

General multi-attack detection for continuous-variable quantum key distribution with local local oscillator

Zhuo Kang(康茁), Wei-Qi Liu(刘维琪), Jin Qi(齐锦), and Chen He(贺晨)

Citation: Chin. Phys. B, 2024, 33 (**5**): 050308. DOI: 10.1088/1674-1056/ad2bf2
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Analysis of atmospheric effects on the continuous variable quantum key distribution

Tao Liu(刘涛), Shuo Zhao(赵硕), Ivan B. Djordjevic, Shuyu Liu(刘舒宇), Sijia Wang(王思佳), Tong Wu(吴彤), Bin Li(李斌), Pingping Wang(王平平), and Rongxiang Zhang(张荣香)
Chin. Phys. B, 2022, 31 (**11**): 110303. DOI: 10.1088/1674-1056/ac6b1f

Practical security analysis of continuous-variable quantum key distribution with an unbalanced heterodyne detector

Lingzhi Kong(孔令志), Weiqi Liu(刘维琪), Fan Jing(荆凡), and Chen He(贺晨)
Chin. Phys. B, 2022, 31 (**7**): 070303. DOI: 10.1088/1674-1056/ac4102

Short-wave infrared continuous-variable quantum key distribution over satellite-to-submarine channels

Qingquan Peng(彭清泉), Qin Liao(廖骎), Hai Zhong(钟海), Junkai Hu(胡峻凯), and Ying Guo(郭迎)
Chin. Phys. B, 2022, 31 (**6**): 060306. DOI: 10.1088/1674-1056/ac490a

Parameter estimation of continuous variable quantum key distribution system via artificial neural networks

Hao Luo(罗浩), Yi-Jun Wang(王一军), Wei Ye(叶炜), Hai Zhong(钟海), Yi-Yu Mao(毛宜钰), and Ying Guo(郭迎)
Chin. Phys. B, 2022, 31 (**2**): 020306. DOI: 10.1088/1674-1056/ac2807

Continuous-variable quantum key distribution based on photon addition operation

Xiao-Ting Chen(陈小婷), Lu-Ping Zhang(张露萍), Shou-Kang Chang(常守康), Huan Zhang(张欢), and Li-Yun Hu(胡利云)
Chin. Phys. B, 2021, 30 (**6**): 060304. DOI: 10.1088/1674-1056/abd931

General multi-attack detection for continuous-variable quantum key distribution with local local oscillator

Zhuo Kang(康茁), Wei-Qi Liu(刘维琪)†, Jin Qi(齐锦), and Chen He(贺晨)

School of Information Science and Technology, Northwest University, Xi'an 710127, *China*

(Received 13 December 2023; revised manuscript received 1 February 2024; accepted manuscript online 22 February 2024)

Continuous-variable quantum key distribution with a local local oscillator (LLO CVQKD) has been extensively researched due to its simplicity and security. For practical security of an LLO CVQKD system, there are two main attack modes referred to as reference pulse attack and polarization attack presently. However, there is currently no general defense strategy against such attacks, and the security of the system needs further investigation. Here, we employ a deep learning framework called generative adversarial networks (GANs) to detect both attacks. We first analyze the data in different cases, derive a feature vector as input to a GAN model, and then show the training and testing process of the GAN model for attack classification. The proposed model has two parts, a discriminator and a generator, both of which employ a convolutional neural network (CNN) to improve accuracy. Simulation results show that the proposed scheme can detect and classify attacks without reducing the secret key rate and the maximum transmission distance. It only establishes a detection model by monitoring features of the pulse without adding additional devices.

Keywords: CVQKD, generative adversarial network, attack classification

PACS: 03.67.Hk, 03.67.Dd, 03.67.–a **DOI:** 10.1088/1674-1056/ad2bf2

1. Introduction

Continuous-variable quantum key distribution (CVQKD) uses standardized optical communication devices, enabling the sender Alice and the receiver Bob to exchange key information in an insecure quantum channel.[1] Its theoretical security is guaranteed by the Heisenberg uncertainty principle[2] and quantum non-cloning principle.[3] The Gaussian modulated coherent state (GMCS) protocol is a popular CVQKD scheme, which has been proven to be safe against collective attack in theory[4–6] and can be easily implemented in practical experiment.[7–9] In the system, Alice needs to transmit a beam of local oscillator (LO) following the quantum signal, mainly acting as the reference pulse for homodyne detection, so LO will be much brighter than the quantum signal and inevitably induce the more photon-leakage noise.[10] Meanwhile, Eavesdropper (Eve) might use LO to launch attacks on the practical system, such as the LO fluctuation attack[11] and the calibration attack.[12]

Focused on the practical security problem caused by LO, CVQKD with a local local oscillator (LLO CVQKD) scheme was proposed.[13–16] In this scheme, the LO is generated locally by Bob and does not need to be transmitted in the channel, which can prevent all security problems caused by the loophole of the LO. This scheme can be achieved by using two independent narrow linewidth lasers with the same central wavelength, one generates the signal at Alice's side and the other generates LO at Bob's side. To achieve the phase reference between the two free-running lasers, Alice and Bob need to share the low-intensity phase reference pulses. Moreover, there has been extensive research on improving the performance of the LLO scheme.[17–19] In particular, when part of the phase estimation error noise is treated as trusted noise and the phase drift noise as untrusted noise, the scheme will achieve more excellent performance.[20–22]

Though Alice generates a relatively weak reference pulse as a phase reference between the signal and the LO, it still becomes a vulnerability for Eve to launch an attack. At present, there are several attack schemes,[22–26] two typical ones are reference pulse attack[22] and polarization attack.[23] The solution to these attacks is to add a suitable real-time monitoring module to the system, but due to the inevitable fluctuations in the reference pulse and electronic devices, the accuracy of the monitoring module cannot be well guaranteed. Moreover, additional optical devices and monitoring systems will inevitably affect the performance of the system. Therefore, we need a low-system-impact and general-purpose identification scheme to detect attacks.

