electronics

Article

Performance Comparison of Adversarial Example Attacks Against CNN-Based Image Steganalysis Models

Hyeonseong Kim, Hweerang Park and Youngho Cho *

Department of Cyber Security and Computer Engineering, Graduate School of Defense Management, Korean National Defense University, Nonsan 33021, Republic of Korea; hshc0314@gmail.com (H.K.); sharku7@gmail.com (H.P.)
* Correspondence: younghocho@korea.kr

Abstract

A steganography technique hides a secret message stealthily within multimedia files such as images, videos, or even the skin image of an avatar in a metaverse environment. Conversely, a steganalysis technique detects steganographic files containing hidden messages. Recently, with the rapid advancement of Convolutional Neural Network (CNN) architectures, CNN-based image steganalysis models have been proposed to accurately detect steganography in image files. Meanwhile, Deep Learning (DL) models, including CNNs, are known to be vulnerable to evasion attacks such as adversarial example attacks, which can cause a CNN-based classifier to misclassify an input image according to the attacker's intent. Given the lack of prior research in this domain, this paper investigates how effectively state-of-the-art adversarial example attack methods can evade three representative CNN-based image steganalysis ML models (XuNet, YeNet, and SRNet). Specifically, we first describe a system model consisting of three participating entities—a naïve attacker, a defender (Defender Lv. 1 and Defender Lv. 2), and an adversarial attacker. Next, we present experimental results comparing nine adversarial example attack methods against the three representative CNN models in terms of various metrics, including classification accuracy (CA), missed detection rate (MDR), attack success index (ASI), and adversarial example generation time (AEGT).

Keywords: image steganography; image steganalysis; convolutional neural network; adversarial example attacks

check for updates

Academic Editor: Miin-shen Yang

Received: 23 August 2025
Revised: 9 November 2025
Accepted: 11 November 2025
Published: 13 November 2025

Citation: Kim, H.; Park, H.; Cho, Y. Performance Comparison of Adversarial Example Attacks Against CNN-Based Image Steganalysis Models. *Electronics* **2025**, *14*, 4422. https://doi.org/10.3390/electronics14224422

Copyright: © 2025 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

Steganography techniques have long been employed for covert communication by embedding secret messages within various forms of communication media [1]. With the proliferation of digital media—such as images, videos, and document files—advanced and novel steganographic methods have been actively developed [1–5]. Conversely, steganalysis refers to the set of techniques used to detect steganographic content and, in some cases, extract the hidden messages [6]. Thanks to the rapid advancement of deep learning (DL), convolutional neural network (CNN)-based steganalysis models have been introduced to enhance detection performance, achieving higher classification accuracy and lower missed detection rates [2–4,6–8].

However, like many deep learning (DL) models, CNN-based image steganalysis machine learning (ML) models are also vulnerable to security threats posed by adversarial machine learning (AML), which can evade detection or significantly reduce classification reliability [6–11]. In particular, adversarial example attacks constitute a form of AML wherein

an attacker can easily generate an adversarial example—i.e., an adversarial stego image—by adding small perturbations to an image, thereby causing the steganalysis ML model to misclassify it [9–16]. For instance, Zhang et al. [9] were the first to generate adversarial stego images targeting neural network-based steganalysis systems. Tang et al. [10] subsequently proposed a novel steganographic scheme employing adversarial embedding to deceive CNN-based steganalysis models. Similarly, Liu et al. [11] introduced an improved adversarial embedding method that enhances the security performance of steganography compared to conventional techniques. Furthermore, several other approaches have also been proposed [12].

Meanwhile, according to our extensive survey, there are no related studies examining how existing adversarial example methods can pose a serious threat to CNN-based image steganalysis models. Motivated by this, this paper presents an extensive empirical study that compares and analyzes the attack performance of adversarial example methods against CNN-based image steganalysis models.

The main contributions of this study are as follows.

- To the best of our knowledge, this is the first study to extensively compare the performance of state-of-the-art adversarial example attack methods against CNN-based image steganalysis machine learning (ML) models (hereafter referred to as CNN models). To this end, we implemented three representative CNN models—namely, XuNet [13], YeNet [14], and SRNet [15]—using two well-known datasets, Bossbase [17] and BOWS2 [18], which are among the most widely used and well-recognized benchmark datasets in the field of image steganography and steganalysis [19–21]. Subsequently, we compared nine representative adversarial example attack methods introduced in Section 2.2 using the IBM Adversarial Robustness Toolbox v.1.17. (ART) [22].
- To better understand the security problem between an attacker and a defender in the field of steganalysis research, we formally describe a system model consisting of three participating entities: a naïve attacker (a basic steganography system), a defender (two types of CNN-based image steganalysis systems, namely Defender Lv. 1 and Defender Lv. 2), and an adversarial attacker (an adversarial example attack system). In particular, while Defender Lv. 1 represents a basic CNN-based image steganalysis system, we newly introduce an advanced defender type—Defender Lv. 2—equipped with a human visual inspection capability to address a more challenging security problem.
- According to our experimental results, we found that existing performance metrics—such as classification accuracy (CA) and missed detection rate (MDR)—do not adequately reflect how effectively adversarial example attackers can evade defenders. This is because several adversarial attack methods significantly degrade the visual quality of stego images, making them appear suspicious and easily detectable by Defender Lv. 2.
- To address the limitations of using existing metrics in evaluating attack performance against Defender Lv. 2, we propose a new metric called the Attack Success Index (ASI). The ASI measures the degree to which adversarial examples are successfully generated and delivered to the receiver while evading an advanced defense system that combines CNN-based steganalysis with human-like visual inspection (Defender Lv. 2), as described in Section 2.

The remainder of this article is organized as follows. In Section 2, we briefly describe the system model comprising the three participating entities considered in this study and summarize nine adversarial attack methods. In Section 3, we present a comparative performance evaluation of representative adversarial attack methods against three CNN-

based steganalysis models. Finally, Section 4 concludes the paper and outlines directions for future research.

2. Background and Related Works

2.1. System Model

We briefly describe the system model considered in our study (see Figure 1) as follows. In the system model, there are three participating entities: (1) a naïve attacker (an image steganography system), (2) a defender—Defender Lv. 1 (an ML-based steganalysis system) and Defender Lv. 2 (an ML-based steganalysis system equipped with human-like visual inspection)—and (3) an adversarial attacker (an adversarial example attack system). In this study, the attackers' ultimate goal is to generate a stego image that embeds a secret message and to evade detection by the defending systems. These defending systems are ML-based image steganalysis models that classify a given image as either a plain image (i.e., non-stego) or a stego image, based on their pre-trained ML models. We further explain our research problem by detailing each participant's capabilities and limitations below.

