electronics

MDPI

Article

Adversarial Perturbation Elimination with GAN Based Defense in Continuous-Variable Quantum Key Distribution Systems

Xun Tang [1], Pengzhi Yin [2], Zehao Zhou [3] and Duan Huang [4,*]

[1] School of Physics and Electronics, Central South University, Changsha 410083, China; xuntang@csu.edu.cn
[2] School of Automation, Central South University, Changsha 410083, China; pengzhiyin@csu.edu.cn
[3] School of Software, Xinjiang University, Urumqi 830001, China; zhouzehao@stu.xju.edu.cn
[4] School of Computer Science, Central South University, Changsha 410083, China
* Correspondence: duanhuang@csu.edu.cn

Abstract: Machine learning is being applied to continuous-variable quantum key distribution (CVQKD) systems as defense countermeasures for attack classification. However, recent studies have demonstrated that most of these detection networks are not immune to adversarial attacks. In this paper, we propose to implement typical adversarial attack strategies against the CVQKD system and introduce a generalized defense scheme. Adversarial attacks essentially generate data points located near decision boundaries that are linearized based on iterations of the classifier to lead to misclassification. Using the DeepFool attack as an example, we test it on four different CVQKD detection networks and demonstrate that an adversarial attack can fool most CVQKD detection networks. To solve this problem, we propose an improved adversarial perturbation elimination with a generative adversarial network (APE-GAN) scheme to generate samples with similar distribution to the original samples to defend against adversarial attacks. The results show that the proposed scheme can effectively defend against adversarial attacks including DeepFool and other adversarial attacks and significantly improve the security of communication systems.

Keywords: CVQKD; adversarial attack; DeepFool; APE-GAN

Citation: Tang, X.; Yin, P.; Zhou, Z.; Huang, D. Adversarial Perturbation Elimination with GAN Based Defense in Continuous-Variable Quantum Key Distribution Systems. *Electronics* **2023**, *12*, 2437. https://doi.org/10.3390/electronics12112437

Academic Editor: Aryya Gangopadhyay

Received: 13 April 2023
Revised: 21 May 2023
Accepted: 25 May 2023
Published: 27 May 2023

1. Introduction

Quantum key distribution (QKD) [1] is being applied worldwide, motivated by the innovation and development of quantum secure communication technologies, which allow two remote parties, typically referred to as Alice and Bob, to exchange secret keys in an untrustworthy setting without being overheard by an eavesdropper. The fundamental laws of quantum physics [2] guarantee that in the event of an eavesdropper, Eve, the permitted receiver, Bob, will be able to recognize Eve's illegal actions and erase the leaked information.

However, real continuous-variable quantum key distribution (CVQKD) [3–6] systems face security vulnerabilities due to some deviations between the theoretical assumptions and the implementation, which gives Eve the opportunity to compromise security by stealing information from legitimate parties. The eavesdroppers can employ wavelength attacks [7,8], calibration attacks [9,10], local-oscillation (LO) intensity attacks [11], homodyne-detector-blinding attacks [12], saturation attacks [13], and other attack strategies to compromise the safety of the GMCS CV-QKD. The major idea of these practical attack strategies is to use optical device flaws to deviate from the redundant noise estimates, while the essence of the corresponding typical countermeasures is to incorporate suitable real-time monitoring modules or measurement devices into the system, which depends considerably on the accuracy of the estimated excess noise. However, in practical experiments, we cannot predict what kind of attacks Eve will execute, so we need a generic defense scheme that can defend against as many attacks as possible.

Fortunately, the boom in machine learning has brought benefits to CVQKD defense, and numerous developments based on artificial neural networks (ANN) have been demonstrated to be successful. An ANN model for attack detection and classification was proposed by Mao et al. [14], which functions by extracting the feature vectors of most known attacks as input; Du et al. [15] proposed a multi-attack ANN detection model to handle the coexistence of multiple attacks; and a semi-supervised deep learning strategy was put forth by Luo et al. [16] to identify known assaults or potential threats. The key to the above approaches is to implement specific defensive measures according to the classification results of the ANN model. However, recent research has observed that ANN are vulnerable to well-designed input samples (called adversarial samples), which can easily fool a well-behaved deep learning model with only a little interference detected by humans. Even though adversarial attacks on CVQKD systems are uncommon, the vulnerability of the quantum classifier itself has recently attracted a lot of attention, creating the prerequisites for adversarial samples. According to the principle of adversarial attacks [17], the original input can be misclassified by specific minor perturbations on the input vector, which represents a significant security risk for this delicate system.

This study proposes an adversarial attack that can be applied to the QKD domain, the DeepFool attack [18], to directly counter the convolutional neural network (CNN) classification networks in AI-based CVQKD defense strategy. The schematic diagram of our attacked CVQKD system is shown in Figure 1. The DeepFool attack method, based on linear approximation, is adapted to the CVQKD attack detection network. The classification results of the perturbed input reveal that DeepFool can successfully move attacks from the image recognition domain to the CVQKD attack detection domain. To demonstrate that adversarial attacks are not limited to one forensic classifier, we built four representative CNNs in the CVQKD system that were trained to distinguish different attack classes which include three known attacks, one hybrid attack, and a normal state. In addition, we proposed improved methodologies for the APE-GAN [19] defense scheme to eliminate the adversarial perturbations added to the CVQKD system.

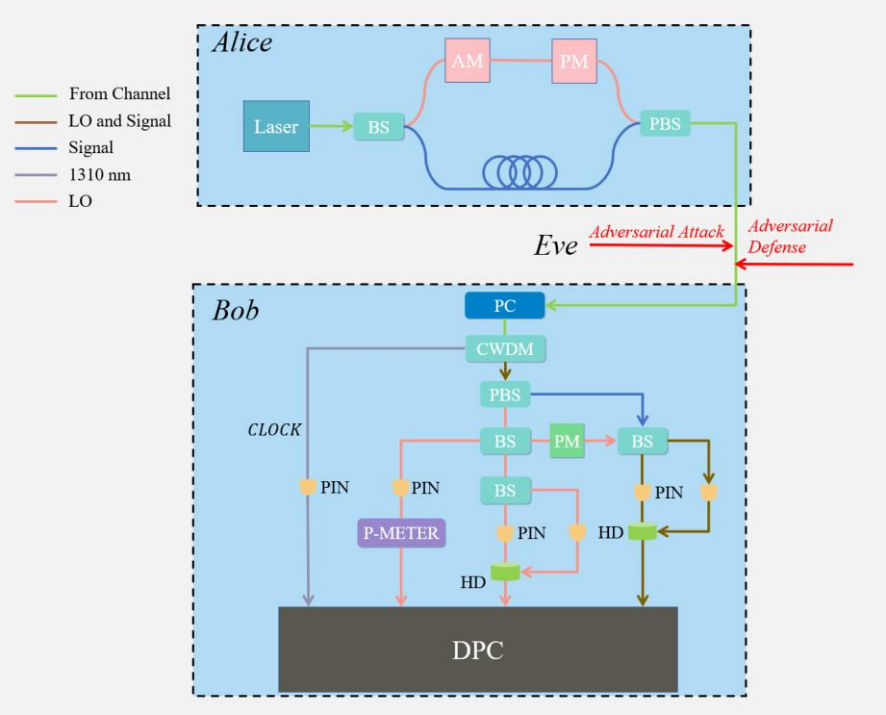

Figure 1. The schematic diagram for the attack detection portion of the CVQKD system considering adversarial attacks and defenses. The attack from Eve includes normal attacks and adversarial perturbations.