In this paper, to improve the practical security of LLO CVQKD, we propose a general defense strategy via deep learning that can be monitored in real-time without adding additional optics considering the development of machine learning opens a door for QKD.[27–35] We investigate the typical characteristics of pulses affected by attacks and the deviations in the changes in these characteristics after being subjected to different attacks. After analysis, several features are selected to construct feature vectors for training generative adversarial networks (GANs). We add an automatic preprocessing module, and the collected features can be automatically turned into

†Corresponding author. E-mail: vickylwq1991@nwu.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

appropriate data. Bob can use this module to detect abnormal data in real time automatically. The experimental results show that the model can classify and identify attacks well, and achieve an accuracy rate of 99.8% over a transmission distance of 20 kilometers.

This paper is organized as follows: In Section 2, we briefly introduce the LLO CVQKD scheme and two typical attack schemes. In Section 3, we discuss the data processing, feature extraction and the training process of the GAN model. In Section 4, we analyze the performance of our proposed model. Finally, the conclusion is given in Section 5.

2. Description of the LLO CVQKD system and two attack models

2.1. Description of the LLO CVQKD system

The schematic diagram of the LLO CVQKD system in our experiment is shown in Fig. 1. The advantage of this scheme is that LO is generated locally, which eliminates the possibility of attacks caused by the LO pulse during transmission. At Alice's side, the CW laser is split into signal and reference pulses by a beam splitter (BS). The signal is continuously modulated with a Gaussian distribution centered at 0 and $V_A N_0$ as the variance. Here N_0 is the shot noise variance. After Gaussian modulation, the signal is transmitted to Bob through a quantum channel, which leads to an arbitrary phase rotation of the state in space. In order to compensate for the phase shift of the signal, the signal is sent out together with a delayed reference pulse. At Bob's side, a locally generated LO pulse is used to perform heterodyne detection of the signal and reference pulse. Due to the fact that the signal and reference pulse are generated using the same laser and transmitted on the same channel, Bob can estimate the phase rotation of the quantum signal between Alice and Bob by measuring the two quadratures of the reference pulse. Finally, the data received by Bob is input into the preprocessing module of the attack detection model after measured by the heterodyne detector. Then we extract a set of feature vectors from the processed data and put the processed data into GAN for attack detection and classification. It is worth noting that the preprocessing module comes into play after Bob received the data, and there is no need to add additional devices to the system to do the parameter prediction.

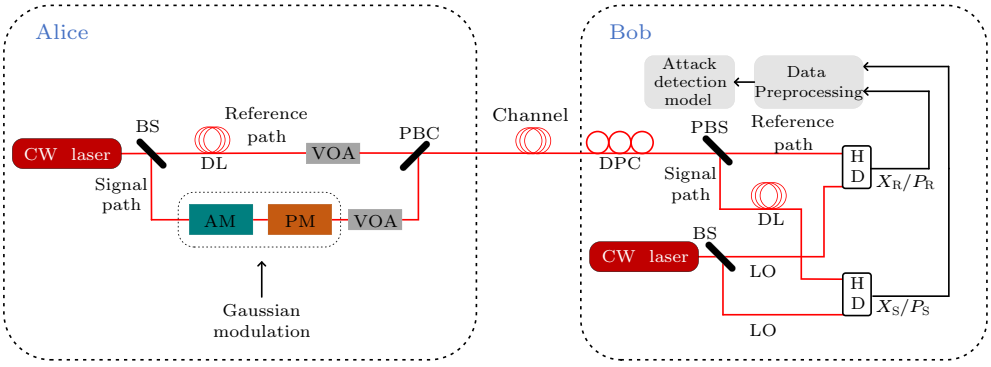


Fig. 1. Setup of the LLO CVQKD scheme. CW laser: continuous-wave laser; AM: amplitude modulator; PM: phase modulator; BS: beams splitter; DL: delay line; VOA: variable optical attenuator; PBC: polarizing beam collector; PBS: polarizing beamsplitter.

2.2. Two attack models

In the LLO CVQKD system, the total noise[17,23] of the system is expressed as

$$\xi_t = \xi_0 + \xi_{AM} + \xi_{ADC} + \xi_{LE} + \xi_{phase}. \tag{1}$$

Here, ξ_0 is the system excess noise from unidentified or unprotected sources, $\xi_{AM} = (E^A_{Smax})^2 10^{-d_{dB}/10}$ is the noise induced by the finite dynamics of the realistic amplitude modulator, $E^A_{Smax} \approx \sqrt{10V_A}$ represents the maximum amplitude of the quantum signal, d_{dB} represents the dynamics of the amplitude modulator, ξ_{ADC} is the analog-to-digital quantization noise referring to the channel input with the expression $\geq (E^A_{Smax})^2/(12\times 2^n)$ with n representing the quantization bits, $\xi_{LE} = 2|E^A_{Smax}|^2/R_e$ is the noise induced by the leakage from the reference pulse to the signal with R_e being the finite extinction ratio, ξ_{phase} is induced by the inability of the reference pulse to fully correct the phase rotation of the quantum signal, which can be compartmentalized into two parts, the trusted phase noise and untrusted phase noise. Part of the phase reference measurement noise associated with the detector efficiency η and the electronic noise v_{el} as well as the phase reference intensity on the receiver side that can be locally calibrated by Bob is considered to be trusted. The trusted phase noise[17,21,23] can be expressed as