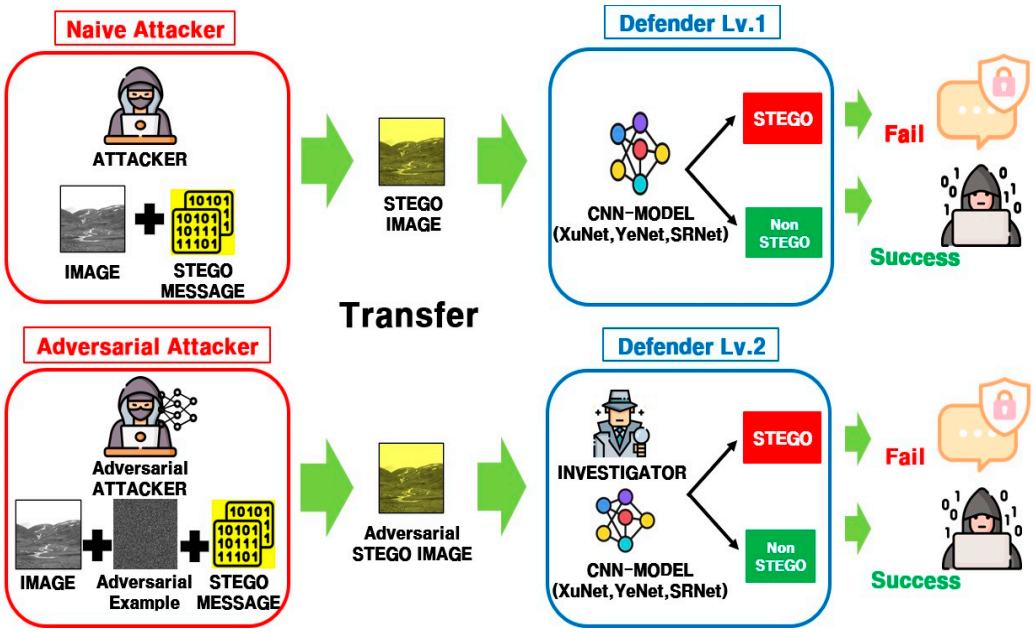


Figure 1. Attack and Defense model.

2.1.1. Naïve Attacker: Image Steganography System

The naïve attacker employs a conventional image steganography system that generates a stego image by embedding a secret message within a cover image. After producing the stego image, the naïve attacker submits it to the defender (an ML-based image steganalysis system). If the defender misclassifies it as a plain image (i.e., a non-stego image), the attack is considered successful. Various algorithms have been developed for embedding secret messages into images, such as HUGO [23] and WOW (Wavelet Obtained Weights) [24]. HUGO was the first steganographic algorithm that dynamically determines embedding locations based on the characteristics of the cover image, while WOW is a robust steganographic method that modifies the texture regions of a cover image without altering its edge regions. In this study, we assume that such naïve attacks can be detected by the defender (an ML-based image steganalysis system).

2.1.2. Defender: ML-Based Image Steganalysis System

Instead of relying on traditional image steganalysis methods [3,4], we adopt state-of-the-art machine learning (ML) techniques to construct the defender, hereafter referred to as Defender Lv. 1. To implement Defender Lv. 1, we employ three representative CNN-based image steganalysis models: XuNet [13], YeNet [14], and SRNet [15]. Specifically, XuNet uses tanh and ReLU activation functions and demonstrates higher accuracy than conventional SRM (Spatial Rich Model)-based steganalysis. YeNet introduces a novel activation function called TLU (Truncated Linear Unit) and reportedly achieves better accuracy than XuNet. Finally, SRNet is a CNN-based model that won the ALASKA steganalysis competition [19]. To further enhance detection capability, we consider an advanced defender—Defender Lv. 2—that augments the ML-based detector with a human visual inspection module. This module is intended to capture low-quality adversarial stego images that result from heavy perturbations injected by an attacker (e.g., to embed a large payload or to produce an adversarial stego image). In this study, we implement the visual-inspection module using peak signal-to-noise ratio (PSNR) computed between a cover image and its corresponding adversarial stego image; when the PSNR is below 30 dB, the difference becomes perceptible to the human visual system [25], and the adversarial stego image is considered detectable by Defender Lv. 2.

2.1.3. Adversarial Attacker: Adversarial Example Attack System

In this system model, the adversarial attacker employs various adversarial example attack methods to evade the ML-based defense systems (Defender Lv. 1 and Defender Lv. 2) described above. The objective of the adversarial attacker is to generate an adversarial stego image that is misclassified as a plain image (i.e., non-stego image) by the ML-based steganalysis models. For this purpose, we use nine state-of-the-art adversarial attack methods provided by the IBM Adversarial Robustness Toolbox (ART) v1.0.1 [22]: FGSM [26], BIM [27], PGD [28], C&W [29], EAD [30], Deepfool [31], JSMA [32], NewtonFool [33] and Wasserstein [34].

2.2. Adversarial Example Attacker

In this section, we provide a brief overview of various adversarial attack techniques in the context of image classification. An adversarial example attack refers to a method that generates an image by adding subtle perturbations to an original image, thereby inducing misclassification in the ML model [6]. The perturbed image is then fed into the trained ML model to produce an output according to the attacker's intent. Such adversarial attacks are particularly dangerous because they can achieve the desired misclassification without directly modifying the model itself, and they pose a serious threat as they can be easily generated by anyone [7]. Since the introduction of the Fast Gradient Sign Method (FGSM) [26] in 2014, adversarial attack research has been actively conducted across diverse domains, including medical imaging and autonomous navigation security. In this study, we compare and analyze nine adversarial example attack methods—ranging from FGSM to the recently proposed Wasserstein attack [34]—through a comprehensive evaluation. Before presenting the comparative analysis, we briefly introduce each method below.

2.2.1. Fast Gradient Sign Method (FGSM) [26]

FGSM [26], proposed by I. Goodfellow et al. in 2014, is a simple, computationally efficient method for generating adversarial examples. FGSM constructs an adversarial perturbation for a given input by using the sign of the gradient of the loss with respect to the input; the resulting perturbation is measured using the L_∞ norm and the method is designed to be fast to compute. This method assumes that the ML-based model encounters

a problem when the training phase is compromised by adding perturbations in a direction opposite to the gradient of the loss function.

