electronics

MDPI

Article

Harnessing the Adversarial Perturbation to Enhance Security in the Autoencoder-Based Communication System

Zhixiang Deng * and Qian Sang

College of Internet of Things Engineering, Hohai University, Changzhou 213022, China; sangqian_hohai@163.com

* Correspondence: dengzhixiang@hhuc.edu.cn

Received: 17 January 2020; Accepted: 6 February 2020; Published: 8 February 2020

Abstract: Given the vulnerability of deep neural network to adversarial attacks, the application of deep learning in the wireless physical layer arouses comprehensive security concerns. In this paper, we consider an autoencoder-based communication system with a full-duplex (FD) legitimate receiver and an external eavesdropper. It is assumed that the system is trained from end-to-end based on the concepts of autoencoder. The FD legitimate receiver transmits a well-designed adversary perturbation signal to jam the eavesdropper while receiving information simultaneously. To defend the self-perturbation from the loop-back channel, the legitimate receiver is re-trained with the adversarial training method. The simulation results show that with the scheme proposed in this paper, the block-error-rate (BLER) of the legitimate receiver almost remains unaffected while the BLER of the eavesdropper is increased by orders of magnitude. This ensures reliable and secure transmission between the transmitter and the legitimate receiver.

Keywords: deep learning; physical layer security; autoencoder communication system; adversarial attacks; adversarial training

1. Introduction

Traditionally, the communication systems are usually described by various theories and mathematical models from information theory. The communication system itself is often abstracted into three blocks: An encoder at the transmitter, a noisy channel, and a decoder at the receiver. The blocks of the communication system are designed and optimized separately. However, in the practical communications, precise mathematical models are difficult to express, and global optimality cannot be guaranteed due to the local optimization of the separate blocks [1].

With the development and advancement of the deep learning (DL) technology, it has been successfully applied in various fields, such as computer vision, data mining, and natural language processing. Due to its rapid processing capability and powerful optimization capability, researchers have exploited the potential applications of DL to the communication systems with the block structure and the communication systems with the end-to-end structure merging all the blocks [1,2]. Specifically, it was shown in [2] that the transmitter, the channel, and the receiver can be represented by deep neural networks (DNNs) that can be trained as an autoencoder and can achieve close performance to the practical baseline techniques.

In this paper, we are interested in both the reliable and secure transmission of the autoencoder based wireless communication system. Secure communication encounters a great challenge owing to the broadcast nature and openness of the wireless channels [3]. The physical layer security approaches utilize the intrinsic channel properties to achieve security transmission [4]. To improve

the physical layer security, friendly jammers transmit artificial noise to jam the potential eavesdropper. The artificial noise is usually assumed to be white Gaussian noise. On the other hand, a malicious node existed in the communication system may transmit artificial noise to attack the legitimate receiver in decoding the messages. The received signal at the legitimate receiver will be interfered with severely and this will lead to high block error rate (BLER).

In the DL based communication systems such as the autoencoder based wireless communication system considered in this paper, if a malicious node transmits a well-designed perturbation signal sought in the feature space, erroneous predictions of the classification models will be caused, since the DNNs are highly vulnerable to adversarial attacks [5–8]. This raises security and robustness concerns about the applications of deep learning in the physical layer. For example, high modulation classification errors [9] and high BLER in the autoencoder based communication system [10] are caused with slight perturbations added to the original inputs.

Therefore, in the autoencoder based communication system, how to defend the adversarial attacks from the malicious nodes to achieve reliable transmission is one of the concerns in this paper. On the other hand, if an external neural network-based eavesdropper exists in the system, friendly helpers may also design adversarial perturbation signals to attack the eavesdropper such that the eavesdropper is confounded in decoding the confidential message. This motivates us to consider an autoencoder-based communication system with a full-duplex (FD) legitimate receiver and an external eavesdropper. It is assumed that the system is trained from end-to-end based on the concepts of autoencoder. The FD legitimate receiver transmits a well-designed adversary perturbation signal to jam the eavesdropper while receiving information simultaneously. However, the legitimate receiver may also be jammed through the loop-back channel. Though the self-perturbation signal can be cancelled partly by self-interference cancellation (SIC) technique, the residual self-perturbation will still cause high classification errors of the decoder at the legitimate receiver. Thus, appropriative methods should be adopted to confound the eavesdropper while the self-perturbation is suppressed at the legitimate receiver as much as possible. The key contributions of this paper are summarized as follows:

(1) Two communication scenarios: An anti-attacking communication system and an anti-eavesdropping communication system are considered to study the security performance of the autoencoder based wireless communication.
(2) For the anti-attacking communication system, where a malicious jammer transmits adversarial perturbation signal to attack the legitimate receiver, the adversarial training method [11] is used to defend the adversarial attacks from the sneaky jammer. The simulation results show that active adversarial attack from the jammer increase the BLER of the legitimate receiver very slightly, no matter if the autoencoder structure of the jammer is the same as that of the legitimate receiver.
(3) In the anti-eavesdropping end-to-end autoencoder communication system with a FD receiver and a passive eavesdropper, adversarial training method is also adopted to defend the self-perturbation from the loop-back channel at the legitimate receiver. Simulation results show that the BLER of the eavesdropper is increased by orders of magnitude, while the BLER of the legitimate receiver is almost unchanged.

These results indicate the potential of the proposed anti-attacking and anti-eavesdropping autoencoder communication system in both reliable and secure transmission.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the system model considered in this paper. The proposed adversarial training scheme is introduced in detail in Section 4. Section 5 provides simulation results, and conclusions are given in Section 6.

