies that, despite being trained with poisoned instances, the model performed exceptionally well, rendering it challenging for the victim to detect any abnormal behavior in the model.

Table 3. Results of object detection for the test dataset.

Class	Precision	Recall	mAP@0.5
All	0.894	0.797	0.858
Boat	0.992	0.886	0.937
Vessel/ship	0.894	0.941	0.960
Ferry	0.833	0.862	0.855
Kayak	0.737	0.467	0.590
Buoy	1.000	0.789	0.895
Sailboat	0.902	1.000	0.995
Others	0.901	0.630	0.772

Next, a poisoning attack was executed using a target instance, causing a boat to be misclassified as a ferry with high confidences of 0.87 and 0.89, as shown in Figure 9. The model misclassified the object owing to the features of the target instance hidden in the poisoned instances used to train the model.

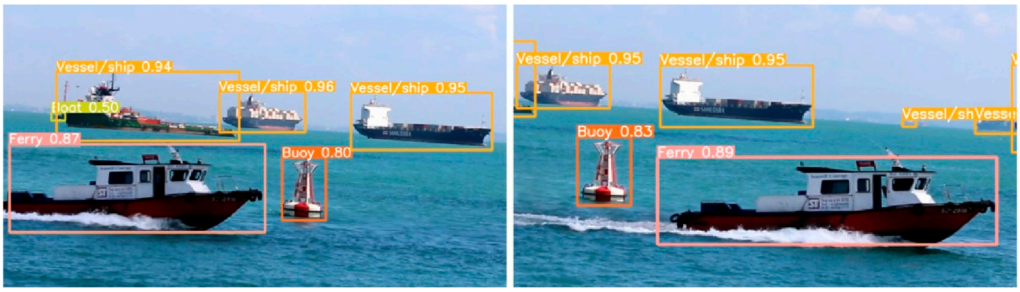

Figure 9. Object detection results for the target instance.

5.4. Result Analysis

To conduct a more detailed examination of the misclassification of boats resulting from the clean-label poisoning attacks, we compared the data extracted from two video clips, as summarized in Table 4 [24–27]. The right-hand side image in Figure 8 was obtained from the MVI_1469_VIS.avi file in the SMD-Plus dataset, with a total of 528 images used for testing. Furthermore, the images in Figure 9 were extracted from the MVI_1470_VIS.avi file in the same dataset, with 168 images used for the attack. In general, errors in object detection can be categorized as either failure to detect or misclassification. Among the images extracted from the MVI_1469_VIS.avi file, which constituted the test set, only two instances of undetected objects were observed, with no instances of misclassification recorded. These missed detections could be attributed to the performance limitations of the model, indicating a need for improvement, but without raising significant concerns. On the other hand, all 168 images extracted from the MVI_1470_VIS.avi file that were used for the attack exhibited misclassification. The occurrence of 168 misclassifications throughout the entire video clip of about 7 s triggered under specific conditions by the poisoned images can be considered a critical error.

Table 4. Comparison of object detection results on two image sets.

Source	Input Images	Not Detected	Misclassified
MVI_1469_VIS.avi	528	2	0
MVI_1470_VIS.avi	168	0	168

This experiment allowed for us to confirm two key findings. First, the model exhibited high-confidence misclassifications, indicating that it confidently identified the detected object as a ferry with probabilities of 87% and 89%; however, adjusting the confidence threshold did not resolve the misclassifications. Second, the model generally performed well under normal circumstances when a test dataset was used, but triggered misclassifications under specific situations, such as when a target instance was input. Thus, it is challenging to determine whether the model has been subjected to poisoning.

6. Discussion

The vulnerability of computer vision using AI algorithms, such as YOLOv5, has garnered significant attention and highlighted the importance of studying it among researchers in the field of AI. However, stakeholders in the maritime industry, while focused on improving object detection for MASSs under challenging conditions (such as small size, partial visibility, or adverse weather), have shown less concern regarding the vulnerabilities of AI. This study aims to raise awareness among stakeholders of the security threats posed by AI in the maritime industry. Therefore, the objective of this study was to propose and validate a risk scenario wherein AI misclassifies boats as ferries, hindering the detection of approaching boats and potentially leading to piracy incidents. To validate this hypothesis, a four-phase experiment was conducted. In the first phase, the SMD-Plus dataset was preprocessed to meet the YOLOv5's requirements, and the dataset was split for experimentation. In the second phase, poisoned instances were generated using the ResNet50 neural network and the poison frog algorithm. In the third phase, a poisoned model was created through transfer learning using poisoned instances and a training dataset with a pretrained YOLOv5s model. In the fourth phase, the accuracy of the poisoned model was evaluated using the test dataset, and the attack was executed using the target instance.