2.2.2. Basic Iterative Method (BIM) [27]

BIM (Basic Iterative Method) [27] is an extension of FGSM [26]. The key idea of BIM is to apply the FGSM algorithm iteratively with small step sizes, allowing the perturbation to gradually approach the decision boundary of the target model. At each iteration, the adversarial example is clipped to ensure that the cumulative perturbation remains within a predefined L_∞ bound.

2.2.3. Projected Gradient Descent (PGD) [28]

Madry et al. [28] proposed the Projected Gradient Descent (PGD) method, which can be regarded as an extended and generalized form of FGSM and BIM. PGD generates adversarial examples by iteratively maximizing the loss of the target classification model with respect to the input, while constraining the perturbation within a predefined L_p norm bound. In each iteration, the input is updated by taking a small step in the direction of the gradient of the loss function (scaled by a fixed step size or learning rate), followed by a projection step that ensures the perturbed image remains within the allowed perturbation region. PGD is known for its high attack success rate and strong transferability, and it has been widely used in both adversarial attack and defense research. As a powerful first-order attack method, PGD has been demonstrated to efficiently and effectively fool deep neural network models.

2.2.4. Carlini and Wagner (C&W) [29]

Carlini and Wagner [29] introduced a set of three attacks, known as C&W attacks, in 2016. They proposed an attack method by constraining distance metrics L_0, L_2, and L_∞. The basic idea is to build powerful attacks that generate adversarial perturbations by solving a norm-restricted constrained optimization problem. Their experiments show that defensive distillation cannot defend against C&W attacks. This method achieves a 100% success rate in attacking the trained distillation network model. In this paper, the L_2-based version used for performance verification and comparison is applied in many studies.

2.2.5. ElasticNet (EAD) [30]

The EAD method [30] is an extension of the C&W attacks. The difference is that it additionally controls the L_1 norm of the adversarial perturbations, which is not considered in the original C&W method. Like C&W, the EAD attack can successfully bypass defensive distillation.

2.2.6. DeepFool [31]

DeepFool [31], proposed by Moosavi-Dezfooli et al., is a geometric method for generating perturbations and is based on the L_2 distance metric. DeepFool assumes local linearity of deep learning models and iteratively linearizes the classifier to estimate the minimal perturbation that moves an input across the decision boundary. This method computes perturbations by successive pixel-wise adjustments that push a clean example toward and eventually across the decision boundary, yielding a valid adversarial example. DeepFool is known for obtaining (an estimate of) the minimum perturbation required for a successful adversarial attack and is therefore widely used in research and experimentation.

2.2.7. Jacobian Saliency Map (JSMA) [32]

Papernot et al. [32] proposed JSMA, which is first based on the L_0 distance metric. The basic idea is to modify a sufficient number of pixels to attack the model, rather than

the entire image's pixels. JSMA aims to find the salient points by calculating the positive derivative to find the input perturbations that cause the classifier model to be fooled.

2.2.8. NewtonFool [33]

Jang et al. [33] proposed the NewtonFool algorithm, which decreases the probability of the original class label by applying Newton's method to solve nonlinear equations. This attack gradually searches for the minimal perturbation that affects the classification. NewtonFool produces effective perturbations and significantly reduces the accuracy of the trained model.

2.2.9. Wasserstein Attack [34]

Wong et al. [34] proposed the Wasserstein attack, which generates adversarial examples using the Wasserstein distance rather than traditional norm-based metrics. The Wasserstein distance arises from an optimal transport formulation that computes the minimum cost of moving probability mass. To generate Wasserstein adversarial samples for image-classification models, the authors employed a method that projects onto the Wasserstein ball and modifies the Sinkhorn iterations.

2.3. Existing Studies

First, Zhang [9] conducted an FGSM-based adversarial example attack on CNN-based image steganalysis models such as XuNet. A limitation of this approach is that the adversarial perturbation can corrupt the embedded message, rendering message extraction impossible. To mitigate this issue, the authors applied the adversarial perturbation to the cover image prior to embedding the secret message, thereby attempting to bypass the steganalysis model. Reported error rates for XuNet ranged from 25.54% to 86.71%, and for YeNet from 21.34% to 74.26%. However, this work has experimental limitations: only a single adversarial method was applied, and perturbations were added only to the cover image. In some cases, the injected noise was severe, making the perturbation perceptible to the human visual system.

Second, Tang [10] proposed a novel steganography approach that leverages adversarial examples to evade steganalysis. The proposed method, called ADV-EMB, applies adversarial perturbations to regions where steganography is not used, thereby deceiving the steganalysis model. By integrating steganography with adversarial-example techniques, the researchers demonstrated successful attacks against CNN-based steganalysis. To identify effective combinations of steganography and adversarial attacks, they conducted comparative experiments to determine which adversarial methods can be applied most aggressively and successfully.

Third, Shang et al. [35] demonstrated that GAN-based steganography incorporating adversarial-example generation via backpropagation can effectively evade CNN-based steganalysis models. The proposed technique generates perturbations through backpropagation to compensate for weaknesses in conventional embedding methods. Their approach reduced detection accuracies from 83.5% and 86.7% to 78.5% and 2%, respectively, for the evaluated steganalysis models. However, the applied perturbations were often excessive, and the resulting stego images exhibited PSNR values below 30 dB.

Additionally, Din et al. [36] proposed an image-agnostic perturbation method for steganography-based adversarial attacks. Based on transform-domain steganography, this method identifies frequency bands that encourage classification as a cover image and inserts perturbations that transcend these bands rather than directly modifying individual pixels.

Finally, Li et al. [37] addressed limitations of traditional pixel-modification steganography, which is vulnerable to statistical steganalysis. They proposed embedding secret messages during the generation of AI-generated, art-style images. Given a content image,

a style image, and a secret message, their encoder–decoder neural network generates a stylized image that contains the hidden information. An adversarial training strategy is used to improve the imperceptibility of the stego image and to reduce detection by steganalysis models.

3. Performance Evaluation

3.1. Experimental Purpose and Procedures

The main goal of this experiment is to compare and analyze how effectively nine state-of-the-art adversarial example methods can evade detection by three representative CNN-based image steganalysis ML models in terms of classification accuracy (CA), miss detection rate (MDR), attack success index (ASI), and adversarial example generation time (AEGT).

To this end, we conducted extensive comparative experiments following the steps and methods outlined below (see Figure 2). For our ML experimental platform, we utilized Google Colab Pro (Intel Xeon CPU, 2.00 GHz; Tesla P100; Ubuntu 18.04.5 LTS), and all experiment programs were implemented in Python 3.7.13.