PC: polarization controller. CWDM: coarse wavelength-division multiplexing. PBS: polarization beam splitter. BS: beam splitter. AM: amplitude modulator. PM: phase modulator. PIN: PIN photodiode. HD: homodyne detector. P-METER: the power meter is to monitor LO intensity. DPC: data processing center used to sample analog signal, and attack detection.

The remainder of this paper is organized as follows. In Section 2, we present the preparation of the dataset and the adversarial attack strategy, as well as the APE-GAN defense method used, which includes details of the CNN classification model used, the DeepFool algorithm, and APE-GAN. The relevant experimental results and analysis are presented in Section 3. Section 4 serves as our discussion and conclusion.

2. Methodology

2.1. Preparation of the CVQKD System Dataset

The GG02 protocol, based on Gaussian modulation and balanced homodyne detection, is the closest practical CVQKD protocol available. According to the protocol, we used an easy to prepare coherent state as the light source and a relatively low-cost balanced homodyne detector on the detection side. The protocol was implemented as follows:

(a) Alice, as the sender, uses a true random number generator to produce two sets of random sequences $\{X_A\}$ and $\{P_A\}$ of length n, with zero mean, while obeying Gaussian distribution. Then, n coherent states $|X_A + iP_A\rangle$ are successively prepared from this random sequence and sent to Bob, the receiver, through the quantum channel.

(b) After arriving at the receiver, Bob randomly switches the measurement base $(0,\ \pi\ /2)$ for the received quantum state to perform the balanced homodyne measurement. Through this process, the measurement result can be obtained as X_B *or* P_B.

(c) Bob, at the receiver side, publishes the measurement base and part of the detection results through the classical channel; Alice filters the data according to the published measurement base and keeps the corresponding results X_A *or* P_A.

(d) Alice evaluates the security of this communication by estimating the channel parameters based on her own data and the data published by Bob.

(e) If the communication is secure, Alice and Bob perform data post-processing, including data coordination and private amplification. Finally, Alice and Bob will share the exact same set of security keys.

However, the experimental implementation of CVQKD can lead to various practical security issues due to the imperfect properties of the device and the complex environment in the experiment. Eve can exploit these imperfections to implement quantum hacking strategies, which in turn can mask classical entanglement cloning attacks or interception retransmission attacks to successfully steal key information. In our experiments, we mainly consider three typical attack strategies for the CVQKD system with homodyne detection, including the LO intensity attack, the calibration attack, the saturation attack, and a hybrid attack strategy that combines LO intensity attacks and wavelength attacks. In addition, since the individual wavelength attack is only applicable to the heterodyne detection CVQKD system, it was not considered here.

In the quantum channel, the LO signal is transmitted together with the quantum signal and multiplexed, while a classical strong LO signal may be controlled by Eve. In a practical CVQKD system, the LO signal can be used as a reference signal to compensate the phase and polarization of the quantum signal for stable coherent detection. Therefore, it is important to ensure the stability and safety of the LO signal. However, researchers have found that Eve can manipulate the local oscillation signal in the quantum channel to perform LO intensity attacks and calibration attacks on the system to successfully steal the key information. In addition, Eve may perform wavelength attacks, which can use other wavelengths to attack Bob and affect the balanced homodyne detection results.

In order to evaluate the impact of each attack on the security of the CVQKD system, the process of estimating each parameter is introduced next. In the CVQKD system, we

focus on estimating two parameters, one of which is the variance of Alice's and Bob's sites $(x^2$ and $y^2)$, and the other is the covariance in Alice and Bob (xy). We note that:

$$\left\langle x^2 \right\rangle = V_A, \ \langle xy \rangle = \sqrt{\eta T}\, V_A \tag{1}$$

$$\left\langle y^2 \right\rangle = \eta T V_A N_0 + N_0 + \eta T \xi \ + V_{el} \tag{2}$$

where T is the quantum channel transmittance and η is the efficiency of the homodyne detector. V_{el} is the electronic noise coefficient of the detector, ξ, is the technical excess noise of the system.

During Gaussian modulation CVQKD, two distributed random numbers $\equiv (q_{\mathrm{A}}, p_{\mathrm{A}})$ are taken from the two-dimensional Gaussian distribution of mean value 0 and variance V_{A} from Alice. Then, Alice prepares the coherent state $\left| \frac{q_A + i p_A}{2} \right\rangle$ and sends it to Bob through the quantum channel. Bob uses heterodyne detection to simultaneously measure the two orthogonal components of the received coherent state and obtains $Y_1 \equiv \left(q_{\mathrm{B}_1}, p_{\mathrm{B}_1}\right)$. Alice and Bob obtain the associated Gaussian variables as $(X_i, Y_{1,i})_{i=1\cdots N}$ after N transmission rounds. These associated Gaussian variables are used as the original keys. The $\rho_{\ \mathrm{A}'\ \mathrm{B}_1}$ state covariance matrix can be obtained by [20]:

$$V_{\ \mathrm{A}'\ \mathrm{B}_1} = \begin{pmatrix} VI & \sqrt{T} Z_{\mathrm{A}} \sigma_{\mathrm{z}} \\ \sqrt{T} Z_{\mathrm{A}} \sigma_{\mathrm{z}} & T(V + \chi) I \end{pmatrix} \tag{3}$$

After Bob implements heterodyne detection, he produces:

$$V_{\ \mathrm{A}'|Y_1} = \left(V - \frac{T\left(V^2 - 1\right)}{T(V + \chi) + 1} \right) I \tag{4}$$