2. Related Work

In recent years, deep learning has begun to be applied in the communication systems. DL could be model-driven or data-driven, or a combination of the two methods. For the applications of DL in

the communication systems with the block structure, the DL approaches are usually model-driven, and are combined with the communication domain knowledge [12]. DL has been applied to refine the conventional block-structure communications, including the modulation recognition [13], the channel estimation and detection [14–16], and the channel decoding [17,18]. The data-driven DL approaches in physical communications treat the entire communication system as an end-to-end reconstruction task. The modules of the traditional communications systems are replaced by DNNs, which are trained in a supervised learning manner to optimize the end-to-end performance [2,19,20]. In [2], the communication was interpreted as an autoencoder, and the communication system design was considered as an end-to-end reconstruction task. In [20], it was verified that BLER of the "learned" communication system was close to the practical baseline techniques. The application of the autoencoder based communication system achieves the close performance without extensive communication theoretic analysis and enables the system to cope with new channel scenarios.

Wireless physical layer security has also received considerable attention recently. The existing works focusing on the physical layer security of the conventional communication systems can be divided into two categories: Active jamming and passive eavesdropping [21]. A malicious node in the communication system may transmit random jamming signals to degrade the legitimate receiver and tries to eavesdrop on the confidential messages of the legitimate users [22]. A passive eavesdropper only tries to decode the secret information of the legitimate users. Different from the key-based encryption technique, physical layer security based on information theory takes full advantage of wireless channel inherent characteristics to enforce the security performance in terms of information-theoretic security, beamforming, cooperative relaying, and artificial jamming [23–25]. For example, to improve the secrecy rate of the confidential message, friendly jammers may transmit artificial noise to confound the eavesdropper. In [26], the full-duplex receiver transmits jamming noise to degrade the eavesdropper channel. However, the legitimate receiver will also be interfered with by the jamming noise from the self-interference channel, since the self-interference cannot be cancelled completely in practice.

As shown in [5–8], DNNs are very vulnerable to adversarial attacks. Recently, some related works focus on the security problems of the applications of DL in the communication systems. As shown in [9,10], high classification errors were caused with slight perturbations added to the original inputs. In [9], DL-based radio signal classification task was considered, and white-box and black-box adversarial attacks were specifically designed to cause misclassification of a DL-based modulation classifier with extremely small perturbation power. In [10], the vulnerability of end-to-end communication systems based on autoencoder-like network was presented, and the algorithms of crafting physical adversarial attacks were proposed to effectively increase the BLER of a communication system. The results showed that the BLER of the communication system was increased by orders of magnitude under the adversarial attack. It was revealed that the adversarial attacks were more destructive than jamming attacks. In [27], the end-to-end learning of communication systems with autoencoders was extended to a scenario in which an eavesdropper must be kept ignorant about the legitimate communication. It was shown that the secrecy of the transmission was achieved by utilizing a modified secure loss function based on cross-entropy, and the neural network could learn a trade-off between reliable communication and information secrecy.

In this paper, based on the ideas in [10] and different from the work [27], we consider the autoencoder based wiretap channel with a FD legitimate receiver, which transmits the adversarial perturbation to confound the eavesdropper.

3. System Model

Due to the broadcast nature and openness of the wireless channel, a legitimate receiver may encounter the active attack of a jammer or an external eavesdropper may wiretap the confidential information which the transmitter wants to secretly transmit to the intended receiver. As shown in Figure 1, we consider two types of wiretap system models. In Figure 1a, a transmitter Alice communicates with a legitimate receiver Bob, while an active jammer transmits a well-designed adversarial perturbation signal to jam the transmissions between Alice and Bob. In Figure 1b, a

passive eavesdropper tries to eavesdrop information of the legitimate users Alice and Bob. The legitimate receiver Bob works in full-duplex mode and is deployed with two antennas, one for transmitting and the other for receiving. To confound Eve, Bob transmits the adversarial perturbation signal while receiving information from Alice simultaneously.

We implement the communication scenario using an end-to-end autoencoder-like network setting as that in [2]. The transmitter (called encoder) and the receiver (called decoder) are represented as fully connected DNNs or other neural networks, such as convolutional neural network (CNN), long short-term memory (LSTM). Accordingly, the additive white Gaussian noise (AWGN) channel from the transmitter to the receiver is represented as a simple noise layer with certain variance. The end-to-end autoencoder communication systems with respect to Figure 1a,b are shown in Figure 2a,b, respectively.

In the encoding stage, Alice encodes a message $s \in \mathcal{M} = \{1, 2, ..., M\}$ to the transmitted signal $\mathbf{x} = f(s) \in \mathbb{R}^{2N}$ with average power constraint $\mathrm{E}\left[\left|x_i^2\right|\right] \le 0.5 \ \forall i$, where $f : \mathbb{R} \to \mathbb{R}^{2N}$ and $M = 2^k$ is the cardinality of the message set $\mathcal{M}$, with k being the number of bits per message. Note that the output of Alice is a N-dimensional complex vector, which is transformed to a $2N$-dimensional real vector. Alice transmits signal $\mathbf{x}$ to Bob using the channel N times. We set $N = 7$ and $k = 4$. Accordingly, the rate in bits per channel use is $R = k / N = 4/7$.