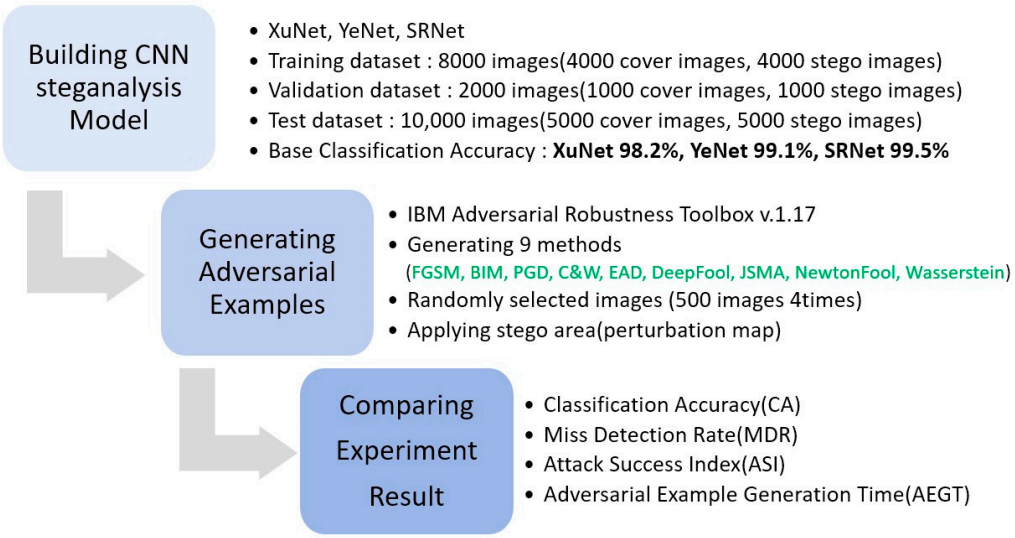


Figure 2. Overview of experimental methods and procedures.

- Step 1 (Preparing image dataset): To train the CNN models, we prepared a total of 40,000 images with a resolution of 256 × 256 pixels. Among them, 20,000 plain images (non-stego images) were collected from the BOSSBase v1.01 dataset [17] and the BOWS2 dataset [18], which are widely used in the steganalysis research field [19–21]. The remaining 20,000 stego images were generated from these plain images using the WOW algorithm.
- Step 2 (Building CNN steganalysis models): We trained three CNN-based steganalysis models (XuNet [13], YeNet [14], and SRNet [15]) using the prepared image dataset. For training, 8000 images (4000 cover and 4000 stego) were used, and for validation, 2000 images (1000 cover and 1000 stego) were employed. For testing, 10,000 images (5000 cover and 5000 stego) were used to measure the base classification accuracy (BCA) of each model. To ensure a fair comparison, all models were trained until full convergence. The resulting BCAs were 98.2% for XuNet, 99.1% for YeNet, and 99.5% for SRNet. XuNet consists of five layers with a softmax activation function and 14,906 parameters. YeNet is composed of nine layers, also using a softmax activation function, with 107,698 parameters. SRNet comprises 26 layers with a softmax activation function and 4,776,962 parameters.

- Step 3 (Generating adversarial images and reapplying stego): For comparison, we evaluated nine adversarial example methods (FGSM [26], BIM [27], PGD [28], C&W [29], EAD [30], Deepfool [31], JSMA [32], NewtonFool [33] and Wasserstein [34]). For each attack method, we randomly selected 2000 (500 × 4) stego images from the dataset and converted them into adversarial stego images using the IBM Adversarial Robustness Toolbox (ART) v1.17. [22] Parameters used for experiments in IBM ART (v1.17) are shown in Table 1. Next, we generated images by reapplying the perturbation map previously obtained from the WOW algorithm.

Table 1. Parameters used for experiments in IBM ART (v1.17).

Attack Method	Epsilon (ε) /Parameter	Iteration Count	Step Size (eps_Step/Step)	Norm Type
FGSM	eps = 0.3	1	Same as ε (single step)	L_∞
BIM	eps = 0.3	100	eps_step = 0.1	L_∞
PGD	eps = 0.3	100	eps_step = 0.1	L_∞
C&W (L2)	confidence = 0.0, initial_const = 0.01	10	learning_rate = 0.01	L_2
EAD	confidence = 0.0, beta = 0.01	10	learning_rate = 0.01	Elastic-net ($L_1 + L_2$)
DeepFool	epsilon = 1×10^{-6}	100	Internal step	L_2
JSMA	theta = 1.0	100	Pixel change = θ	L_0
NewtonFool	eta = 0.1	100	Internal optimization step	L_2
Wasserstein Attack	transport budget = 0.1	200	learning_rate = 0.01	Wasserstein

- Step 4 (Testing and analyzing experiment results): To measure the classification accuracy and missed detection rate of the adversarial example methods, we prepared a test dataset consisting of 2000 (500 × 4) non-stego images and 2000 (500 × 4) adversarial stego images. We then evaluated the CNN-based steganalysis models using this test dataset and recorded the corresponding metrics. In addition, to calculate the Attack Success Index (ASI), we verified whether the hidden messages embedded in undetected adversarial examples could be successfully extracted and whether their PSNR values exceeded 30 or 40 dB. PSNR (Peak Signal-to-Noise Ratio) is commonly used to quantify the reconstruction quality of an image affected by lossy compression or distortion. Several studies have reported that the difference of two images becomes difficult to perceive by the human visual system when the PSNR value exceeds 30 dB [25,38,39]; we use this PSN value as a base threshold. Given a reference image A and a test image B, both of size $M \times N$, the PSNR between A and B is defined as follows:

$$PSNR(A,B) = 10\log\left(\frac{255^2}{MSE(A,B)}\right) \quad (1)$$

$$MSE(A,B) = \frac{1}{AB}\sum_{i=1}^{A}\sum_{j=1}^{B}(A_{ij} - B_{ij}) \quad (2)$$

3.2. Experimental Results and Analysis

3.2.1. Classification Accuracy (CA) and Miss Detection Rate (MDR)

We use two basic metrics (CA and MDR) to examine how severely an adversarial example attack method degrades the overall classification performance of a CNN-based steganalysis model (as measured by CA) and how frequently the model fails to detect actual stego images (as measured by MDR). To this end, we first define the confusion matrix for our classification problem (see Table 2), where True indicates that the actual and predicted values match, and False indicates that they do not.