Alice and Bob simultaneously use heterodyne detection to obtain the associated Gaussian variables $\left(X_i', Y_{1,i}\right)_{i=1\cdots \mathrm{N}}$. In the data post processing step, they randomly disclose a part of the original key $(X_i, Y_{1,i})_{i=1\cdots m}$ to estimate the quantum channel [21]:

$$\left\langle q_{\mathrm{A}}^2 \right\rangle = \left\langle p_{\mathrm{A}}^2 \right\rangle = V_{\mathrm{A}} \tag{5}$$

$$\left\langle q_{\mathrm{A}} q_{\mathrm{B}_1} \right\rangle = \left\langle p_{\mathrm{A}} p_{\mathrm{B}_1} \right\rangle = \sqrt{\frac{T}{2}} V_{\mathrm{A}} \tag{6}$$

$$\left\langle q_{\mathrm{B}_1}^2 \right\rangle = \left\langle p_{\mathrm{B}_1}^2 \right\rangle = \frac{T}{2}(V_{\mathrm{A}} + \xi) + 1 \tag{7}$$

which satisfy the following relationship:

$$Y_1 = \frac{t}{\sqrt{2}} X + Z \tag{8}$$

where, $t = \sqrt{T}$ and $Z \equiv (q_\delta, p_\delta)$. q_δ and p_δ meet the normal distribution, and the cumulative noise variance is $\sigma^2 = 1 + T\xi/2$, which can be applied to determine the signal-to-noise ratio and construct the covariance matrix $V_{\ \mathrm{A}'\mathrm{B}_1}$. The covariance matrix can be used to calculate the coding rate after the SNR is constructed. In the case of low SNR, they can use multidimensional negotiation. The specific multidimensional negotiation process is: suppose Alice and Bob divide the remaining original keys whose Gaussian vector are X and Y_1 into $2n/d$ d-dimensional Gaussian vectors X^d, Y_1^d, and normalize them to acquire $\|X^d\| = \sqrt{\langle X^d, X^d \rangle}$, $\|Y_1^d\| = \sqrt{\langle Y_1^d, Y_1^d \rangle}$, $X'^d = X^d / \|X^d\|$, $Y_1'^d = Y_1^d / \|Y_1^d\|$. Bob calculates it to obtain $v = M\left(Y_1'^d, u\right) X'^d$ on the d-dimensional hypercube, where

$u \in \left\{-1/\sqrt{d}, 1/\sqrt{d}\right\}^d$ is a uniformly distributed vector. After multi-dimensional negotiation, Alice and Bob have v and u respectively, and they can then apply channel coding technology to correct signals. According to the estimated signal-to-noise ratio, they can choose the multilateral low-density parity check code with the appropriate code rate. After the error correction is successful, they implement privacy amplification to extract the final key. In a practical CVQKD system, the impact of different attack strategies on measurable feature values is distinct. Therefore, we chose a few of these feature values, the shot noise variance N_0, the intensity I_{LO} of the LO, and the mean value $\overline{y}$ and variance V_y of Bob's measurement, which are more influenced by the attack, for the analysis. We first constructed the feature vector $\vec{u} = \{ I_{LO}, N_0, \overline{y}, V_y \}$ and then used CNNs to learn these variations of these features to help detect and classify different attacks. The values of the feature vectors corresponding to various attacks varied because different attacks have varying effects on various characteristics and alter their values in distinct ways. It is significant to highlight that while there are errors between the feature values of the full dataset and these values, neural networks can still distinguish between very similar attacks since the errors under different attacks are also distinct.

In the data pre-processing stage, we manipulated the data for different attacks, including normalization, vectorization, feature extraction, and sequential processing to make it more suitable for input to CNNs. The detailed steps are as follows. Firstly, for each CVQKD attack strategy, 1000 original sampling data with Gaussian distribution are generated for each attack type, including normal conditions. Then, a total of 5000 data are extracted to the output feature vector. The original data are input in time series, and every 25 pulses are grouped into groups. After that, we compute the four statistical features of the group and obtain the feature vector $\{ I_{LO}, N_0, \overline{y}, V_y \}$. To construct the rational data set, all data are arranged in a random fashion. Finally, the feature vectors are divided into a training set and a test set by a ratio of 2:1.

2.2. CNN Model Establishment for Attack Classification

It is simple for the classifier to make a misclassification by adding perturbation to the input photos, since deep neural networks are susceptible to adversarial samples. We therefore need some well-trained models as scoring functions in order to validate the attack effectiveness of the DeepFool method on CVQKD attack classification networks.

The DeepFool algorithm, which was first proposed in the field of images by Moosavi-Dezfooli et al., was found to perform very effectively against a variety of deep neural networks.

Adversarial samples are produced through numerous iterations or queries on the neural network, implying that the neural network serves a dual role: being targeted by adversarial sample attacks and also being the foundation for creating those adversarial samples. The neural networks that are commonly used in various applications include graph neural networks (GNN), convolution neural network (CNN), recurrent neural network (RNN), and deep neural network (DNN) [22–24]. In CVQKD systems, data is typically represented as a series of continuous variables derived from photon quantum states. This data representation is more natural as a grid or image-like structure, which CNN can effectively process with the ability to learn local spatial patterns and hierarchical features. However, GNN need to be discretized or converted into graphical structures, which can lead to loss or distortion of information [25].

Therefore, in our work, we chose CNN models for our experiments. We selected four models, the AlexNet [26], VGG-16 [27], ResNet [28], and DenseNet [29] networks, based on the generality of the attacked models. To better understand our CNN model establishment for attack classification in CVQKD systems, we introduce in detail a typical CNN model (AlexNet).

AlexNet deep neural network has been widely applied in the direction of image recognition, which is an important breakthrough in the field of computer vision in recent years. The brief structure of AlexNet is shown in Figure 2. In 2012, Alex et al. achieved

the best result in the ImageNet competition image classification task using AlexNet with convolutional neural networks, making convolutional neural networks a great success in image classification. AlexNet applies ReLU instead of Sigmiod as the activation function of CNN, and verifies that ReLU outperforms Sigmoid in deeper networks, effectively solving the problem of gradient dispersion caused by Sigmoid. We chose the AlexNet network model that we used for a brief introduction.

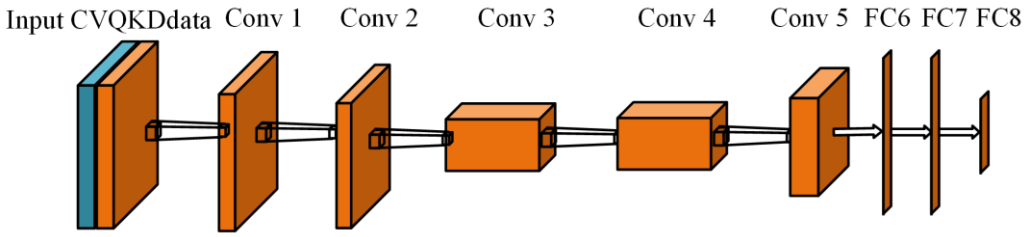

Figure 2. The brief structures of the AlexNet for CVQKD attack detection.