$$\xi^T_{phase} = V_A \frac{2-\eta+2v_{el}}{\eta T E_R^2}, \tag{2}$$

where V_A is the modulation variance, T is the transmittance of the channel, η is the efficiency of the heterodyne detector, v_{el} is the electrical noise of the detector. E_R is the amplitude of the phase reference on Bob's side.

When Eve performs the reference pulse attack,[21] the excess noise induced by Eve will be eliminated by induc-

ing the trusted phase noise. The specific implementation scheme is shown in Fig. 2 (attack 1). The reference pulse attack occurs in the process of signal transmission. When both the signal and the reference pulse are transmitted via the standard single-mode fiber with an attenuation coefficient of $\alpha_{\mathrm{Std}} = 0.2$ dB/km, Eve can separate the signal and reference pulses into two separate channels to transmit, the signal pulse is transmitted in the standard channel with α_{Std} and the reference pulse will be transmitted in the low-loss channel with $\alpha_{\mathrm{Low}} < \alpha_{\mathrm{Std}}$. Then, Eve reassembles the signal and the reference pulse at Bob's side. Because the reference pulse is transmitted through the low-loss channel, the attenuation factor of the channel is smaller and the reference pulse intensity E_{R}^2 will be increased. Then, the trusted noise is reduced, Eve can obtain information that is not detected.

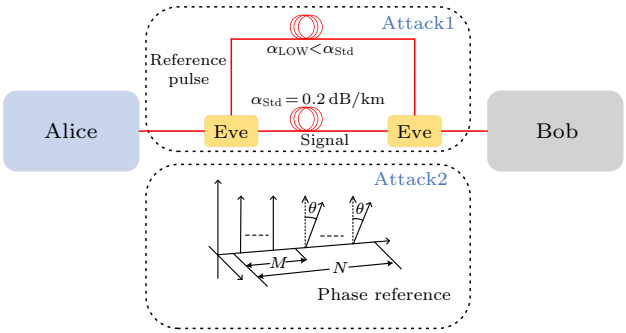


Fig. 2. A pattern diagram of two attacks.

When Eve performs polarization attacks,[23] the excess noise induced by Eve can be hidden by overestimating the trusted phase noise. In the actual LLO CVQKD system, the phase references intensity E_{R}^2 is usually measured before the QKD runs, and the trusted noise is calibrated meanwhile. Therefore, if Eve manipulates the intensity of the phase reference during quantum key distribution process, while Bob still uses the previously measured intensity to estimate the trusted phase noise, Alice and Bob will get the insecure key. In the practical system, due to the random disturbance of the channel, the signal will have the polarization drift and needs the polarization compensation. However, the polarization compensation cycle contains N phase reference pulses, of which only M pulses are used to measure the angle of polarization drift, as shown in Fig. 2 (attack 2). Therefore, in order to implement a polarization attack, Eve can intercept all quantum signals and reference pulse sent by Alice at the channel input and then separates them into her own two perfect quantum channel, then deviates the polarization direction of $N-M$ reference pulses, thus the intensity projection of $N-M$ pulses becomes $E_{\mathrm{R}}^2\cos^2\theta$, where θ represents the misalignment angle when the polarization direction deviates. At this time, if Bob still uses the previously measured intensity of the phase reference to estimate the trusted phase noise, Bob will overestimate the trusted phase noise of the system and then Eve can hide her attack.

3. Data preparation and model training

3.1. Preparation of datasets

In our system, a standard Gaussian modulation protocol is used, Alice prepares a set of coherent states $|X_{\mathrm{A}} + \mathrm{i}P_{\mathrm{A}}\rangle$, where the two quadrature values X_{A} and P_{A} obey the central Gaussian distribution with variance $V_{\mathrm{A}}N_0$. Here, N_0 is the vacuum shot noise unit. Alice transmits the prepared data to Bob through the channel and then Bob performs heterodyne detection with the help of an LO pulse generated locally. In this way, Alice and Bob will obtain two sets of data $x = \{x_1, x_2, \ldots, x_N\}$ and $y = \{y_1, y_2, \ldots, y_N\}$, where x is the quadrature value (X_{A} and P_{A}) modulated by Alice, and y is the quadrature value (X_{B} and P_{B}) measured by Bob. At this point, the variance of Alice's and Bob's data is given by

$$V_x = V_{\mathrm{A}}N_0, \tag{3}$$

$$V_y = \eta T(V_{\mathrm{A}}N_0 + \xi_{\mathrm{t}}) + N_0 + V_{\mathrm{el}}. \tag{4}$$

In order to prepare the datasets for the training model, we analyze the impact of the two attacks on the parameters of the practical LLO CVQKD system in detail (please see Appendix A). We find that the reference pulse attacks affect the intensity E_{R}^2 and variance V_{r} of the reference pulse. The polarization attacks affect the intensity E_{R}^2, variance V_{R} of the reference pulses, variance V_y and the total noise of system ξ_{t}. Based on the above analysis, in the following we will discuss the features that can be extracted and used for deep learning.