Table 2. Confusion Matrix for Our Classification Problem.

		Predicted Value	
		Stego	**Non-Stego**
Actual Value	Stego	True Positive (TP)	False Negative (FN)
	Non-Stego	False Positive (FP)	True Negative (TN)

In this study, considering the attacker's objective of evading a CNN-based steganalysis model, we define a classification result as True Positive (*TP*) when the model correctly classifies an actual stego image as stego. Accordingly, a *TP* represents a correct detection, whereas a False Negative (*FN*) corresponds to a missed detection. The Classification Accuracy (CA) and Missed Detection Rate (MDR) are therefore defined by Equations (3) and (4), respectively.

$$\mathrm{CA}(\%) = \left(\frac{TP + TN}{TP + FP + TN + FN}\right) \times 100 \tag{3}$$

$$\mathrm{MDR}(\%) = \left(\frac{FN}{TP + FN}\right) \times 100 \tag{4}$$

We repeated each attack four times using four image datasets, resulting in a total of 2000 images per attack. Table 3 presents example results of each attack method for each model. The averages and standard deviations were then computed accordingly. The standard deviations of XuNet, YeNet, and SRNet were 0.67, 0.30, and 0.28, respectively, and their 95% confidence intervals were calculated as the mean ± 1.07, ± 0.48, and ± 0.45, respectively. The experimental results for CA and MDR are summarized and discussed below (see Tables 4 and 5). In Tables 4 and 5, p-values are reported as mean $\pm$ standard deviation. Significance was tested using paired t-tests with Holm correction, comparing each attack method against the strongest attack (i.e., the one with the lowest mean CA) for each CNN model. Significance levels are marked as * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$. For example, in XuNet, BIM served as the baseline in Table 4.

Table 3. Result of applying adversarial example to stego image (example).

	Clean Image	**XuNet**	**YeNet**	**SRNet**
[26]				
[27]				
[28]				

Table 3. *Cont.*

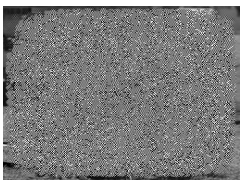
	Clean Image	XuNet	YeNet	SRNet
[29]				
[30]				
[31]				
[32]				
[33]				
[34]				

Table 4 summarizes the classification accuracy (CA) of the three CNN-based steganalysis models (XuNet, YeNet, and SRNet) under various adversarial attack methods, together with the corresponding p-values obtained from paired t-tests relative to the strongest attack method observed for each model. All attack methods significantly reduced the CA of the baseline CNN-based steganalysis models ($p < 0.001$, paired t-test, Holm-corrected). The reduction ranged from 0.7 percentage points (FGSM against YeNet) to 50.4 percentage points (BIM and PGD against YeNet).

These results demonstrate that all adversarial example attacks used in this study statistically and effectively degraded the reliability of representative CNN-based steganalysis models by reducing their classification accuracy.

Table 4. Classification accuracy (CA).

Attack Methods	Classification Acc. (%)			*p*-Value (Adj Holm)		
	XuNet	YeNet	SRNet	XuNet (vs. BIM)	YeNet (vs. PGD)	SRNet (vs. FGSM)
None	98.2	99.1	99.5	-	-	-
FGSM [26]	72.5	98.4	49.9	3.4×10^{-6} (***)	6.06×10^{-6} (***)	-
BIM [27]	54.0	48.7	49.9	-	1	1
PGD [28]	54.3	48.7	49.9	1	-	1
C&W [29]	80.1	95.4	76.8	3.3×10^{-6} (***)	6.32×10^{-6} (***)	1.93×10^{-7} (***)
EAD [30]	93.6	53.9	53.0	2.2×10^{-6} (***)	2.5×10^{-4} (***)	1.08×10^{-3} (**)
DeepFool [31]	95.5	98.3	92.7	2.2×10^{-6} (***)	6.64×10^{-6} (***)	9.82×10^{-7} (***)
JSMA [32]	86.6	87.3	73.4	4.4×10^{-6} (***)	2.44×10^{-5} (***)	8.13×10^{-6} (***)
NewtonFool [33]	95.5	98.3	92.9	2.5×10^{-6} (***)	7.12×10^{-6} (***)	9.82×10^{-7} (***)
Wasserstein [34]	93.7	64.2	53.0	6.8×10^{-6} (***)	2.5×10^{-4} (***)	1.08×10^{-3} (**)
Average	80.6	77.0	65.7	-	-	-

Significance levels are marked as ** for $p < 0.01$, and *** for $p < 0.001$.

Table 5. Miss detection rate (MDR).

Attack Methods	Miss Detection Rate. (%)			*p*-Value (Adj Holm)		
	XuNet	YeNet	SRNet	XuNet (vs. BIM)	YeNet (vs. PGD)	SRNet (vs. FGSM)
None	3.4	1.2	0.2	-	-	-
FGSM [26]	54.6	2.6	100	3.41×10^{-6} (***)	8.36×10^{-7} (***)	-
BIM [27]	91.6	99.4	100	-	7.27×10^{-1}	-
PGD [28]	91.0	99.4	100	1.21×10^{-1}	-	-
C&W [29]	39.4	6.0	46.2	2.64×10^{-8} (***)	2.69×10^{-6} (***)	4.83×10^{-6} (***)
EAD [30]	12.4	89.0	93.8	7.55×10^{-7} (***)	6.65×10^{-5} (***)	2.41×10^{-3} (**)
DeepFool [31]	8.6	2.8	14.4	7.87×10^{-7} (***)	5.22×10^{-6} (***)	2.03×10^{-6} (***)
JSMA [32]	26.8	22.2	53.0	1.43×10^{-8} (***)	1.10×10^{-5} (***)	4.68×10^{-6} (***)
NewtonFool [33]	8.6	2.8	14.0	7.54×10^{-7} (***)	4.93×10^{-6} (***)	2.18×10^{-6} (***)
Wasserstein [34]	12.2	68.4	93.8	7.87×10^{-7} (***)	5.3×10^{-5} (***)	1.45×10^{-3} (**)

Significance levels are marked as ** for $p < 0.01$, and *** for $p < 0.001$.

Among these methods, BIM and PGD exhibited significantly stronger attack performance than the others ($p < 0.001$), particularly against YeNet. By contrast, NewtonFool and DeepFool produced no statistically significant reductions ($p > 0.05$) and were therefore relatively ineffective. Based on the mean CA across all attack methods, SRNet was identified as the most vulnerable CNN model under adversarial attacks.