We simplified and modified the structure slightly, since the input matrix of the model is far simpler than the input for the picture information that was originally designed. We introduced three maximum pooling layers to reduce the size of the model and added a dropout layer after the fully connected layer. The dropout value has a certain influence on the training result of the network. The dropout value was changed from 0.4 to 0.6, and we found that when the dropout value is 0.5, the loss rate was 0.198 and the model was trained best. Hence, the dropout value was also set to 0.5 in this experiment.

The performance of the training model is shown in Table 1 and Figure 3. In actual experiments, the Resnet uses the Resnet-18 architecture. According to the experimental results, the test set can achieve a satisfactory accuracy of 95.31% on average. It is worth noting that the AlexNet classification accuracy in the CVQKD dataset is 92.39%, while in the original CIFAR-10 and ImageNet test datasets, the classification accuracy of each CNNs in the image domain is higher, with 97.09%, 98.26%, 99.28%, and 99.63%, respectively. For this result, we consider the possibility that the noise of the system input has a greater impact on the measured values, resulting in a smaller impact of the attack on the original data, which affects the recognition performance of the model. In addition, the four typical CNNs classification models used in this experiment were not originally designed for the CVQKD attack detection system. We discovered via more research that the feature vectors of the hybrid attack and the LO intensity are similar to the normal vectors and difficult to distinguish. Because of this, the hidden layer of AlexNet has less learning capacity when the number of neurons is minimal, which leads to poor classification accuracy. By improving and optimizing the network structure, the model classification accuracy can be further improved.

Table 1. Classification accuracy (in %) produced by the four different CNNs for the CVQKD system dataset.

Model	Accuracy
AlexNet	92.39
VGG	97.80
ResNet	95.12
DenseNet	95.93
Average	95.31

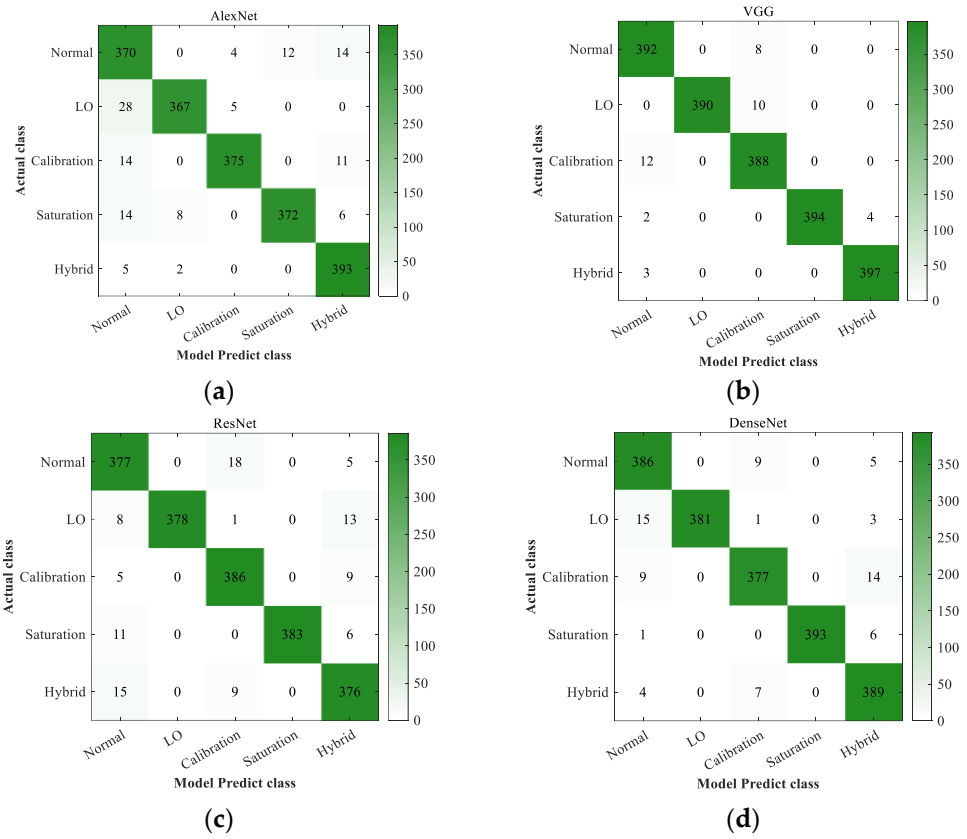

Figure 3. The confusion matrices of CNNs for CVQKD attack detection. (**a**) The AlexNet model classification results in five scenarios. (**b**) The VGG model classification results in five scenarios. (**c**) The ResNet model classification results in five scenarios. (**d**) The DenseNet model classification results in five scenarios.

2.3. DeepFool Attack for CVQKD

By adding a small perturbation to the original data through adversarial example generation techniques, the adversarial attack can produce the smallest perturbation needed to simply and effectively change the classification labels. In order to trick the target classifier into making incorrect predictions, the adversarial attack approach introduces imperceptibly small perturbations to the benign inputs. As shown in Table 2, the DeepFool attack is a strong attack strategy, which, when introduced into CVQKD data, can have a perturbation effect. It makes use of the geometry attribute to investigate the smallest perturbations needed to pass the target classifier's judgment boundary for a given input. In the following, we will explain the DeepFool attack algorithm in detail.

Table 2. Classification of adversarial attacks.

Method	White/ Black	Single/ Iterative	Constrain Norm	Weak/ Strong
DeepFool	White	Iterative	2	Strong
FGSM	White	Single	Infinity	Strong
BIM	White	Single	Infinity	Weak
UMIFGSM	White	Single	Infinity	Strong
CBIM	White	Iterative	Infinity	Weak
RP2	White	Iterative	Infinity, 2	Weak
UPSET	Black	Single	Infinity	Weak
UAP	Black	Iterative	Infinity, 2	Strong
C&W	Black	Iterative	Infinity, 0, 2	Weak

Typically, we define a minimal perturbation r necessary to change the initial classification label p as follows for a particular classifier:

$$\Delta(x; p)\min_{r}\| r \|_2 \tag{9}$$

$$\textit{subject to } p(x+r) \neq p(x) \tag{10}$$

where x and $p(x)$ are an image data and the estimated label, respectively; furthermore, we refer to $\Delta(x; p)$ as the robustness at point x. DeepFool introduces the idea of geometry to compute the minimal norm against perturbations for a given image. In each iteration, they add a small vector while assuming that the original data is situated within the decision boundary's designated area. The final perturbations are obtained by accumulating these vectors up until the picture labels change. We assume $p(x) = sign(f(x))$, where $f{:}R^n \rightarrow$R is an arbitrary scalar value image classification function. The classification is carried out using the equation below:

$$\hat{k}(x) = \arg\max_{k} f_k(x) \tag{11}$$

where $f_k(x)$ is the output of $f(x)$ that corresponds to the kth class. Similarly, we assume that $f(x) = W^T x + b$ for a given W and b is a classifier that solves a multi-classification task. The smallest perturbation that can fool the classifier is as follows:

$$\arg\min_{r} \|r\|_2 \tag{12}$$

$$\text{s.t. } \exists k : \omega_k^T(x_0 + r) + b_k \geq \omega_{\hat{k}(x_0)}^T(x_0 + r) + b_{\hat{k}(x_0)} \tag{13}$$