3.2. Feature extraction

Although the above-mentioned system parameters will have different results in different attacks, considering that if these parameters are measured directly as characteristics, additional optical devices or parameter evaluation modules must be added to the system, which may pose a threat to the performance and practical security of the system. However, the data received by Bob is easy to obtain and closely related to the above-mentioned system parameters, so the data measured by Bob can be selected and we can add a preprocessing module to automatically convert the measured data into the required feature vectors. We can preprocess the data received by Bob and obtain the variance, maximum value, minimum value, mean value, median value and skewness of the signal pulse and the reference pulse respectively. A total of 12 features form the final feature vector. The feature vector can be expressed as $\boldsymbol{u} = \{V_y, V_{\mathrm{r}}, Y_{\max}, R_{\max}, Y_{\min}, R_{\min}, Y_{\mathrm{mean}}, R_{\mathrm{mean}}, Y_{\mathrm{median}}, R_{\mathrm{median}}, Y_{\mathrm{skew}}, R_{\mathrm{skew}}\}$ and M feature vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N\}$ form the input of the GAN model. To avoid data imbalance, 10000 feature vectors are generated for each attack type. The

feature vectors are input in time series and are grouped into groups of every 100.

3.3. Deep-learning method and training process

3.3.1. Deep-learning method

In order to strengthen the practical security level of the LLO CVQKD system, we introduce a neural network model named GAN[36] originated from game theory. Based on three practical scenarios of reference pulse attack, polarized attack and unattacked state, our proposed model aims to identify and classify attacks. The advantage of our proposed model is to detect and classify attacks without reducing the maximum transmission distance and security key rate of the current system, without adding additional optical devices and with minimal impact on system running. Moreover, it is very easy to use, one only needs to pass the data received by Bob into the model and no additional operations are required.

Our proposed model consists of generator and discriminator. Both the generator and discriminator backbone uses a convolutional neural network (CNN details refer to Appendix B). As shown in Fig. 3, in the discriminator, the input layer is normalized and connected to a two-layer CNN and then passes through a fully connected layer to the output layer. The output layer has two parts, a binary classifier and a multi-classifier for adversarial training. In addition to the CNN layer and the output layer, the input of the linear layer is a linear combination between the output weight w and the bias b of the previous layer. The calculation formula of the linear layer is

$$y_j^{\mathrm{h}} = \sigma_{\mathrm{Relu}}\left(\sum_{i=1} x_i w_{i,j}^{\mathrm{h}} + b_j^{\mathrm{h}}\right), \tag{5}$$

where $w_{i,j}^{\mathrm{h}}$ is the weight between the jth element of the current layer and the ith element of the previous layer, and b_j^{h} is the bias of the jth element in the current layer. The activation function σ_{Relu} can be expressed as

$$\sigma_{\mathrm{Relu}} = \max(0, x). \tag{6}$$

In the output layer, the two outputs can be expressed by

$$\sigma_{\mathrm{Softmax}} = \frac{\mathrm{e}^{x_i}}{\sum_i \mathrm{e}^{x_i}}, \tag{7}$$

$$\sigma_{\mathrm{Sigmoid}} = \frac{1}{1+\mathrm{e}^{-x}}. \tag{8}$$

The structure of the generator is similar to the discriminator. The input layer is connected to a two-layer CNN after normalization and then passes through a fully connected layer to output fake data. The fake data is distributed data of various attacks learned from the training epoch and is used as the input of the discriminator for adversarial training. The generator is used as an auxiliary module for training the discriminator, which continuously learns and imitates the distribution of various attack features in the latent space.

A discriminator can be equivalent to a regular deep-learning model, the only difference is that it has two inputs. The input of the discriminator is divided into real data and fake data generated by the generator. The output of the discriminator is the judgment of whether the input data is true or false. Both the generator and the discriminator can be fully trained and continuously improved in performance by the above-mentioned confrontation.

Our model borrows from auxiliary classifier GAN(acGAN),[37] which is different from regular GAN. The input of the generator in GAN is random noise. The input of the generator in acGAN adds a categorical variable to generate more realistic fake data. The output of the discriminator in acGAN adds a multi-classification to classify the security problems existing in the system. After introducing acGAN, our proposed model has better adversarial resistance, classifies the attacks existing in the current system, and accurately identifies abnormal potential threats.[38]

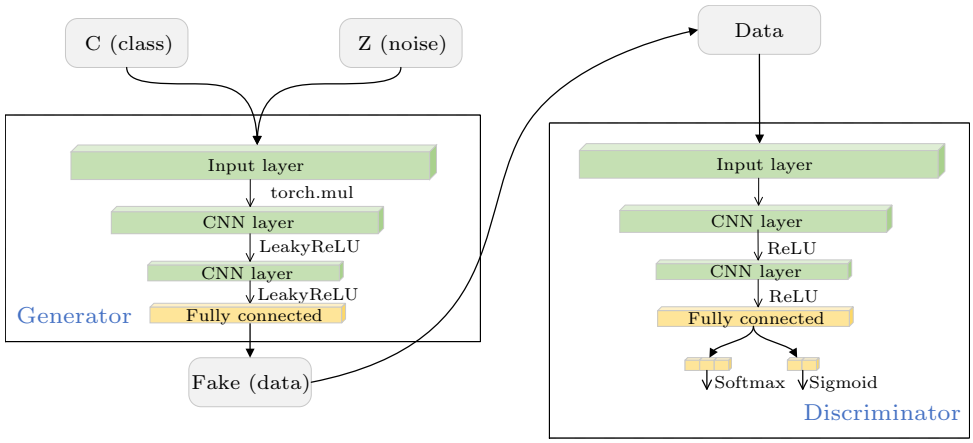


Fig. 3. Schematic diagram of our proposed model containing the general flow of the training and the details of the GAN model. The green squares are the network structures in the generator and discriminator, and the yellow squares are their output layers. The four gray squares represent the flow of data between the generator and the discriminator.