Despite demonstrating high accuracy in object detection with a mAP of 0.858, the poisoned model misclassified the boats as ferries when tested with the target instances due to the poisoning attack. This indicated that the model was affected by the hidden feature information embedded in the target instances, triggering misclassifications upon their input. Particularly noteworthy is that the target instances appeared to be normal to human eyes, hindering detecting the presence of hidden feature information. Therefore, research is needed to detect and remove hidden feature information in seemingly normal images or to modify the model's learning parameters to mitigate vulnerabilities. A robust model demonstrates high performance without triggering misclassification in such clean-label poisoning attacks.

Because this experiment generated a hypothetical scenario using a limited dataset and attack method, it is difficult to generalize the vulnerability of AI systems in actual operational environments. However, the significance of this study lies in raising awareness and fostering appreciation among stakeholders regarding potential threats related to AI when employing datasets acquired from the maritime domain, instead of commonly used datasets such as those involving dogs and cats.

7. Conclusions

In this study, a hypothetical attack scenario targeting an object detection system for MASS was proposed and tested using the SMD-Plus dataset and poison frog algorithm to attack the YOLOv5 model, causing the misclassification of boats as ferries. Despite the high accuracy exhibited by the poisoned YOLOv5 model, its vulnerability was demonstrated by causing misclassification through a data-poisoning attack. This study is significant in raising awareness among stakeholders in the maritime industry about the importance of being cautious of data-poisoning attacks and understanding the security threats posed by AI. In addition, it emphasizes the need for developing strategies and emergency plans to mitigate security threats posed by AI.

During the experiment, we encountered difficulties in obtaining a dataset, which led us to using the SMD-Plus dataset. Although the SMD-Plus dataset is useful, it does not represent all maritime conditions. In addition, we only used it to help stakeholders in understanding the concept of clean-label data-poisoning attacks. Therefore, in future research, we will create a dataset that includes various maritime conditions by incorporating other datasets such as Seaships and MARVEL and perform clean-label data-poisoning attacks on the developed dataset. Furthermore, to ensure the efficiency of this experiment, we employed still images extracted frame by frame. However, in future studies, we intend to conduct experiments using video inputs to examine the impact of a more realistic approach and evaluate the real-time processing capability of YOLOv5.

Lastly, the poison frog algorithm utilized in this study represents a relatively simple approach among various types of clean-label poisoning attack algorithms. To comprehensively evaluate the vulnerability of object detection, we plan to apply different algorithms and suitable methods for the maritime domain to mitigate the discussed vulnerabilities. These methods include techniques for detecting or removing inappropriate information embedded in the images, and model training approaches that enhance the robustness of object detection.

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 the Korea Institute of Marine Science and Technology Promotion (KIMST), funded by the Ministry of Oceans and Fisheries, Korea (20210635).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

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

References

1. Akdağ, M.; Solnør, P.; Johansen, T.A. Collaborative collision avoidance for maritime autonomous surface ships: A review. *Ocean Eng.* **2022**, *250*, 110920. [CrossRef]
2. Xu, H.; Moreira, L.; Guedes Soares, C.G. Maritime autonomous vessels. *J. Mar. Sci. Eng.* **2023**, *11*, 168. [CrossRef]
3. Liu, C.; Chu, X.; Wu, W.; Li, S.; He, Z.; Zheng, M.; Zhou, H.; Li, Z. Human–machine cooperation research for navigation of maritime autonomous surface ships: A review and consideration. *Ocean Eng.* **2022**, *246*, 110555. [CrossRef]