In this algorithm, the classification limitation of the closest data point is calculated as follows:

$$\hat{p}(x_0) = \arg\min_{k \neq \hat{k}(x_0)} \frac{\left| f_k(x_0) - f_{\hat{k}(x_0)}(x_0) \right|}{\left\| \left| \omega_k - \omega_{\hat{k}(x_0)} \right| \right\|_2} \tag{14}$$

The value of the adversarial perturbation $r_*(x_0)$ is defined as the distance from the data point x_0 to the classification boundary $\hat{p}(x_0)$.

$$r_*(x_0) = \frac{\left| f_k(x_0) - f_{\hat{k}(x_0)}(x_0) \right|}{\| \omega_{\hat{l}(x_0)} - \omega_{\hat{k}(x_0)} \|_2^2} \left(\omega_{\hat{l}(x_0)} - \omega_{\hat{k}(x_0)} \right) \tag{15}$$

The above steps are the fundamental method used in the DeepFool attack. According to this method, we modified the input dataset for CVQKD attack detection. First, we added a normalization process to the input data to solve the problem that each metric of the CVQKD attack is at different orders of magnitude. Second, to accommodate the attack algorithm in the image domain and also to enhance the stability of the input, we created an input matrix X from 4 consecutive feature vectors $\vec{u}$, which can be viewed as a 4×4 data matrix. Therefore, the input layer of the AlexNet should be a series of 4×4 vector matrices. On this basis, we imposed the DeepFool attack algorithm on the input data to obtain the perturbed adversarial samples $\hat{X}$. At this point, the fundamental modification to migrate the DeepFool attacks from the image domain to CVQKD attack detection was completed. Through this method, we could implement adding a small perturbation to each input matrix to achieve the purpose of fooling the CVQKD attack detection network.

2.4. APE-GAN Based Defense Strategy

Considering that the misclassification of adversarial samples is mainly caused by some intentional imperceptible perturbations at the pixel level of the input data, we expect to use an algorithm to eliminate the adversarial perturbations of the input data for the purpose of

defending against adversarial attacks. The generative adversarial network (GAN) proposed by Goodfellow et al. [30] can solve this problem well by generating a dataset similar to the original distribution using random noise.

GAN reconstructs data distributions similar to the original clean samples by using adversarial samples. The generator (G) generates new samples by fitting the data generation process, and the discriminator (D) discriminates whether the input samples are the original data distribution. The two learn the transformation of the distribution from some simple input distribution to the image space through mutual gambling.

In our experiments, we utilize adversarial perturbation elimination with GAN (APE-GAN), an improved GAN defense model, to eliminate the adversarial perturbations added to the CVQKD system. Some details of the algorithm are as follows: a small enough perturbation ε of the input X_ε and X that satisfies the following condition can be used to generate an adversarial instance,

$$||X_\varepsilon - X|| = \varepsilon \tag{16}$$

where X_ε is the adversarial example, X is the clean example, and f is the mapping of the classifier from the input images to the discrete set of labels. The defense against adversarial examples is achieved by eliminating or destroying the trivial perturbations of the input X_ε before it is recognized by the target model, which is the basic idea of the defense method.

As shown in Figure 4, in a practical adversarial environment, we train the generator APE-G to apply small changes to the input adversarial samples, while continuously optimizing the discriminator APE-D for separating clean and reconstructed samples, thus achieving the goal of eliminating adversarial perturbations. The final target of APE-GAN is to train a generative model G, which is able to generate the corresponding reconstructed samples for the input adversarial samples with the same distribution as the original samples. The generative network G can be parameterized by θ.

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{k-1}^{N} l_{ape}\left(G_\theta\left(X_\varepsilon^k\right), X^k\right) \tag{17}$$

where l_{ape} is the adversarial perturbation elimination-specified loss function that is used to optimize the weights and basis points of the generative network. X^k corresponds to the original clean image data. X_ε^k is the perturbed image data obtained using the DeepFool algorithm. The optimization objective of APE-GAN can also be formulated as follows:

$$\min_G \max_D V(G,D) = E_{X\sim P_{data}(X)}[logD(X)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{18}$$

The formulation implies training a generative model, APE-G, with the purpose of fooling a distinguishable discriminator, APE-D, that is trained to distinguish the adversarial perturbation-removed data. The well-trained generator is able to produce reconstructed data with a high degree of similarity to the original clean data, which APE-D cannot distinguish. Therefore, by training the generator, the method is able to eliminate the adversarial perturbations in the original input data.

However, the defense effect of APE-GAN on small data sets is relatively poor, in order to improve the robustness of the model. In the process of APE-GAN training, adversary samples are also added, and the iterative method is used for adversary training. The specific objective function is proposed as follows:

$$Loss = \frac{1}{(N-b) + \lambda b}\left(\sum_{x\in X} L(x \mid y) + \lambda \sum_{x^*\in X_\varepsilon} L(x^* \mid y)\right) \tag{19}$$

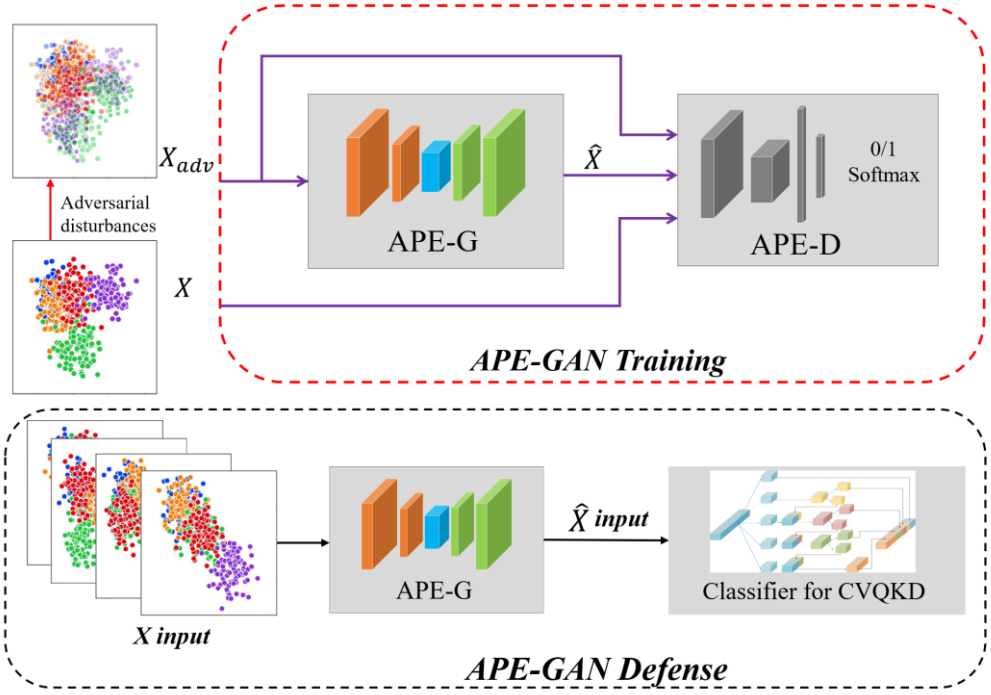

Figure 4. The frame of the APE-GAN based defense strategy. X: the original clean data. X_{adv}: Perturbed data after the adversarial attack. $\hat{X}$: the reconstructed data.