In the decoding stage, Bob tries to decode a message from the received signal as correctly as possible. As shown in Figure 2a, a well-designed perturbation signal $\mathbf{p}$ is transmitted by a malicious jammer, whose goal is to increase the BLER of the legitimate receiver Bob. Bob receives the signal $\mathbf{y}_1$, which is the superposition of the signal $\mathbf{x}$ transmitted from Alice, the AWGN $\mathbf{z}$ of the channel, and the adversarial perturbation signal $\mathbf{p}$ transmitted from the jammer, namely $\mathbf{y}_1 = \mathbf{x} + \mathbf{p} + \mathbf{z}$. Bob, with multiple dense layers followed by the softmax activation layer, decodes the signal $\mathbf{y}_1$ to a M-dimensional probability vector $\hat{\mathbf{s}} = g_1(\mathbf{y}_1) \in \mathbb{R}^M$, where $\hat{\mathbf{s}}$ is the estimate of the one-hot message vector $\mathbf{s}$, where $\sum_i \hat{s}_i = 1, \ \hat{s}_i \ge 0$ and $g_1 : \mathbb{R}^{2n} \to \mathbb{R}^M$. The index of the largest element of $\hat{\mathbf{s}}$ determines one of the M possible messages $\hat{S}$. Different from Figure 2a, Figure 2b shows that the full-duplex receiver Bob transmits a well-designed perturbation signal $\mathbf{p}$ to increase the BLER of Eve to degrade Eve from eavesdropping the secret information. Meanwhile, Bob adopts SIC technology to remove part of the self-interference. Accordingly, the signal $\mathbf{y}_1$ at Bob can be represented as $\mathbf{y}_1 = \mathbf{x} + \alpha\mathbf{p} + \mathbf{z}$, where α represents the attenuation coefficient of the self-perturbation signal after SIC. In Figure 2b, Bob, with multiple dense layers followed by the softmax activation layer, decodes the signal $\mathbf{y}_1$ to a M-dimensional probability vector $\hat{\mathbf{s}} = g_1(\mathbf{y}_1) \in \mathbb{R}^M$, which is similar to that of Figure 2a. Note that Eve in Figure 2b receives the signal $\mathbf{y}_2$, which is the superposition of the signal $\mathbf{x}$, the AWGN $\mathbf{z}$, and the adversarial perturbation signal $\mathbf{p}$ transmitted from Bob, namely $\mathbf{y}_2 = \mathbf{x} + \mathbf{p} + \mathbf{z}$. Eve, with multiple dense layers followed by the softmax activation layer, transforms the signal $\mathbf{y}_2$ into a M-dimensional probability vector $\hat{\mathbf{s}}_e = g_2(\mathbf{y}_2) \in \mathbb{R}^M$, where $g_2 : \mathbb{R}^{2n} \to \mathbb{R}^M$. Eve tries to estimate the message $\hat{s}_e$ determined by the index of the largest element of $\hat{\mathbf{s}}_e$.

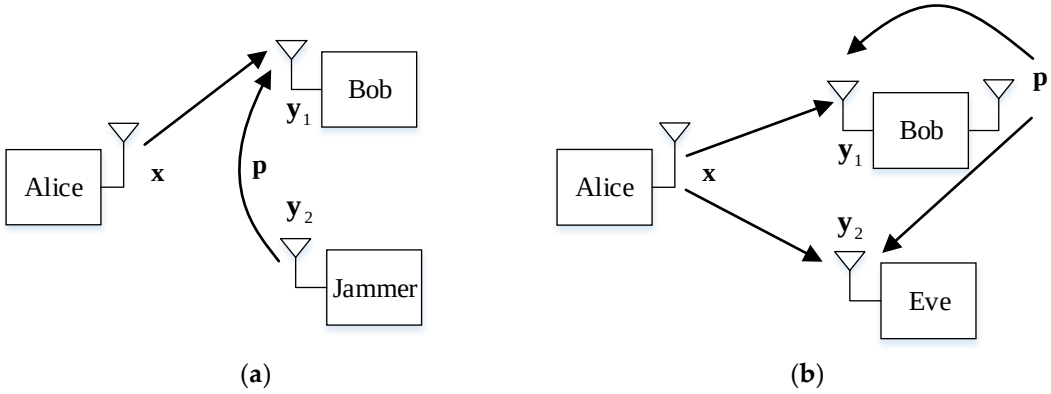
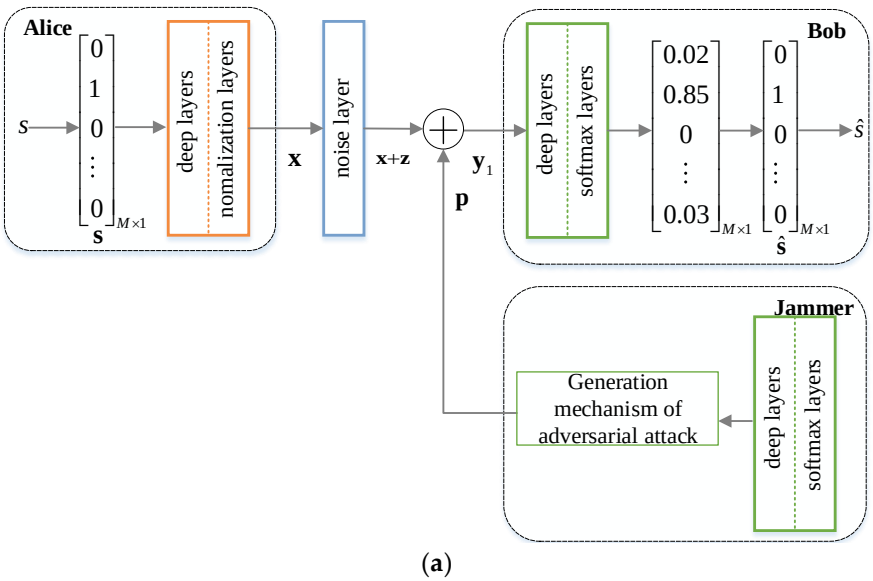
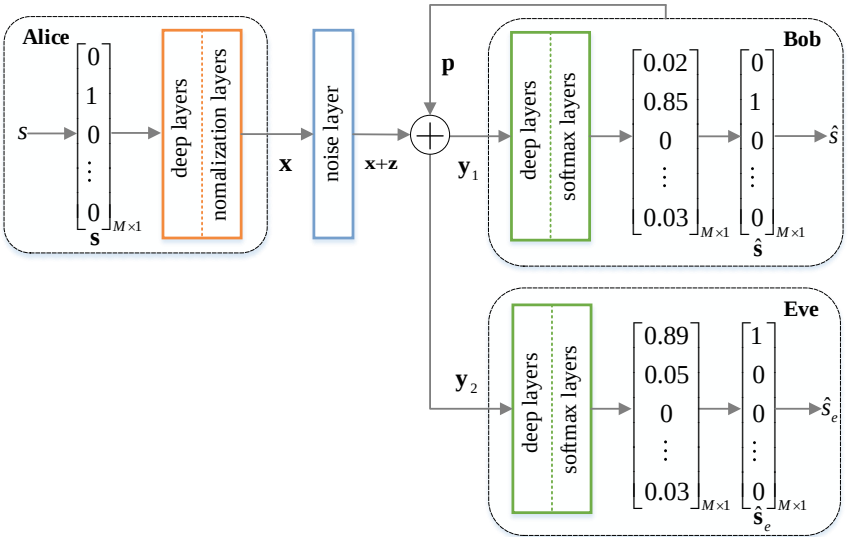


Figure 1. System model. (**a**) Active jamming, where the jammer transmits adversarial perturbation signal. (**b**) Passive eavesdropping, where the full-duplex (FD) Bob transmits adversarial perturbation signal to confound Eve such that the confidential information will be kept as ignorant as possible to Eve.