3.3.2. Training process

We implemented the training and testing of the GAN model on the Jupyter Notebook. We trained for 10–15 min on a laptop with 16 GB of memory, an Intel Core 4.0 GHz processor, and a Windows 11 Professional operating system. In the experiments, the learning rate of the GAN model is set to 0.01 and the number of iterations is 200. The dataset size of each attack type is 5×10^7, and the number of pulses in each block is $Q=5\times10^3$, therefore, the dataset of each attack type can be divided into 10000 feature vectors. Three types of data including unattack, reference pulse attack, and polarization attack constitute a training set of 30000 feature vectors. Then in order to reduce memory usage and improve training speed, each batch size of 100 feature vectors is used as an input to the model.

In the specific training process, we firstly use the Pandas library to subtract the unqualified part of the obtained feature vector and convert the dataset to a standardized format. Then we shuffle and normalize the data without changing the accuracy, so that the fluctuations in the training process are smaller and the convergence speed of the model can be improved. Finally, a matrix of 100×12 pixels is used as the input of the model, which is formed by a batch of 100 feature vectors. As shown in Fig. 3, a random noise and a classification variable are input in the generator, where the classification variable is the real classification result. The generator generates a set of fake data based on the real classification results. The discriminator performs discrimination twice. For fake data, the discriminator judges whether the input data is fake data, and gives the classification result. For real data, the discriminator performs gradient descent with the sum of two loss functions for judging fake data and attack classification. After adversarial training, the discriminator can effectively classify attacks.

4. Performance analysis of the detection model

4.1. Evaluation indexes

In order to comprehensively evaluate the performance of our attack detection model, we use the accuracy rate, principal component analysis (PCA),[39] confusion matrix, and receiver operating characteristic (ROC)[40] curve to evaluate the model. PCA is a common data analysis method, mainly used for dimensionality reduction of high-dimensional data. We use PCA to reduce the data dimension of reference pulse attack, polarization attack, and no attack to two dimensions, and make a PCA graph. From this, we can see the degree of discrimination of the data before and after GAN classification. The confusion matrix is mainly used to compare the classification results and the actual measured values. In the confusion matrix, each column represents the predicted category, and the total number of each column indicates the number of data predicted as that category. Each row represents the true category of data, and the total number of data in each row represents the number of data instances of this category. Moreover, the four concepts TP, TN, FP, and FN are all used in the calculation formulas of accuracy rate and ROC curve, so these four concepts are introduced firstly. TP, TN, FP, and FN are mainly used to count binary classification problems and divide samples into positive samples and negative samples. TP indicates that the data is attacked by Eve and recognized as an attack by GAN. TN indicates that the data is not attacked by Eve and is not recognized as an attack by GAN. FP means that the data is not attacked by Eve but be recognized as an attack by GAN. FN indicates that the data is attacked by Eve but not be recognized as attack by GAN. Then we can obtain

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\times100\%, \tag{9}$$

$$\text{TPR}=\frac{\text{TP}}{\text{TP}+\text{FN}}\times100\%, \tag{10}$$

$$\text{FPR}=\frac{\text{FP}}{\text{FP}+\text{TN}}\times100\%. \tag{11}$$

As shown in Eq. (9), the accuracy rate predicts the proportion of correctly recognized samples to all samples, and the correct prediction may have positive samples or negative samples. As shown in Eq. (10), the true positive rate (TPR) indicates the proportion of the attacked and predicted correctly accounts for the total correct prediction. As shown in Eq. (11), false positive rate (FPR) indicates the proportion of the attacked and wrong predictions to the total prediction errors. The ROC curve describes the true positive rate of a certain classifier as a function of its FPR.[41] In binary classification problems, the threshold is an artificially set decimal between 0 and 1. When the probability value of the model prediction result is greater than the threshold, it is a positive sample. By constantly changing the threshold, a set of correspondences between FPR and TPR can be obtained. We use FPR and TPR as the x and y axes, respectively, and then connect the coordinate points under different thresholds. The resulting curve is the ROC curve.

4.2. Performance evaluation

In order to judge the performance of our proposed model, we firstly use accuracy rate and loss functions as indicators to evaluate model performance during the training phase. Figure 4(a) shows the accuracy rate of our model during the training process. The accuracy rate of the model increases steadily in the first 50 rounds of training and stabilized at about 99.8% after 50 rounds. The loss function gradually decreases in the first 50 training epochs and then stabilizes. This shows that our model converges fast and has a high accuracy rate. Figure 4(b) shows a confusion matrix. The diagonal of the confusion matrix represents the amount of agreement between the predicted values and the true values. As is shown, when the system is not under attack, the data are all concentrated on the diagonal

line, which means that our model can well distinguish whether the system is under attack or not. When there is a reference pulse attack on the system, most of the data is concentrated on the diagonal line of the second column, which means that our model can also easily determine whether the system is attacked by a reference pulse. In the third column, when there is a polarization attack on the system, the data is more scattered than when there is a reference pulse attack, which means that polarization attacks are relatively difficult to distinguish.