This can control the number of normal samples and counter samples in each small batch of training. b represents the number of counter samples, $L(x|y)$ is the loss function of normal sample x to the real mark y, and $L(x^*|y)$ represents the loss of counter samples, λ is used to control the weight of the counter sample. To solve the label leaking effect in APE-GAN and improve the versatility of it, we do not directly use the real sample label when constructing the confrontation sample, but replace it with the most unlikely category label; we mark it as:

$$y_{LL} = \underset{i \in [0, C-1]}{\operatorname{argmin}} O^i(x) \tag{20}$$

Then, the target of the corresponding objective function is to make the prediction label of the model for the antagonistic sample close to the most unlikely label:

$$x^* = D(x - \varepsilon \cdot \operatorname{sign}(\nabla_x G(O(x), y_{LL})) \tag{21}$$

The general architecture of our generator network APE-G is shown in Figure 4. In our generator network APE-G, some convolutional layers and some deconvolutional layers with stride = 2 are used to obtain low resolution feature maps and to recover the original resolution. Additionally, we trained a discriminator network APE-D with the general architecture depicted in Figure 4 in order to distinguish between the original clean data X and the rebuilt data $\hat{X}$, as shown in Algorithm 1. It also includes two dense layers, two convolutional layers with stride = 2, and a final sigmoid activation function to produce high-level feature maps and a probability for classifying the samples.

Algorithm 1 Improved APE-GAN Algorithm

Input: input data X, perturbation ε, number of data N, attack type D, parameter θ, λ
Initialization: θ, λ and ε
For $i \leftarrow 0$ to N do
$(x_i, y_i) \sim X$ // Sampling from normal distribution
$x_i^* \leftarrow \mathrm{G}(O(x_i), y_{LL}, \varepsilon)$ // Construct adversarial samples through (14) and (15)
Get the adversarial loss function through (13)
$\theta \leftarrow \theta - \eta \cdot \nabla_\theta(\mathrm{Loss}(\theta))$ // Update network parameters
Get $\hat{\theta}$ of the generative network G through (11)
Train APE-GAN through (12)
End for
Output: generator D

3. Implementations and Results

3.1. Experimental Environment

Our experimental environment is a Windows 10 64-bit system with PyCharm 2020.3.3 64-bit with an Intel(R) Core(TM) i5-10200 H CPU@2.40 GHz processor. The equipment comes from Lenovo Beijing Co., Ltd. in Beijing, China. The operating environment is TensorFlow + Python3.7.3.

3.2. Attack Results

According to the data preparation process described above, five data sets $Y = \left\{ y_{normal}, y_{LOIA}, y_{calib}, y_{sat}, y_{hyb} \right\}$ were generated, where five types of status are mixed in the same proportion. After that, we randomly selected 1000 sets of data, and we performed four DeepFool attacks on each input, so that we could achieve 4000 attacks for each model.

By placing the clean sample data and the attacked sample data into each of the four already trained networks for classification, the corresponding prediction results can be obtained. Then, the results were compared with the classification labels to obtain a confusion matrix plot for each classification network. This process is shown in Figure 5.

The classification error rates can be calculated from the confusion matrix as shown in Table 3. which mainly hovers around 89%, for AlexNet 93.86%, VGG 88.55%, ResNet 82.50%, and DenseNet 90.75%. Due to the different structures and depths of each network, there are discrepancies in the recognition accuracy of different classification networks after applying the DeepFool attack. Overall, there is a significant decrease in accuracy compared to the original classification network in Table 1. This indicates that our DeepFool attack algorithm successfully spoofed all four representative CNNs for CVQKD attack detection. Compared with the one pixel attack and multi pixel attacks already implemented in the CVQKD system, the DeepFool attack has a better effect because it has a stronger disturbance on the dataset and can bring more serious security issues.

Table 3. Classification error rates (in %) produced by the four different CNNs for the CVQKD system dataset.

Input	AlexNet	VGG	ResNet	Dense-Net	Avg.
Clean sample	7.61	2.20	4.88	4.07	4.69
Under Attack	93.86	88.55	82.50	90.75	88.93

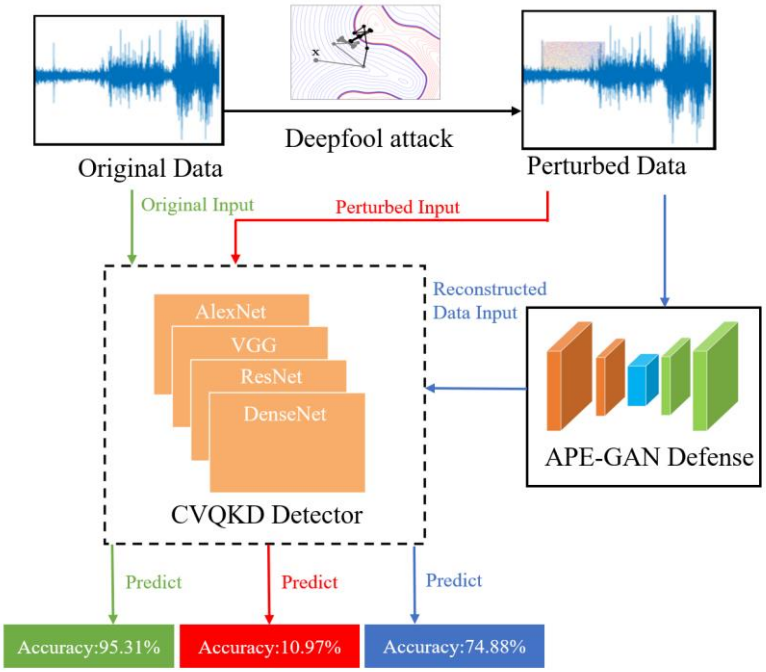

Figure 5. Schematic diagram of DeepFool attack and APE-GAN defense, where our chosen CNN classification models are AlexNet, VGG, ResNet, and DenseNet. The classification accuracy of the original data is 95.31% on average. Additionally, the average classification accuracy of perturbed data is 10.97% on average. The reconstructed data input achieves an accuracy of 74.88%.