In the following, we abuse the notation attacker to define the node which transmits the perturbation signal (e.g., the jammer in Figure 2a, the legitimate receiver Bob in Figure 2b), and the target object to define the receiver which is attacked (e.g., Bob in Figure 2a, Eve in Figure 2b). We assume that the attacker has no knowledge about the autoencoder structure of the target object. Thus, two cases are considered: (1) The attacker and the target object use the same autoencoder structure; (2) the autoencoder structures of the attacker and the target object are different. For the first case, both the attacker and the target object adopt the DNN-based autoencoder structure. For the other case, the attacker uses the DNN-based autoencoder structure while the target object uses the CNN-based autoencoder structure. The structures of the two networks are shown in Table 1 in detail. The two autoencoders are both trained by Adam optimizer at a fixed SNR with sparse categorical cross-entropy as the loss function to optimize the BLER performance of the end-to-end communication system.





(**b**)

Figure 2. End-to-end autoencoder model. (a) Active jamming; (b) passive eavesdropping.

The vocabulary table of the variables used is shown in Table 2.

Table 1. The structure of the considered autoencoders.

	DNN Autoencoder		CNN Autoencoder	
Block Name	**Layer Name**	**Output Dim.**	**Layer Name**	**Output Dim.**
Encoder	Input	M	Input	M
	Dense + eLU	M	Dense + eLU	M
	Dense + Linear	2N	Conv1d + Flattening	$16\times M$
	Normalization	2N	Dense + Linear	2N
			Normalization	2N
Channel	Noise (+Perturbaton)	2N	Noise (+Perturbaton)	2N
Decoder	Dense + ReLU	M	Conv2d	$16\times 2N$
	Dense + Softmax	M	Conv2d + Flattening	$8\times 2N$
			Dense + Softmax	2M
			Dense + ReLU	M

Table 2. Vocabulary table.

Variables	Name
$\mathbf{x}$	the transmit signal
$\mathbf{y}_i\,(i=1,2)$	the received signal at the legitimate receiver or eavesdropper
$\mathbf{p}$	the perturbation signal
$\mathbf{z}$	the AWGN of the channel
$\mathcal{M}$	the message set
s	the message to be transmitted
$\hat{\mathbf{s}}$ （$\hat{\mathbf{s}}_e$）	the estimated message at Bob (Eve)
$\mathbf{X}_{adv}$	adversarial example
$\mathbf{X}_{clean}$	clean example

4. Adversarial Attack and Adversarial Training

In the anti-attacking and the anti-eavesdropping end-to-end autoencoder communication systems, the adversarial perturbation signal $\mathbf{p}$ is well-designed by the jammer or the legitimate receiver, which can fully mislead the classification model with very small and imperceptible perturbation

power. Under the adversarial attack, the legitimate receiver will have very high BLER if no additional defense steps are taken because its decoder network model will misclassify the input signal. In order to defend the adversarial attack from the jammer or the self-perturbation, the legitimate receiver improves the robustness of the classification model to adversarial examples through adversarial training.

4.1. Adversarial Attack

The deep neural networks are extremely vulnerable to adversarial perturbation attacks in spite of extraordinary success in solving complicated classification problems. In fact, the very small and imperceptible perturbations fully mislead the state-of-the-art DL-based classifiers, leading to erroneous classification. The reason for the surprising universal perturbations' existence lies in the important geometric correlations among the high dimensional decision boundary of the classifiers [8].

It is assumed that the attacker does not have perfect knowledge about the target object's model, such as the number of the layers, the weights, and the bias parameters. Moreover, we also consider the situation where the adversarial perturbation signal may be not synchronous with the signal transmitted by the transmitter. Considering the transferability of the adversarial attacks, adversarial attacks designed for a specific model can also attack other different models with high probability [9]. It means that the attacker can use its own model as a substitute model to design an adversarial perturbation and then attack the unknown models. In this paper, the attacker will craft universal perturbation vectors according to the second algorithm (we define the algorithm as SIP algorithm) in [10], which involves two important operations: (1) Generate a pool of adversarial perturbations by effectively increasing the loss function leading to incorrect classification with fast gradient symbol method (FGSM); (2) find their main principal direction, which hopefully shows a better shift-invariant property by singular value decomposition (SVD). The adversarial attacks created by the SIP algorithm in [10] are robust for unknown object's model and random time shifts, which indicates that we can ignore the synchronicity requirement. The brief description of SIP algorithm in [10] is shown in the following. For more details on the SIP algorithm (Algorithm 1), please refer to [10].

Algorithm 1 Design Shift-Invariant Perturbations [10]

1: Using the substitute network, generate I adversarial perturbations using FGSM.
2: Calculate the BLER of a randomly shifted version of each of the I perturbations on the substitute network.
3: Select the first t perturbations associated with the n least BLERs. Denote them as $\{\mathbf{p}_1,...,\mathbf{p}_n\}$.
4: Set $\mathbf{P}_{norm} = \left[\frac{\mathbf{p}_1}{\|\mathbf{p}_1\|_2}, \frac{\mathbf{p}_2}{\|\mathbf{p}_2\|_2}, ..., \frac{\mathbf{p}_n}{\|\mathbf{p}_n\|_2}\right]^T$.
5: Calculate the SVD of P norm as $\mathbf{P}_{norm} = \mathbf{U}\sum\mathbf{V}^T$.
6: Select the first column of $\mathbf{V}$ as the candidate shift-invariant perturbation, i.e., $\mathbf{p}_{si} = \mathbf{V}\mathbf{e}_1$.