4. Qiao, Y.; Yin, J.; Wang, W.; Duarte, F.; Yang, J.; Ratti, C. Survey of deep learning for autonomous surface vehicles in marine environments. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3678–3701. [CrossRef]
5. Wang, L.; Wu, Q.; Liu, J.; Li, S.; Negenborn, R. State-of-the-art research on motion control of maritime autonomous surface ships. *J. Mar. Sci. Eng.* **2019**, *7*, 438. [CrossRef]
6. Jorge, V.A.M.; Granada, R.; Maidana, R.G.; Jurak, D.A.; Heck, G.; Negreiros, A.P.F.; Dos Santos, D.H.; Gonçalves, L.M.G.; Amory, A.M. A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions. *Sensors* **2019**, *19*, 702. [CrossRef]
7. Cho, S.; Orye, E.; Visky, G.; Prates, V. *Cybersecurity Considerations in Autonomous Ships*; NATO Cooperative Cyber Defence Centre of Excellence: Tallinn, Estonia, 2022.
8. *ISO/IEC. TR 24028*; Information Technology—Artificial Intelligence—Overview of Trustworthiness in Artificial Intelligence. ISO: Geneva, Switzerland, 2020.
9. Rekavandi, A.M.; Xu, L.; Boussaid, F.; Seghouane, A.-K.; Hoefs, S.; Bennamoun, M. A Guide to Image and Video based Small Object Detection using Deep Learning: Case Study of Maritime Surveillance. *arXiv* **2022**, arXiv:2207.12926.
10. Shao, Z.; Lyu, H.; Yin, Y.; Cheng, T.; Gao, X.; Zhang, W.; Jing, Q.; Zhao, Y.; Zhang, L. Multi-scale object detection model for autonomous ship navigation in maritime environment. *J. Mar. Sci. Eng.* **2022**, *10*, 1783. [CrossRef]
11. Yao, Z.; Chen, X.; Xu, N.; Gao, N.; Ge, M. LiDAR-based simultaneous multi-object tracking and static mapping in nearshore scenario. *Ocean Eng.* **2023**, *272*, 113939. [CrossRef]
12. Yang, H.; Xiao, J.; Xiong, J.; Liu, J. Rethinking YOLOv5 with feature correlations for unmanned surface vehicles. In Proceedings of the 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022); Springer Nature: Singapore, 2023; pp. 753–762. [CrossRef]
13. Wróbel, K.; Gil, M.; Krata, P.; Olszewski, K.; Montewka, J. On the use of leading safety indicators in maritime and their feasibility for Maritime Autonomous Surface Ships. *Proc. Inst. Mech. Eng. Part O* **2023**, *237*, 314–331. [CrossRef]
14. Li, X.; Oh, P.; Zhou, Y.; Yuen, K.F. Operational risk identification of maritime surface autonomous ship: A network analysis approach. *Transp. Policy* **2023**, *130*, 1–14. [CrossRef]
15. Akpan, F.; Bendiab, G.; Shiaeles, S.; Karamperidis, S.; Michaloliakos, M. Cybersecurity challenges in the maritime sector. *Network* **2022**, *2*, 123–138. [CrossRef]
16. Ben Farah, M.A.; Ukwandu, E.; Hindy, H.; Brosset, D.; Bures, M.; Andonovic, I.; Bellekens, X. Cyber security in the maritime industry: A systematic survey of recent advances and future trends. *Information* **2022**, *13*, 22. [CrossRef]
17. Walter, M.J.; Barrett, A.; Walker, D.J.; Tam, K. Adversarial AI testcases for maritime autonomous systems. *AI Comput. Sci. Robot. Technol.* **2023**, *2*, 1–29. [CrossRef]
18. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.* **2018**, *84*, 317–331. [CrossRef]
19. Steinhardt, J.; Koh, P.W.; Liang, P.S. Certified defenses for data poisoning attacks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3517–3529.
20. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572. Available online: https://arxiv.org/abs/1412.6572 (accessed on 28 May 2023).
21. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Examples in the Physical World. *arXiv* **2016**, arXiv:1607.02533.
22. 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]
23. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2018**, arXiv:1706.06083.
24. Turner, A.; Tsipras, D.; Madry, A. Clean-label backdoor attacks. In Proceedings of the ICLR 2019 Conference, New Orleans, LA, USA, 6–9 May 2019.
25. Saha, A.; Subramanya, A.; Pirsiavash, H. Hidden trigger backdoor attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11957–11965. [CrossRef]
26. Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; Jiang, Y.-G. Clean-label backdoor attacks on video recognition models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14431–14440. [CrossRef]
27. Shafahi, A.; Huang, W.R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
28. Zhu, C.; Huang, W.R.; Li, H.; Taylor, G.; Studer, C.; Goldstein, T. Transferable clean-label poisoning attacks on deep neural nets. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 10–15 June 2019.
29. Aghakhani, H.; Meng, D.; Wang, Y.-X.; Kruegel, C.; Vigna, G. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; Volume 2021. [CrossRef]
30. Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against support vector machines. In Proceedings of the 29th International Conference on Machine Learning (ICML-12), Edinburgh, UK, 26 June–1 July 2012; pp. 1467–1474.
31. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.P.; Tygar, J.D. Adversarial machine learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Chicago, IL, USA, 21 October 2011; pp. 43–58. [CrossRef]

32. Steinhardt, J.; Koh, P.W.; Liang, P. Certified defenses against adversarial examples. In Proceedings of the 2017 Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; pp. 281–292.
33. Yerlikaya, F.A.; Bahtiyar, Ş. Data poisoning attacks against machine learning algorithms. *Expert Syst. Appl.* **2022**, *208*, 118101. [CrossRef]
34. Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. In Proceedings of the 6th International Conference on Learning Representations (ICLR'18), Vancouver, BC, Canada, 30 April–3 May 2018.
35. Xiao, H.; Biggio, B.; Brown, G.; Fumera, G.; Eckert, C.; Roli, F. Is feature selection secure against training data poisoning? In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1224–1235.
36. Prasad, D.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1993–2016. [CrossRef]
37. 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.* **2023**, *10*, 377. [CrossRef]

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.