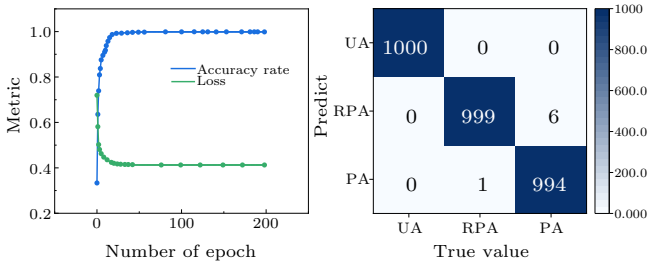


Fig. 4. (a) Variation curve of accuracy rate and loss function during model training. (b) Confusion matrix of the GAN model with 3000 feature vectors. UA: the unattacked state. RPA: reference pulse attack. PA: polarization attack.

To further judge the performance of our proposed model, we tested the model on independent experimental data, which can be considered as the test set. The test set adopts three different sequence groups, including unattack, reference pulse attack, and polarization attack, a total of 3000 feature vectors. We firstly test our model using PCA. As can be seen from Fig. 5(a), there is a significant gap between the situation with and without attacks, while the reference pulse attack and the polarization attack are difficult to distinguish through statistical analysis. After the classification by our GAN model, we can see the result shown in Fig. 5(b), which illustrates that reference pulse attack and polarization attack can be easily distinguished, and our model plays a great role in data differentiation.

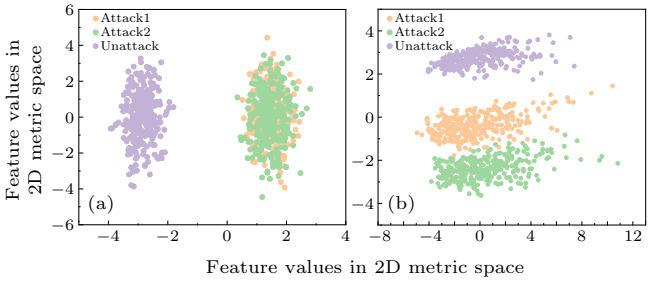


Fig. 5. (a) Results of PCA before GAN classification, (b) results of PCA processing after GAN classification.

The ROC curve can furthermore clearly represent the performance of the model for each type of attack. The closer the ROC curve is to the point (0, 1), the better performance of the model. In Fig. 6(a), the gray dotted line in the figure is the result of random guessing, indicating that there is no change in performance without applying any model. The three solid lines correspond to the three binary classification problems converted from the multiclass classification problem of the model. The blue solid line is to classify unattack into one category while grouping reference pulse attack and polarization attack into another category. The red solid line is to classify reference pulse attack into one category while grouping unattack and polarization attacks into another category. The green solid line is to classify polarization attack into one category while grouping unattack and reference pulse attack into another category. To express the model performance more intuitively, we calculated the area under the curve (AUC) of each label. The closer the AUC value is to 1, the better performance of the model. From Fig. 6(a), we can see that the blue solid line indicates that the AUC value is 1, which means that our proposed model can perfectly predict whether there is an attack on the current system. The AUC values of the red solid line and the green solid line are both greater than 0.99, indicating that the model can distinguish reference pulse attack and polarization attack with a high accuracy rate.

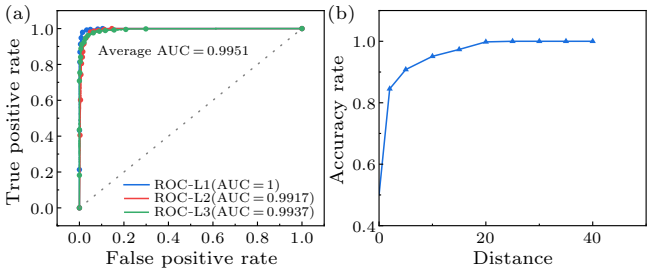


Fig. 6. (a) ROC curve of the model, where the gray dotted line is the performance of random guesses. (b) The classification ability of the GAN model for attacks under different transmission distances.

The above performance analysis is based on a fixed transmission distance. However, Eve launches the reference pulse attack and the polarization attack by manipulating the reference pulse in channel. Eve may obtain more security keys when the transmission distance becomes longer. Therefore, we analyze the performance of our model at different distances in Fig. 6(b). The horizontal axis is the transmission distance and the vertical axis is the model accuracy rate. The accuracy rate is 84.5% at 5 km, then steadily increases to 99.8% at 20 km, and finally stabilized at 1. Although Eve can obtain more security keys during long-distance transmission, the accuracy rate of our model will also be greatly improved. Therefore, our model can classify and identify attacks well when the transmission distance is long.

5. Conclusion

To detect different attack types in LLO CVQKD systems, this paper proposes a GAN-based deep learning model for attack classification in the system. We firstly investigate the characteristics of pulses with and without attacks, then a set of feature vectors are established and be automatically turned

into appropriate data as the input for training the GAN model with an automatic preprocessing module. To improve the accuracy of the proposed model, a convolutional neural network is employed in two parts of the model. The results show that the trained GAN model can identify and classify attacks with an accuracy of 99.8% over a transmission distance of 20 km. Moreover, by comparing the accuracy of the model under different transmission distances, we find that our model has a better ability to distinguish attacks with the increase of communication distance.