4.2. Adversarial Training

The main idea of adversarial training is to take adversarial examples as input data to re-train the original classification model. Adversarial examples are made by adding the adversarial perturbations to the original input data, i.e., $\mathbf{X}_{adv} = \mathbf{X}_{clean} + \mathbf{p}$. In order to retain good classification performance for clean examples and improving robustness of the classifier to adversarial attacks, we mix clean and adversarial examples by a certain ratio, e.g., $\{\mathbf{X}^1_{clean},...,\mathbf{X}^m_{clean},\mathbf{X}^1_{adv},...,\mathbf{X}^t_{adv}\}$, where m is the size of the clean examples, and t is the size of the adversarial examples. To improve the generalization ability of the model, the training set needs to be shuffled. The ratio of the clean and adversarial examples is important, as it has an impact on the decoding performance with or without adversarial attacks. The model learns and exploits regularities in the construction process of adversarial attacks because the

adversarial examples use the true label during training. For fast convergence, the parameters of the trained model are used to initialize the network to be trained.

Considering that the end-to-end autoencoder communication system essentially implements classification function, we choose sparse softmax cross entropy as a loss function. Adam Optimizer is adopted, which is robust to a wide range of non-convex optimization problems in the field of deep learning, and can achieve faster convergence rate than normal stochastic gradient descent (SGD) method for sparse features. The learning rate is also one of the important factors affecting the convergence speed. Larger learning rate leads to a higher loss error, while the lower learning rate leads to slower convergence. Therefore, we adopt moderate and frequently-used value 0.001 as the learning rate. The detailed process of adversarial training is described in Algorithm 2, and the flow chart of the training and testing process is also shown in Figure 3.

Algorithm 2 The process of adversarial training

The training set is $\left\{\mathbf{X}_{clean}^{1},...,\mathbf{X}_{clean}^{m},\mathbf{X}_{adv}^{1},...,\mathbf{X}_{adv}^{t}\right\}$

The size of the clean examples is m, the size of the adversarial examples is t, and the size of the training mini-batch is n

1: Use the parameters of the trained model to initialize the network to be trained.
2: Shuffle the training set.
3: **repeat:**
5: Read mini-batch $B=\left\{\mathbf{X}^{1},\mathbf{X}^{2}...,\mathbf{X}^{n}\right\}$ from the training set.
6: Do one-step training to update weights and bias parameters with Adam optimizer.
7: **until** the trained-model is converged.

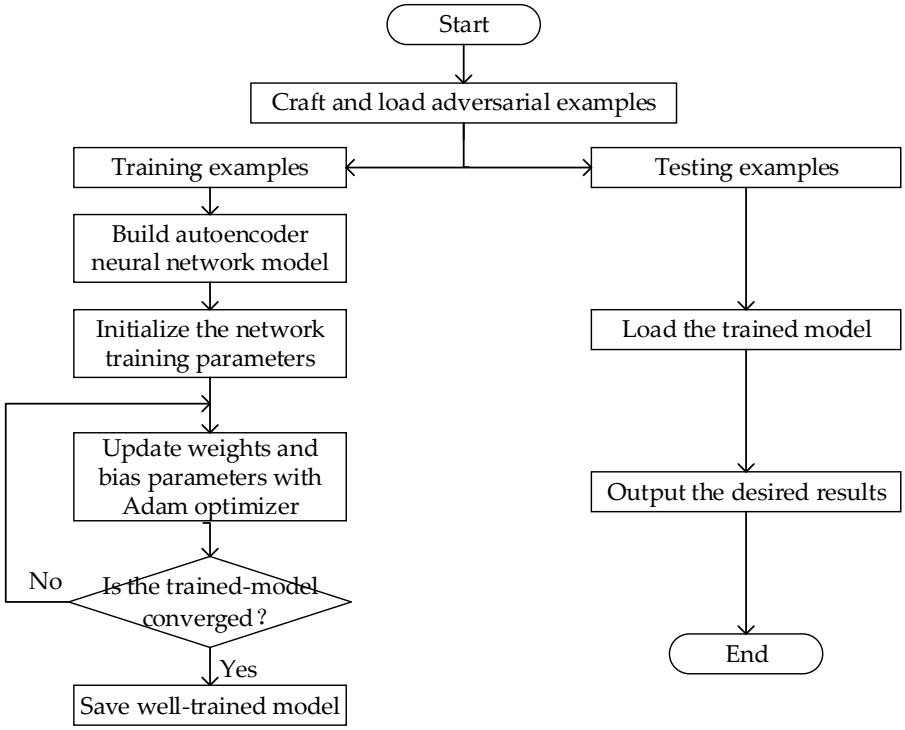


Figure 3. Flow chart of the training and testing process.

5. Numerical Results

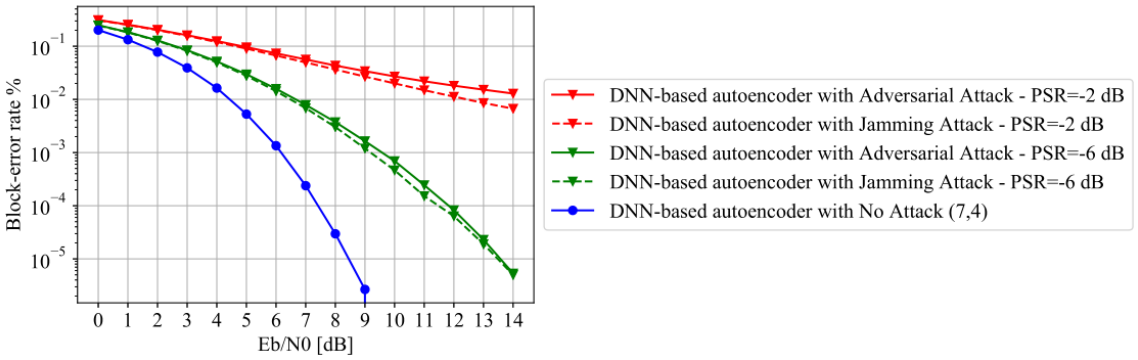
In this section, numerical results are presented to show the performance of the proposed anti-attacking and anti-eavesdropping end-to-end autoencoder communication system. The attenuation coefficient α of the loop-back channel is set to be $5\,\mathrm{dB}$. The two autoencoder models are both trained by setting SNR to be $8.5\,\mathrm{dB}$. The PSR (perturbation-to-signal ratio) is the ratio of the power of the perturbation signal to that of the received signal. The 1,000,000 training examples and the