Furthermore, all attack methods increased the miss detection rate (MDR) of the baseline models by at least 1.4 percentage points (FGSM against YeNet) and up to 99.8 points (FGSM, BIM, and PGD against SRNet). For SRNet, three attack methods (FGSM, BIM, and PGD) completely evaded detection. In particular, PGD and BIM achieved over 90% MDR across all CNN models, outperforming the other attack methods. Conversely, FGSM produced the lowest MDR (2.6%) against YeNet and was not statistically different from JSMA ($p > 0.05$).

Overall, these results consistently indicate that BIM and PGD are the most destructive attacks, while FGSM and JSMA remain relatively weak across all CNN models. Based on the average MDR, SRNet was again the weakest CNN model, whereas XuNet demonstrated the strongest resistance to all attack methods.

Finally, these findings collectively confirm that BIM and PGD represent the most destructive adversarial attack strategies, while FGSM, DeepFool, and NewtonFool are

comparatively weaker. The statistical significance of these differences ($p < 0.05$) confirms that the observed variations are not due to random noise but reflect genuine performance differences among attack methods.

3.2.2. Attack Success Index (ASI)

We introduce a new attack performance metric, ASI (Attack Success Index), which quantifies the degree to which an adversarial example is successfully generated and delivered to the receiver while evading an advanced defense system that incorporates ML-based steganalysis and human-like visual inspection (Defender Lv. 2), as described in Section 2. An adversarial stego image is considered successfully generated—and thus capable of evading Defender Lv. 2—when it satisfies the following three conditions. First, the CNN-based steganalysis model (classifier) must fail to detect it. Second, the image must contain a completely preserved hidden message embedded by a steganographic algorithm. Lastly, it must be visually indistinguishable from its corresponding cover image to human observers; this visual similarity can be evaluated using *PSNR* [25,38,39]. By considering these three factors, we define ASI as shown in Equation (5).

$$\text{ASI} = \left(\frac{FN}{TP + FN}\right) \times \left(\frac{N_{AI} \text{ with } PSNR \geq 30\text{dB}}{N_{AI}}\right) \tag{5}$$

where N_{AI} denotes the number of adversarial stego images, and N_{AI} with $PSNR \geq 30$ dB refers to the subset of adversarial stego images whose *PSNR* values are greater than or equal to 30 dB.

Thus, by using ASI, we can compare two adversarial example attack methods (AM_A and AM_B) such that AM_A with a higher ASI is better than AM_B with a lower ASI. Thus, AM_A can generate more successful adversarial examples (adversarial stego images) and better deliver them to a receiver than AM_A by avoiding an advanced CNN steganalysis model without breaking a steganographic hidden message within them. We note that when ASI = 0, the attack failed either due to full detection (MDR = 0) or perceptible perturbations ($PSNR < 30$ dB).

We report the experimental results of ASI and *PSNR*-based pass rates in Tables 6 and 7, respectively. The values of ASI and PSNR-based pass rate were calculated based on two PSNR threshold values of 30 dB and 40 dB. In this study, since we use the *PSNR* of 30 dB as a base threshold, we first explain the results when *PSNR* = 30 dB and then report results additionally when *PSNR* = 40 dB.

Table 6. Attack Success Index (ASI).

Attack Methods	XuNet		YeNet		SRNet	
	30 db	**40 db**	**30 db**	**40 db**	**30 db**	**40 db**
FGSM [26]	0.2184	0	0	0	0.2000	0
BIM [27]	0	0	0.3976	0	0	0
PGD [28]	0	0	0	0	0	0
C&W [29]	0.3152	0.2679	0.0120	0	0.2772	0.1709
EAD [30]	0.0992	0	0.1780	0.0356	0.1876	0
DeepFool [31]	0.0516	0	0.0224	0	0.1152	0
JSMA [32]	0.2144	0.2144	0.0888	0	0.2120	0
NewtonFool [33]	0.0344	0.0249	0.0168	0.0095	0	0
Wasserstein [34]	0	0	0.2736	0	0.1876	0

Table 7. PSNR-based pass rate.

Attack Methods	XuNet		YeNet		SRNet	
	30 db	**40 db**	**30 db**	**40 db**	**30 db**	**40 db**
FGSM [26]	0.78	0	0	0	0.20	0
BIM [27]	0	0	0.40	0	0	0
PGD [28]	0	0	0	0	0	0
C&W [29]	0.80	0.68	0.20	0	0.60	0.37
EAD [30]	0.80	0.7	0.20	0.04	0.20	0
DeepFool [31]	0.60	0	0.80	0	0.80	0
JSMA [32]	0.80	0.8	0.40	0	0.40	0
NewtonFool [33]	0.40	0.29	0.60	0.34	0	0
Wasserstein [34]	0	0	0.10	0	0.20	0

First, only three attack methods (DeepFool, JSMA, and EAD) successfully bypassed all CNN models (Defender Lv. 2) since their ASI values > 0, and the five attack methods failed to bypass at least one CNN model. An ASI value of 0 indicates that the attack completely failed against both proposed defense mechanisms. Notably, PGD was unable to generate any successful adversarial stego images against any CNN model when evaluated using ASI, even though it was one of the strongest adversarial attack methods in terms of CA and MDR. This finding is particularly interesting and meaningful, as it demonstrates that relying solely on traditional metrics such as CA and MDR is insufficient to evaluate the performance of adversarial attack methods in this domain. For the PGD method, the epsilon (ε) value was set to 0.3 to evade steganalysis, which likely affected the PSNR. Therefore, it is necessary to further analyze the results by adjusting the epsilon value.

Second, C&W exhibited the best attack performance against XuNet and SRNet, while BIM was the most effective against YeNet. For example, approximately 31.5% and 27.7% of adversarial stego images generated by C&W successfully evaded detection by XuNet and SRNet, respectively. Similarly, about 40% of adversarial stego images generated by BIM successfully deceived YeNet.

Finally, YeNet was identified as the weakest CNN model, achieving the highest ASI value (39.7%), whereas SRNet was the strongest CNN model, exhibiting the lowest ASI value (27.7%).