Appendix A: The impact of two attacks on parameters of system

In order to prepare the dataset, we need realistic assumptions based on Alice and Bob's implementation settings and Eve's capabilities. In our experiments, based on the realistic LLO CVQKD standard assumption,[16–20] the system parameters are set as $V_{\rm A}=4$, $\eta=0.5$, $\xi_{\rm t}=0.043$, $T=10^{-\alpha L/10}$, where L is the transmission distance of 20 km and α is the loss coefficient of the optical fiber. Then we introduce the attack parameters and experimental methods of reference pulse attack and polarization attack.

In the reference pulse attack, Eve separates the signal and reference pulse into two separate channels to transmit, the signal pulse is transmitted in the standard channel and the reference pulse will be transmitted in the low-loss channel, which means that the reference pulse intensity will be increased and then the trusted phase noise will be decreased. The impact of reference pulse attack on the system can be expressed as

$$E_{\rm R}^{2\prime}=E_{\rm R}^{2}\times\frac{T_{\rm low}}{T_{\rm std}}, \tag{A1}$$

$$V_{\rm r}{}'=\eta T_{\rm low}(V_{\rm r}N_0+\xi_{\rm t})+N_0+V_{\rm el}, \tag{A2}$$

where $T_{\rm low}$ is the channel transmittance of the reference path increased by Eve, $T_{\rm std}$ is the transmittance of fiber with attenuation coefficient $\alpha_{\rm Std}=0.2$ dB/km. In the reference pulse attack,[21] the increase of the channel transmittance of the reference path affects the reference pulse intensity $E_{\rm R}^2$ and the variance of the reference pulse received by Bob.

In the polarization attack, Eve intercepts all quantum signals and reference pulse sent by Alice at the channel input and then separates them into her own two perfect quantum channel, then deviates the polarization direction of $N-M$ reference pulses. Thus the intensity of the reference pulse decrease will be reduced to $E_{\rm R}^2\cos^2\theta$ and the trusted phase noise in the system will be increased as follows:[23]

$$E_{\rm R}^{2\prime}=E_{\rm R}^{2}\cos^2\theta, \tag{A3}$$

$$\Delta\xi_{\rm phase}^{\rm T}=\xi_{\rm phase}^{\rm T}\left[\frac{1}{K+(1-K)\cos^2\theta}\right]-\xi_{\rm phase}^{\rm T}, \tag{A4}$$

where $k=M/N$ is the ratio of the measured pulses to the compensation cycle pulses, which is named as polarization measurement rate (PMR). Then, the total excess noise of the system will be given by

$$\xi_{\rm t}'=\xi_{\rm t}+\Delta\xi_{\rm phase}^{\rm T}. \tag{A5}$$

Hence the variance of the signal and reference pulse can be written as[23]

$$V_y{}'=\eta T_{\rm low}(V_{\rm A}N_0+\xi_{\rm t}+\Delta\xi_{\rm phase}^{\rm T})+N_0+V_{\rm el}, \tag{A6}$$

$$V_{\rm r}{}'=\eta T_{\rm low}(V_{\rm r}N_0+\xi_{\rm t}+\Delta\xi_{\rm phase}^{\rm T})+N_0+V_{\rm el}. \tag{A7}$$

From the above, the impact of the two attacks on the parameters can be obtained. The two attack modes will cause different numerical changes to the following parameters, V_y, $V_{\rm r}$, $E_{\rm R}^2$ and the total excess noise of system $\xi_{\rm t}$, which can all be used as feature vectors of the GAN model. Considering the data received by Bob is easy to obtain and closely related to the above-mentioned system parameters, the data measured by Bob can be selected in order to avoid the security threat of additional devices required for direct measurement of system parameters. Moreover, we add a preprocessing module and when only data distribution is obtained, Bob will input distribution data to the GAN model after preprocessing to determine whether the current system is safe or not. It establishes a detection model by monitoring features of the pulse without adding additional devices.

Appendix B: Supplementary information of CNN

Convolutional neural networks (CNNs) are currently one of the most widely used neural network architectures. The CNN in this article is part of our GAN model, so we mainly focus on its convolutional layer[42] and pooling layer.

The role of the convolutional layer is to extract information in the input data, which is called features. The specific operation process of the convolutional layer is to multiply and sums the elements corresponding to each position on the input matrix x and the convolution kernel k (weight matrix), and saves the result to the corresponding position of the output. As shown in Fig. B1(a), the convolution kernel slides from the upper left corner of the matrix, and slides through each area of the matrix in turn according to the specified pixel pitch. Take the yellow box in the input image as an example. The data in the yellow box is multiplied with the kernel matrix, and the obtained data is stored in the corresponding position of the feature map. The convolution operation formula is

$$y_j=\max\left[\sum k_{i,j}x_i+b_{j,0}\right]. \tag{B1}$$

Here, x_i is the i-th input, $k_{i,j}$ is the convolution kernel between the i-th input mapping and the j-th output matrix, and $b_{j,0}$ is the bias of the j-th output matrix.