testing examples are randomly and uniformly generated with a given random seed. We adopt Python 3.6.0 with TensorFlow 1.7.0, and use a Nvidia GTX 1080Ti GPU and 14-core Intel CPU for training and testing, respectively.

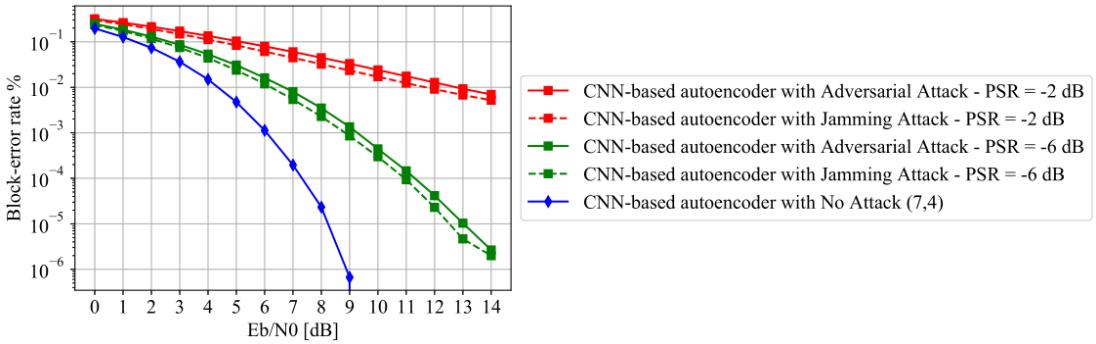
For 10,000 training samples, the training time of well-trained DNN-based and CNN-based autoencoder model are 122.942 and 1450.393 s, respectively, when the learning rate is set to 0.001 and the number of iterations is taken as 10,000. The testing time of well-trained DNN-based and CNN-based autoencoder model are 0.198 and 1.457 s, respectively, for 10,000 testing samples over 1000 tests. In addition, the adversarial training time of DNN-based autoencoder model is 170.916 s, and the prediction time of well-trained model is 0.208 s for 10,000 testing samples.

In the following, Figures 4 and 5 are presented to show the BLER performances of the legitimate receiver under the adversarial attack of a malicious jammer as well as the BLER performances adopting adversarial training method with different ratios of clean to adversarial samples. Figures 6 and 7 are presented to compare the BLER performances of the legitimate receiver and the eavesdropper when the FD receiver transmits adversarial perturbation to confound the eavesdropper.

Figure 4 shows the BLER performance of the legitimate receiver Bob shown in Figure 2a. The structure of Bob is constructed based on DNN or CNN respectively. The universal adversarial perturbation signal $\mathbf{p}$ is created according to the SIP algorithm in [10], assuming that the autoencoder of Alice and Bob is constructed based on the DNN. The adversarial perturbation signal $\mathbf{p}$ is randomly shifted in each testing phase. From Figure 4, it can be observed that the BLER of DNN-based and CNN-based autoencoder are both increased by orders of magnitude, even for very small PSR values under adversarial attack. For the sake of comparison, the traditional jamming attack is also considered. The jammer creates Gaussian jamming signals with the same power as that of the adversarial attack. It can be found from Figure 4 that the BLERs of DNN-based and CNN-based autoencoder under adversarial attack are higher than those under jamming attack. Therefore, adversarial attack is more destructive compared to the jamming attack in some sense. Comparing Figure 4a with Figure 4b, we can observe that the BLERs of CNN-based autoencoder are only slightly lower than that of DNN-based autoencoder. This validates that adversarial attacks designed for a specific model can also attack other unknown models with very high probability.



(**a**)



(**b**)

Figure 4. Block-error-rate (BLER) versus E_b / N_0 under randomly shifted adversarial attacks and jamming attacks. (**a**) autoencoder based on deep neural networks (DNN); (**b**) autoencoder based on convolutional neural network (CNN).

In order to show the effects of the ratio of the clean to adversarial examples on adversarial training, Figure 5 compares the BLER performance of DNN-based autoencoder with different ratios. The DNN-based autoencoder is re-trained with different ratios of clean to adversarial examples. From Figure 5, it can be observed that the BLERs of DNN-based autoencoder with adversarial attack are higher than those with no attack, when the ratio of clean to adversarial examples is set to be 1:9, 2:8, 3:7, or 4:6, respectively. We can also observe that DNN-based autoencoder with adversarial attack has almost the same BLER performance as that with no attack, when the ratio is 5:5. This indicates that with adversarial training method the legitimate receiver can defend the adversarial attack from the jammer, especially when the ratio of clean to adversarial examples is set to be 5:5. Therefore, in the following simulations, the ratio is fixed as 5:5.