We now report the ASI results when the PSNR threshold is 40 dB. When the ASI was computed based on the 40 dB threshold, the C&W, JSMA, and NewtonFool attack methods in XuNet, the NewtonFool and EAD attacks in YeNet, and only the C&W attack in SRNet bypassed Defender Lv. 2. Meanwhile, all other attack methods failed to satisfy the 40 dB PSNR criterion and consequently yielded ASI values of zero. As shown in Table 7, when the threshold was set to 40 dB, the number of cases with a PSNR-based pass rate of 0 increased across models, thus making it difficult to compare the attack methods individually. This result is reasonable because a higher *PSNR* value represents a stricter threshold. In this study, we do not aim to study finding the optimal *PSNR* threshold; rather, following prior studies [25,38–43], we used 30 dB as a representative and widely accepted perceptual boundary for ASI computation.

3.2.3. Adversarial Example Generation Time (AEGT)

To compare the computational efficiency of the attack methods, we introduce the metric AEGT (Average Example Generation Time), defined as the average time in seconds required to generate a single adversarial stego image for each attack method. The AEGT results are reported in Table 8. In this analysis, we focus exclusively on computational speed and do not consider the attack validity (see previous sections).

Table 8. Adversarial example generation time (AEGT).

Attack Methods	XuNet	YeNet	SRNet
FGSM [26]	0.002 s	0.017 s	0.027 s
BIM [27]	0.108 s	0.924 s	0.960 s
PGD [28]	0.058 s	0.925 s	1.010 s
C&W [29]	5.168 s	8.647 s	7.164 s
EAD [30]	10.35 s	4.507 s	54.951 s
DeepFool [31]	0.776 s	0.370 s	4.005 s
JSMA [32]	5.376 s	11.432 s	24.170 s
NewtonFool [33]	0.590 s	2.544 s	2.462 s
Wasserstein [34]	1.212 s	1.276 s	3.051 s
Average	2.627 s	3.405 s	10.867 s

First, FGSM consistently demonstrated the fastest AEGT across all CNN models, whereas JSMA was the slowest method for YeNet and EAD was the slowest for XuNet and SRNet. Specifically, FGSM generated adversarial stego images approximately 5172 times and 2035 times faster than EAD on XuNet and SRNet, respectively. Similarly, compared to JSMA on YeNet, FGSM was about 672 times faster.

Second, the mean AEGTs across all attack methods were 2.627 s for XuNet, 3.404 s for YeNet, and 10.807 s for SRNet. These results indicate that generating adversarial stego images against SRNet requires the longest average time, whereas XuNet requires the shortest.

4. Conclusions and Future Works

In this paper, we first formally described the system model, which comprises four participants: a naïve attacker, an adversarial attacker, and two types of defenders (Defender Lv. 1 and Defender Lv. 2). Next, we conducted extensive experiments comparing nine adversarial example attack methods (FGSM, DeepFool, JSMA, BIM, C&W, EAD, NewtonFool, PGD, and Wasserstein) against three representative CNN-based steganalysis models (XuNet, YeNet, and SRNet) using multiple evaluation metrics, including classification accuracy (CA), missed detection rate (MDR), attack success index (ASI), and adversarial example generation time (AEGT).

Our main findings are as follows. First, conventional metrics such as CA and MDR do not fully capture how effectively adversarial attackers can evade defenders. Second, our proposed metric, ASI, more appropriately quantifies the degree to which adversarial examples are successfully generated and delivered to the receiver while evading an advanced defense that combines CNN-based steganalysis with human-like visual inspection (Defender Lv. 2). Finally, in terms of ASI, C&W achieved the best performance against XuNet and SRNet, whereas BIM was most effective against YeNet.

For our future work, we plan to extend this study in several directions as follows.

First, we will conduct an in-depth investigation into the strengths and weaknesses of various adversarial attack methods when applied to state-of-the-art steganalysis models. In particular, we aim to understand why certain methods perform better than others with respect to specific performance metrics and model architectures.

Second, while the current study provided meaningful results based on the WOW steganography algorithm, we plan to extend our approach to other steganographic schemes such as UNIWARD and HUGO. By doing so, we will conduct a more comprehensive comparative analysis and explore more effective attack and defense strategies under diverse steganographic conditions.

Third, we intend to analyze variations in message recovery and message recovery rates under different adversarial attack methods. Such analyses will provide valuable insight into the trade-offs between attack success and payload integrity, thereby contributing to a more holistic understanding of adversarial impacts on steganographic communication.

Finally, we aim to develop an enhanced adversarial example generation method that surpasses existing state-of-the-art attacks by explicitly optimizing it for our proposed metric, ASI. We expect that incorporating ASI into the optimization objective will lead to more effective and realistic adversarial examples that can better challenge advanced CNN-based steganalysis systems.

Author Contributions: Conceptualization, H.K. and H.P.; Methodology, H.K. and Y.C.; Software, H.K. and H.P.; Validation, H.K., H.P. and Y.C.; Formal analysis, H.K. and Y.C.; Investigation, H.K.; Resources, H.K.; Data curation, H.K. and H.P.; Writing—original draft, H.K.; Writing—review & editing, Y.C.; Visualization, H.K.; Supervision, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. The BOSSbase dataset is available at: http://dde.binghamton.edu/download/ and the BOWS-2 dataset is available at: https://data.mendeley.com/datasets/kb3ngxfmjw/1 (accessed on 10 November 2025).