3.3. Performance of APE-GAN Based Defense

As we found above, the DeepFool attack strategy applied to deceptive image classification can be used to fool the CVQKD attack detection network and can have a high misclassification effect. To defend against this adversarial attack, we put the adversarial samples and the original samples through the trained generator APE-G first, and then input the four target classifiers for classification after the adversarial perturbations were eliminated.

In the experiment, we adopted the capsule structure and parameters of the defense network as the feature extractor. According to the experimental results, APE-G can successfully resist adversarial examples generated using the DeepFool attack strategy. With a 6.14% average classification accuracy for adversarial examples, the trained AlexNet is easily misled. The VGG and ResNet are more robust against adversarial examples but are still vulnerable. It is interesting to note that the APE-G model can improve the recognition accuracy from 10.97% to 74.88%. Especially for the ResNet model, the classification accuracy can reach 84.63%. According to the experimental data we obtained, the classification error rate of each network was significantly reduced only after the perturbed data were passed through the APE-GAN to remove the adversarial perturbation. In addition, we investigated the effect of APE-GAN on clean samples and random noise as shown in Table 4.

Table 4. Reconstruction network settings based on GAN.

Model	Layer Name	Configuration
Defense model generator based on GAN	Fully connected	Number of neural: 1024 Activation: LeakyRelu
	Fully connected	Number of neural: 4096 Activation: LeakyRelu
	Deconv1~3	Filter: 4 × 4 Strides = 2 ×2 Number of Filters: 128 Padding = same Activation: LeakyRelu
	Conv	Filter: 3 × 3 Number of Fliters: 3 Padding = same Activation: tanh
Defense model discriminator based on GAN	Conv1	Filter:: 3 × 3 Strides = 2 × 2 Number of Fliters: 64 Padding = same Activation: LeakyRelu
	Conv2~3	Filter: 3 Strides = 2 × 2 Number of Fliters: 128 Padding = same Activation: LeakyRelu
	Conv4	Filter: 3 × 3 Strides = 2 × 2 Number of Fliters: 256 Padding = same Activation: LeakyRelu
	Fully Connected	Activation: sigmoid

The prediction accuracy of the adversarial cases processed by the APE-G model and input to the target model is shown in the "With Attacks and Defense" column in Table 5. The experimental results indicate that for clean sample data, there is no significant increase in the classification error rate after passing APE-G. At the same time, APE-G also has some robustness to random noise, as shown in Table 6. Also, there is a certain decrease in the classification error rate for random noise through APE-G. It can be concluded that the use of the APE-GAN defense method has less effect on the sample data that are not under adversarial attack, while the data that are under adversarial attack can play a role in eliminating the perturbation, and also can be robust to Gaussian noise in the CVQKD system.

Table 5. Classification accuracy of the sample data in the four models under three different scenarios.

Model	Without Attacks and Defense	With Attacks	With Attacks and Defense
AlexNet	92.39%	6.14%	63.29%
VGG	97.80%	11.45%	71.29%
ResNet	95.12%	17.05%	84.63%
DenseNet	95.93%	9.25%	80.32%
Average	95.31%	10.97%	74.88%

Table 6. Classification error rates (in %) for clean and random gaussian noise samples after APE-GAN defense in four different CNN classification models.

Input	AlexNet		VGG		ResNet		DenseNet	
	Only	After APE-G	Only	After APE-G	Only	After APE-G	Only	After APE-G
Clean sample	7.61	8.23	2.20	2.93	1.88	2.01	4.07	4.12
Random Gaussian noise sample	11.32	11.14	8.82	8.77	9.32	9.30	10.92	10.87

3.4. Strong Applicability of APE-GAN Based Defense

The APE-GAN defense scheme is effective in a wide range of adversarial attack strategies, which is far more than just the DeepFool attacks. We applied FGSM [31], CBIM, and JSMA [32] attacks of different attack strengths to the CVQKD system in the same way, and compared the classification accuracy before and after adopting the defense mechanism. From Table 7, we can conclude that the defense method we proposed has a good effect on a variety of adversarial attacks.

Table 7. Classification accuracy of attack detection networks in CVQKD against different attacks based on APE-GAN defense.

Adversarial Attack	Attack Strength	No Attack and Defense	Attack but No Defense	With Attack and Defense
FGSM	0.1	95.31%	39.42%	91.53%
	0.2		29.16%	85.34%
	0.3		20.52%	78.65%
CBIM	0.1	95.31%	25.43%	82.64%
	0.2		20.96%	72.61%
	0.3		14.38%	60.87%
JSMA	0.1	95.31%	20.14%	80.27%
	0.2		13.65%	72.56%
	0.3		3.54%	59.67%

4. Conclusions

In this paper, we proposed to apply the adversarial attacks algorithm used in the image domain to perturb the attack detection of CVQKD systems. In a simulated CVQKD system, we carried out the necessary experimental demonstration, where the DeepFool attack strategy led to an error probability of 88.93% on average for the four typical classification models and had the highest success rate of 93.86% in AlexNet. The above results fully demonstrate the destructiveness of adversarial attacks on CNNs in CVQKD systems and the vulnerability of quantum systems to adversarial samples. Furthermore, to address the adversarial attacks present in the CVQKD system, we proposed to train the improved generative network using the APE-GAN method to generate samples that are similar to the original clean sample distribution. This defense scheme has a significant effect of eliminating adversarial perturbations from the sample. The simulation results show that the trained APE-G network can significantly improve the accuracy of the CVQKD attack detection networks in identifying and classifying attacks, and it improves the mean accuracy from 10.97% to 74.88% when subjected to adversarial attacks. Additionally, APE-GAN has no significant increase in error rate in clean images, and also has some robustness to random noise. The results show that the proposed scheme can effectively eliminate adversarial attacks and significantly improve the security of the communication system.

Author Contributions: Conceptualization, X.T.; methodology, X.T. and P.Y.; resources, D.H.; software, X.T., P.Y. and Z.Z.; validation, X.T., P.Y., Z.Z. and D.H.; data curation, X.T.; Funding acquisition, P.Y., D.H.; writing—original draft preparation, X.T.; writing—review and editing, X.T., P.Y., Z.Z. and D.H.; visualization, X.T., P.Y. and Z.Z.; supervision, D.H. All authors have read and agreed to the published version of the manuscript.

Funding: National College Innovation Project (2022105330245).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors express appreciation to Y. Guo, Y. Yan, Y. Mao, H. Luo for their pioneering research. Furthermore, we thank the reviewers of this work for their valuable comments and suggestions.