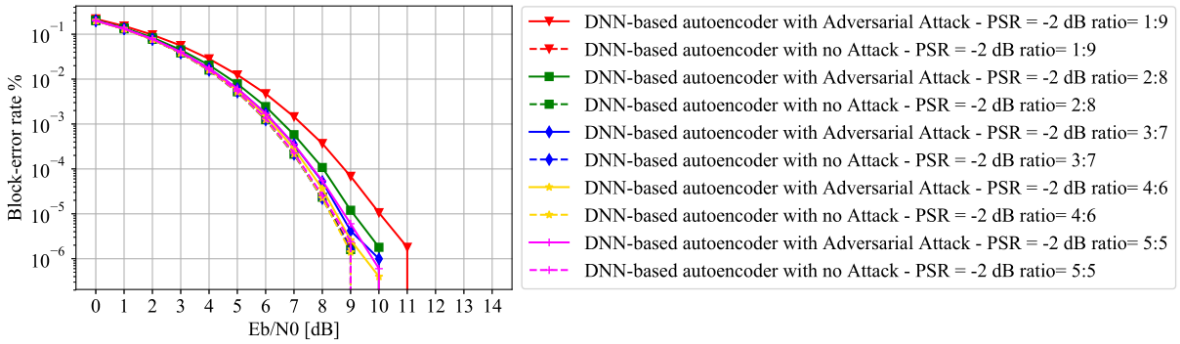


Figure 5. BLER versus E_b / N_0 under randomly shifted adversarial attacks when the DNN-based autoencoder is re-trained with different ratios of clean to adversarial examples.

As shown in Figures 4 and 5, adversarial attack causes significant loss of the BLER performance of the autoencoder-based communication and the adversarial training method can be used to re-train the autoencoder such that the legitimate decoder will defend the adversarial attack. The following Figures 6 and 7 are presented to show the BLER performance of the autoencoder-based wiretap channel considered in this paper.

Figure 6 presents the BLER performance assuming both Bob and Eve employ the DNN network structure as shown in Table 1. From Figure 6, it can be observed that the BLERs of Eve are increased by orders of magnitude even for very small PSR values under adversarial attack. It is worth noticing that the BLERs of Bob are almost unchanged under the adversarial attack with adversarial training and SIC. However, under the random jamming attack, the BLERs of Bob are increased by several orders of magnitude, though the increase is smaller compared to that of Eve. This indicates that the anti-eavesdropping method proposed in this paper not only degrades Eve for eavesdropping information, but also ensures little influence on the reliable transmission between Alice and Bob.

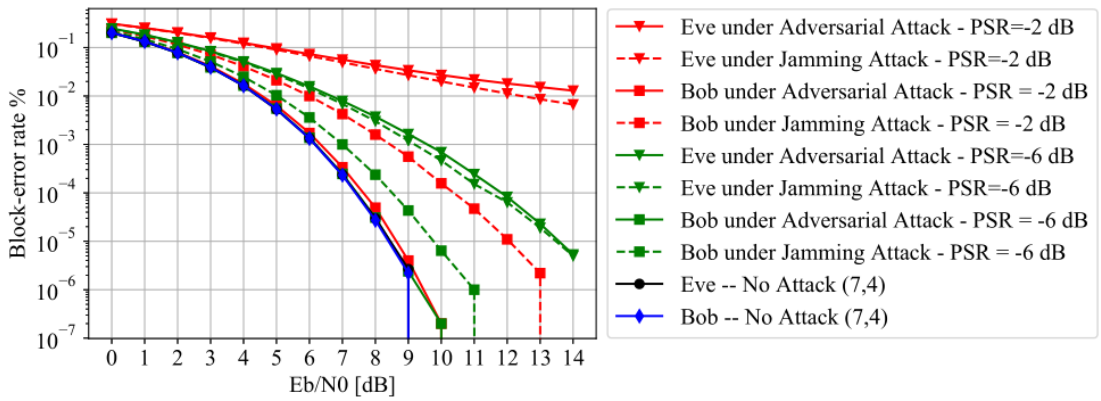


Figure 6. BLER versus E_b / N_0, the autoencoder-based wiretap channel under randomly shifted adversarial attack and artificial noise jamming assuming Bob and Eve use the same DNN network structures.

Figure 7 compares the BLER performance assuming Bob employs the DNN network structure while Eve employs the CNN network structure. Bob uses its own DNN model as a substitute model to generate perturbation signals and then crafts adversarial attacks using the SIP algorithm [10] to attack the CNN decoder model of Eve. From Figure 7, we can also observe that the BLERs of Eve under adversarial attacks are increased by several orders of magnitude and are higher than those under the artificial noise jamming, while the BLERs of Bob almost are almost unchanged under the adversarial attack. Again, Figure 7 shows that the proposed anti-eavesdropping autoencoder communication system can ensure reliable transmission while degrading Eve for eavesdropping secret information under adversarial attacks.

From both Figures 6 and 7, it can be found that, no matter if the legitimate receiver Bob has any knowledge of Eve, it uses its own DNN model as a substitute to generate perturbation signal to jam Eve, and the BLER performance of Eve will be decreased significantly.

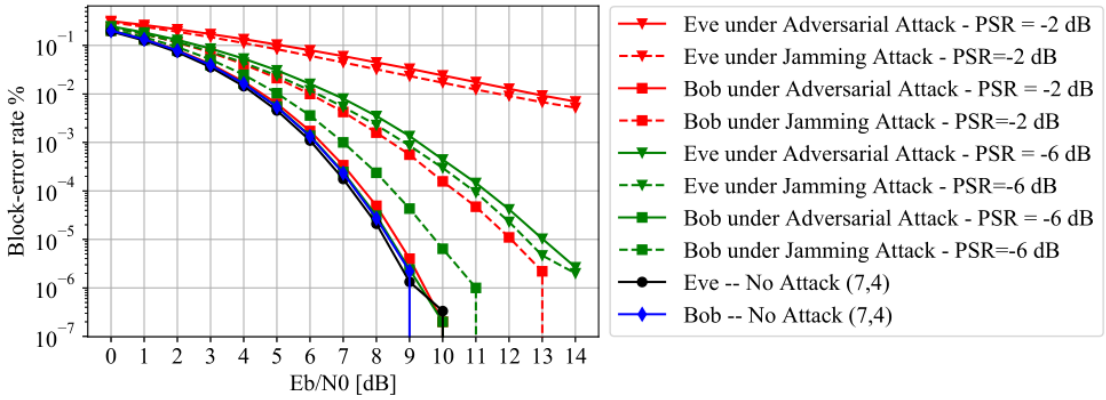


Figure 7. BLER versus E_b / N_0, the autoencoder-based wiretap channel under randomly shifted adversarial attack and artificial noise jamming assuming Bob employs the DNN network structure while Eve employs the CNN network structure.