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

References

1. Johnson, N.F.; Jajodia, S. Exploring Steganography: Seeing the Unseen. *Computer* **1998**, *31*, 26–34. [CrossRef]
2. Fridrich, J.; Goljan, M.; Du, R. Reliable Detection of LSB Steganography in Color and Grayscale Images. In Proceedings of the ACM Workshop on Multimedia and Security, Ottawa, ON, Canada, 5 October 2001; pp. 27–30.
3. Kodovský, J.; Fridrich, J. Steganalysis of JPEG Images Using Rich Models. In Proceedings of the SPIE—The International Society for Optical Engineering, Brussels, Belgium, 16–18 April 2012; Volume 8303, p. 83030A.
4. Pevný, T.; Filler, T.; Bas, P. Using High-Dimensional Image Models to Perform Steganalysis. In *Proceedings of SPIE—The International Society for Optical Engineering*; SPIE: Bellingham, WA, USA, 2010; Volume 7541, p. 754105.
5. Qian, Y.; Dong, J.; Wang, W. Deep Learning for Steganalysis via Convolutional Neural Networks. In Proceedings of the SPIE Media Watermarking, Security and Forensics, San Francisco, CA, USA, 9–11 February 2015; Volume 9409, p. 94090J.
6. Ker, A.D.; Pevný, T.; Bas, P.; Filler, T. Moving Steganography and Steganalysis from the Laboratory into the Real World. In Proceedings of the 1st ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), New York, NY, USA, 17–19 June 2013; pp. 1–10.
7. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Learning and Transferring Representations for Image Steganalysis Using CNNs. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, USA, 11–15 July 2016; pp. 134–139.
8. Akhtar, N.; Mian, A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
9. Zhang, J.; Zhang, W.; Cao, X.; Yu, N. Adversarial Examples in Deep Learning for Steganography and Steganalysis. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), Paris, France, 3–5 July 2019; pp. 5–10.
10. Tang, W.; Tan, S.; Li, B.; Huang, J. Automatic Steganographic Distortion Learning Using Adversarial Networks. *IEEE Signal Process. Lett.* **2020**, *27*, 1660–1664. [CrossRef]
11. Liu, F.; Tan, S.; Li, B.; Huang, J. Adversarial Embedding for Image Steganography. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2142–2155.
12. Chen, M.; Qian, Z.; Luo, W. Adversarial Embedding Networks for Image Steganography. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2670–2674.
13. Xu, G.; Wu, H.-Z.; Shi, Y.-Q. Structural Design of Convolutional Neural Networks for Steganalysis. *IEEE Signal Process. Lett.* **2016**, *23*, 708–712. [CrossRef]
14. Ye, J.; Ni, J.; Yi, Y. Deep Learning Hierarchical Representations for Image Steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2545–2557. [CrossRef]
15. Boroumand, M.; Chen, M.; Fridrich, J. Deep Residual Network for Steganalysis of Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1181–1193. [CrossRef]

16. Rah, Y.; Cho, Y. Reliable Backdoor Attack Detection for Various Size of Backdoor Triggers. *Int. J. Artif. Intell.* **2025**, *14*, 650–657. [CrossRef]
17. Bas, P.; Filler, T.; Pevný, T. Break Our Steganographic System—The Ins and Outs of Organizing BOSS. In Proceedings of the Information Hiding Conference (IH), Prague, Czech Republic, 18–20 May 2011; pp. 59–70.
18. Bas, P.; Furon, T. BOWS-2. Available online: https://data.mendeley.com/datasets/kb3ngxfmjw/1 (accessed on 10 November 2025).
19. Cogranne, R.; Bas, P.; Fridrich, J. The ALASKA Steganalysis Challenge: A Step Toward Steganalysis at Scale. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), Paris, France, 3–5 July 2019; pp. 1–10.
20. Cogranne, R.; Bas, P.; Fridrich, J. Quantitative Steganalysis: Estimating the Payload and Detecting the Embedding Algorithm. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1045–1058.
21. Holub, V.; Fridrich, J. Universal Distortion Function for Steganography in an Arbitrary Domain. *EURASIP J. Inf. Secur.* **2015**, *2015*, 1–13. [CrossRef]
22. Nicolae, M.-I.; Sinn, M.; Tran, M.N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. *Adversarial Robustness Toolbox v1.0.1*; IBM Research Technical Report; IBM Research: New York, NY, USA, 2019.
23. Pevný, T.; Filler, T.; Bas, P. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography (HUGO). In Proceedings of the Information Hiding Conference (IH), Calgary, AB, Canada, 28–30 June 2010; pp. 99–114.
24. Holub, V.; Fridrich, J. Designing Steganographic Distortion Using Directional Filters (WOW). In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, Spain, 2–5 December 2012; pp. 234–239.
25. Huang, C.-T.; Shongwe, N.S.; Weng, C.-Y. Enhanced Embedding Capacity for Data Hiding Approach Based on Pixel Value Differencing and Pixel Shifting Technology. *Electronics* **2023**, *12*, 1200. [CrossRef]
26. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
27. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Examples in the Physical World. *arXiv* **2017**, arXiv:1607.02533. [CrossRef]
28. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
29. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the IEEE Symposium on Security and Privacy (S&P), San Jose, CA, USA, 22–24 May 2017; pp. 39–57.
30. Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; Hsieh, C.-J. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 10–17.
31. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
32. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 21–24 March 2016; pp. 372–387.
33. Jang, U.; Wu, X.; Chen, S. Objective Metrics and Gradient Descent Algorithms for Adversarial Examples in Machine Learning. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), Orlando, FL, USA, 4–8 December 2017; pp. 277–289.
34. Wong, E.; Schmidt, F.; Metzen, J.H.; Kolter, J.Z. Wasserstein Adversarial Examples via Projected Sinkhorn Iterations. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6808–6817.
35. Shang, Y.; Jiang, S.; Ye, D.; Huang, J. Enhancing the Security of Deep Learning Steganography via Adversarial Examples. *Mathematics* **2020**, *8*, 1446. [CrossRef]
36. Din, S.U.; Akhtar, N.; Younis, S.; Shafait, F.; Mansoor, A.; Shafique, M. Steganographic Universal Adversarial Perturbations. *Pattern Recognit. Lett.* **2020**, *135*, 146–152. [CrossRef]
37. Li, L.; Li, X.; Hu, X.; Zhang, Y. Image Steganography and Style Transformation Based on Generative Adversarial Network. *Mathematics* **2024**, *12*, 615. [CrossRef]
38. Hsiao, T.-C.; Liu, D.-X.; Chen, T.-L.; Chen, C.-C. Research on Image Steganography Based on Sudoku Matrix. *Symmetry* **2021**, *13*, 387. [CrossRef]
39. Weng, C.-Y.; Weng, H.-Y.; Huang, C.-T. Expansion High Payload Imperceptible Steganography Using Parameterized Multilayer EMD with Clock-Adjustment Model. *EURASIP J. Image Video Process.* **2024**, *2024*, 37. [CrossRef]
40. Gowda, S.N.; Yuan, C. StegColNet: Steganalysis Based on an Ensemble Colorspace Approach. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Padua, Italy, 21–22 January 2021; pp. 319–328.
41. Kombrink, M.H.; Geradts, Z.J.M.H.; Worring, M. Image Steganography Approaches and Their Detection Strategies: A Survey. *ACM Comput. Surv.* **2024**, *57*, 1–40. [CrossRef]

42. Luo, W.; Liao, X.; Cai, S.; Hu, K. A Comprehensive Survey of Digital Image Steganography and Steganalysis. *APSIPA Trans. Signal Inf. Process.* **2024**, *13*, e30. [CrossRef]
43. Song, B.; Hu, K.; Zhang, Z. A Survey on Deep-Learning-Based Image Steganography. *Expert Syst. Appl.* **2024**, *254*, 124390. [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.