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

References

1. Scarani, V.; Bechmann-Pasquinucci, H.; Cerf, N.J.; Dušek, M.; Lütkenhaus, N.; Peev, M. The security of practical quantum key distribution. *Rev. Mod. Phys.* **2009**, *81*, 1301–1350. [CrossRef]
2. Xu, F.; Ma, X.; Zhang, Q.; Lo, H.; Pan, J. Secure quantum key distribution with realistic devices. *Rev. Mod. Phys.* **2020**, *92*, 025002. [CrossRef]
3. Huang, D.; Huang, P.; Lin, D.; Zeng, G. Long-distance continuous-variable quantum key distribution by controlling excess noise. *Sci. Rep.* **2016**, *6*, 19201. [CrossRef] [PubMed]
4. Guo, Y.; Peng, Q.; Liao, Q.; Wang, Y. Trans-Media Continuous-Variable Quantum Key Distribution via Untrusted Entanglement Source. *IEEE Photonics J.* **2013**, *13*, 1–12. [CrossRef]
5. Kundu, N.K.; Dash, S.P.; Mckay, M.R.; Mallik, R.K. Channel Estimation and Secret Key Rate Analysis of MIMO Terahertz Quantum Key Distribution. *IEEE Trans. Commun.* **2022**, *70*, 3350–3363. [CrossRef]
6. Cao, Y.; Zhao, Y.; Li, J.; Lin, R.; Zhang, J.; Chen, J. Hybrid Trusted/Untrusted Relay-Based Quantum Key Distribution Over Optical Backbone Networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2701–2718. [CrossRef]
7. Li, Y.; Wang, X.; Bai, Z.; Liu, W.; Yang, S.; Peng, K. Continuous variable quantum key distribution. *Chin. Phys. B* **2017**, *26*, 040303. [CrossRef]
8. Huang, J.-Z.; Weedbrook, C.; Yin, Z.-Q.; Wang, S.; Li, H.-W.; Chen, W.; Guo, G.-C.; Han, Z.-F. Quantum hacking of a continuous-variable quantum-key-distribution system using a wavelength attack. *Phys. Rev. A* **2013**, *87*, 062329. [CrossRef]
9. Jouguet, P.; Kunz-Jacques, S.; Diamanti, E. Preventing calibration attacks on the local oscillator in continuous-variable quantum key distribution. *Phys. Rev. A* **2013**, *87*, 062313. [CrossRef]
10. Tang, Z.; Liao, Z.; Xu, F.; Qi, B. Experimental Demonstration of Polarization Encoding Measurement-Device-Independent Quantum Key Distribution. *Phys. Rev. Lett.* **2014**, *112*, 190503. [CrossRef]
11. Ma, X.; Sun, S.; Jiang, M.; Liang, L. Local oscillator fluctuation opens a loophole for Eve in practical continuous-variable quantum-key-distribution systems. *Phys. Rev. A* **2013**, *88*, 022339. [CrossRef]
12. Guo, Y.; Yin, P.; Huang, D. One-Pixel Attack for Continuous-Variable Quantum Key Distribution Systems. *Photonics* **2023**, *10*, 129. [CrossRef]
13. Qin, H.; Kumar, R.; Alléaume, R. Quantum hacking: Saturation attack on practical continuous-variable quantum key distribution. *Phys. Rev. A* **2016**, *94*, 012325. [CrossRef]
14. Mao, Y.; Huang, W.; Zhong, H.; Wang, Y.; Qin, H.; Guo, Y.; Huang, D. Detecting quantum attacks: A machine learning based defense strategy for practical continuous-variable quantum key distribution. *New J. Phys.* **2020**, *22*, 083073. [CrossRef]
15. Du, H.; Huang, D. Multi-Attack Detection: General Defense Strategy Based on Neural Networks for CV-QKD. *Photonics* **2022**, *9*, 177. [CrossRef]
16. Luo, H.; Zhang, L.; Qin, H.; Sun, S.; Huang, P.; Wang, Y.; Wu, Z.; Guo, Y.; Huang, D. Beyond universal attack detection for continuous-variable quantum key distribution via deep learning. *Phys. Rev. A* **2022**, *105*, 042411. [CrossRef]
17. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef]
18. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
19. Jin, G.; Shen, S.; Zhang, D.; Dai, F.; Zhang, Y. APE-GAN: Adversarial Perturbation Elimination with GAN. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019.
20. Desjacques, V.; Jeong, D.; Schmidt, F. Large-scale galaxy bias. *Phys. Rep.* **2018**, *733*, 1–193.
21. Michał, C.; Paul, F.; Alexander, M. Total Top-Quark Pair-Production Cross Section at Hadron Colliders Through O(4S). *Phys. Rev. Lett.* **2013**, *110*, 252004.
22. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
23. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef]
24. Mittal, S. A survey on modeling and improving reliability of DNN algorithms and accelerators. *J. Syst. Archit.* **2020**, *104*, 101689. [CrossRef]
25. Choong, J.J.; Liu, X.; Murata, T. Optimizing variational graph autoencoder for community detection with dual optimization. *Entropy* **2020**, *22*, 197. [CrossRef] [PubMed]
26. Al Badawi, A.; Jin, C.; Lin, J.; Mun, C.F.; Jie, S.J.; Tan, B.H.M.; Nan, X.; Aung, K.M.M.; Chandrasekhar, V.R. Towards the AlexNet Moment for Homomorphic Encryption: HCNN, the First Homomorphic CNN on Encrypted Data With GPUs. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1330–1343. [CrossRef]
27. Wang, S.-H.; Fernandes, S.L.; Zhu, Z.; Zhang, Y.-D. AVNC: Attention-Based VGG-Style Network for COVID-19 Diagnosis by CBAM. *IEEE Sens. J.* **2022**, *22*, 17431–17438. [CrossRef] [PubMed]

28. Liu, P.; Zhang, C.; Qi, H.; Wang, G.; Zheng, H. Multi-Attention DenseNet: A Scattering Medium Imaging Optimization Framework for Visual Data Pre-Processing of Autonomous Driving Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25396–25407. [CrossRef]
29. Joshi, S.; Villalba, J.; Żelasko, P.; Moro-Velázquez, L.; Dehak, N. Study of Pre-Processing Defenses Against Adversarial Attacks on State-of-the-Art Speaker Recognition Systems. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4811–4826. [CrossRef]
30. Goodfellow, I.; Pouget-Abadie, J. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
31. Rao, K.; Chowdary, C. CBIM: Community-based influence maximization in multilayer networks. *Inf. Sci.* **2022**, *609*, 578–594.
32. Cai, F.; Qiu, L.; Kuai, X.; Zhao, H. CBIM-RSRW: An Community-Based Method for Influence Maximization in Social Network. *IEEE Access* **2019**, *7*, 152115–152125. [CrossRef]