6. Conclusions

In this paper, we consider an autoencoder based wiretap channel with a full-duplex legitimate receiver and an external eavesdropper. The communication system considered in this paper is assumed to be trained from end-to-end based on the concepts of autoencoder. The full-duplex receiver transmits a well-designed perturbation signal to jam the malicious eavesdropper such that the information of the legitimate users is kept as secret as possible to the eavesdropper. To defend self-perturbation from the loop-back channel, the FD receiver is re-trained, adopting adversarial training method. Simulation results show that under adversarial attacks, the BLER performance of

the legitimate receiver almost remains unaffected in the anti-attacking and anti-eavesdropping communication systems, and the BLERs of the eavesdropper are increased by orders of magnitude in an anti-eavesdropping communication system, which indicates that the proposed anti-attacking and anti-eavesdropping autoencoder communication systems ensure reliable and secure transmission.

Author Contributions: Conceptualization, Z.D.; data curation, Q.S.; formal analysis Z.D. and Q.S.; funding acquisition Z.D.; investigation Q.S.; methodology, Z.D. and Q.S.; writing—original draft preparation, Q.S.; writing—review and editing, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities in China under Grant 2019B22614.

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

References

1. Qin, Z.; Ye, H.; Li, G.Y.; Juang, B.F. Deep learning in physical layer communications. *IEEE Wirel. Commun.* **2019**, *26*, 93–99.
2. O'Shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575.
3. Mukherjee, A.; Fakoorian, S.A.; Huang, J.; Swindlehurst, A.L. Principles of physical layer security in multiuser wireless networks: A survey. *IEEE Commun. Surv. Tuts.* **2014**, *16*, 1550–1573.
4. Wyner, A.D. The wire-tap channel. *Bell Syst. Tech. J.* **1975**, *54*, 1355–1387.
5. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430.
6. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. Available online: https://arxiv.org/abs/1412.6572 (accessed on 20 February 2019).
7. 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, Las Vegas, NV, USA, 26 June – 1 July 2016; pp. 2574–2582.
8. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21-26 July 2017; pp. 86–94.
9. Sadeghi, M.; Larsson, E.G. Adversarial attacks on deep-learning based radio signal classification. *IEEE Trans. Wirel. Commun.* **2019**, *8*, 213–216.
10. Sadeghi, M.; Larsson, E.G. Physical adversarial attacks against end-to-end autoencoder communication systems. *IEEE Commun. Lett.* **2019**, *23*, 847–850.
11. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. In Proceedings of the ICLR, Toulon, France, 24–26 April 2017.
12. He, H.; Jin, S.; Wen, C.; Gao, F.; Li, G.Y.; Xu, Z. Model-driven deep learning for physical layer communications. *IEEE Wirel. Commun.* **2019**, *26*, 77–83.
13. West, N.E.; O'Shea, T. Deep architectures for modulation recognition. In Proceeding of the DySPAN, Piscataway, NJ, USA, 6–9 March, 2017; pp. 1–6.
14. Ye, H.G.; Li, Y.; Juang, B. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Trans. Wirel. Commun.* **2018**, *7*, 114–117.
15. Soltani, M.; Pourahmadi, V.; Mirzaei, A.; Sheikhzadeh, H. Deep learning-based channel estimation. *IEEE Commun. Lett.* **2019**, *23*, 652–655.
16. He, H.; Wen, C.-K.; Jin, S.; Li, G.Y. Deep learning-based channel estimation for beamspace mmWave massive MIMO systems. *IEEE Wireless Commun. Lett.* **2018**, *7*, 852–855.
17. Gruber, T.; Cammerer, S.; Hoydis, J.; Brink, S. T. On deep learning-based channel decoding. In Proceeding of the 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 22–24 March 2017; pp. 1–6.
18. Nachmani, E.; Marciano, E.; Lugosch, L.; Gross, W.J.; Burshtein, D. Deep learning methods for improved decoding of linear codes. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 119–131.

19. Ye, H.; Liang, L.; Li, G.Y. Deep learning based end-to-end wireless communication systems with conditional GAN as unknown channel. *arXiv* **2019**, arXiv:1903.02551. Available online: https://arxiv.org/abs/1903.02551v1 (accessed on 10 April 2019).
20. Dörner, S.; Cammerer, S.; Hoydis, J.; Brink, S.T. Deep learning based communication over the air. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 132–143.
21. Shiu, Y.S.; Chang, S.Y.; Wu, H.C.; Huang, C.H.; Chen, H.H. Physical layer security in wireless networks: A tutorial. *IEEE Wirel. Commun.* **2011**, *18*, 66–74.
22. Fang, S.; Liu, Y.; Ning, P. Wireless communications under broadband reactive jamming attacks. *IEEE Trans. Dependable Secur. Comput.* **2015**, *13*, 394–408.
23. Ng, D.W.K.; Lo, E.S.; Schober, R. Robust beamforming for secure communication in systems with wireless information and power transfer. *IEEE Trans. Wirel. Commun.* **2014**,*13*, 4599-4615.
24. Chen, G.; Gong, Y.; Xiao, P.; Chambers, J.A. Physical layer network security in the full-duplex relay system. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 574–583.
25. Ouyang, N.; Jiang, X.Q.; Bai, E.; Wang, H.M. Destination assisted jamming and beamforming for improving the security of AF relay systems. *IEEE Access* **2017**, *5*, 4125–4131.
26. Zheng, G.; Krikidis, I.; Li, J.; Petropulu, A.P.; Ottersten, B. Improving physical layer secrecy using full-duplex jamming receivers. *IEEE Trans. Signal Process.* **2013**, *61*, 4962–4974.
27. Fritschek, R.; Schaefer, R.F.; Wunder, G. Deep learning for the Gaussian wiretap channel. In Proceedings of the IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.