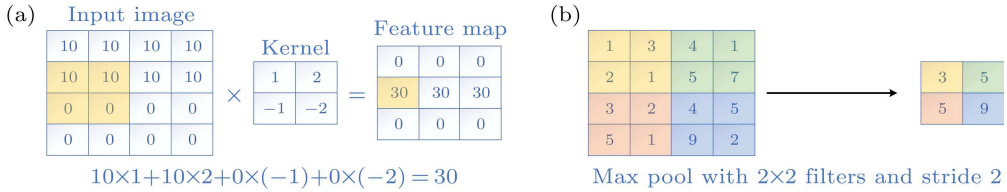


Fig. B1. (a) An example of the calculation process of the convolutional layer. (b) An example of the calculation process of the pooling layer.

The role of the pooling layer is to select the features extracted by the convolutional layer, to reduce the number of parameters, and to prevent the model from overfitting. Pooling layers usually follow convolutional layers, and each convolutional layer is teamed with a pooling layer. The pooling function used in this paper is the most popular max-pooling function. As shown in Fig. B1(b), blocks of the same color correspond to each other. The function of max pooling is to select the largest number in the block and store it in the corresponding block, which can reduce the number of parameters. The pooling operation formula is

$$y_{i,j}^{k} = \max(x_{i\cdot s+t_1, j\cdot s+t_2}^{k}). \tag{B2}$$

After several convolutions and pooling, the fully connected layer converts all feature matrices into one-dimensional feature vectors. Finally, the neural network can output classification results.

Acknowledgement

Project supported by the National Natural Science Foundation of China (Grant No. 62001383).

References

[1] Grosshans F and Grangier P 2002 *Phys. Rev. Lett.* **88** 057902
[2] Gisin N, Ribordy GG, Tittel W and Zbinden H 2002 *Rev. Mod. Phys.* **74** 145
[3] Weedbrook C, Pirandola S, García-Patrón R, Cerf N J, Ralph T C, Shapiro J H and Lloyd S 2012 *Rev. Mod. Phys.* **84** 621
[4] Renner R and Cirac J I 2009 *Phys. Rev. Lett.* **102** 110504
[5] Furrer F, Franz T, Berta M, Leverrier A, Scholz V B, Tomamichel M and Werner R F 2012 *Phys. Rev. Lett.* **109** 100502
[6] Leverrier A 2017 *Phys. Rev. Lett.* **118** 200501
[7] Zhang Y C, Chen Z Y, Pirandola S, Wang X Y, Zhou C, Chu B J, Zhao Y J, Xu B J, Yu S and Guo H 2020 *Phys. Rev. Lett.* **125** 010502
[8] Pi Y D, Wang H, Pan Y, Shao Y, Li Y, Yang J, Zhang Y C, Huang W and Xu B J 2023 *Opt. Lett.* **48** 1766
[9] Tian Y, Zhang Y, Liu S H, Wang P, Lu Z G, Wang X Y and Li Y M 2023 *Opt. Lett.* **48** 2953
[10] Marie A and Alleaume R 2017 *Phys. Rev. A* **95** 012316
[11] Ma X C, Sun S H, Jiang M S and Liang L M 2013 *Phys. Rev. A* **88** 022339
[12] Jouguet P, Kunz-Jacques S and Diamanti E 2013 *Phys. Rev. A* **87** 062313
[13] Qi B, Lougovski P, Pooser R, Grice W and Bobrek M 2015 *Phys. Rev. X* **5** 041009
[14] Soh D B S, Brif C, Coles P J, Luetkenhaus N, Camacho R M, Urayama J J and Sarovar M 2015 *Phys. Rev. X* **5** 041010
[15] Huang D, Huang P, Lin D, Wang C and Zeng G H 2015 *Opt. Lett.* **40** 3695
[16] Wang T, Huang P, Zhou Y M, Liu W Q, Ma H X, Wang S Y and Zeng G H 2018 *Opt. Express* **26** 2794
[17] Wang T, Huang P, Zhou Y M, Liu W Q and Zeng G H 2018 *Phys. Rev. A* **97** 012310
[18] Xu F H, Ma X F, Zhang Q, Lo H K and Pan J W 2020 *Rev. Mod. Phys.* **92** 025002
[19] Guo H, Li Z Y, Yu S and Zhang Y C 2021 *Fundam. Res.* **1** 96
[20] Wang H, Pi Y D, Huang W, Li Y, Shao Y, Yang J, Liu J L, Zhang C L, Zhang Y C and Xu B J 2020 *Opt. Express* **28** 32882
[21] Ren S J, Kumar R, Wonfor A, Tang X K, Penty R and White I 2019 *J. Opt. Soc. Am. B* **36** B7
[22] Shao Y, Wang H, Pi Y D, Huang W, Li Y, Liu J L, Yang J, Zhang Y C and Xu B J 2021 *Phys. Rev. A* **104** 032608
[23] Shao Y, Pan Y, Wang H, Pi Y D, Li Y, Ma L, Zhang Y C, Huang W and Xu B J 2022 *Entropy* **24** 992
[24] Shao Y, Li Y, Wang H, Pan Y, Pi Y D, Zhang Y C, Huang W and Xu B J 2022 *Phys. Rev. A* **105** 032601
[25] Alaghbari K A, Rumyantsev K, Eltaif T, Elmabrok O and Lim H S 2021 *IEEE Photon. J.* **13** 7600107
[26] Huang B, Huang Y M and Peng Z M 2019 *Opt. Express* **27** 20621
[27] Huang D, Liu S S and Zhang L 2021 *Photonics* **8** 511
[28] Huang W T, Mao Y Y, Xie C and Huang D 2019 *Phys. Rev. A* **100** 012316
[29] Mao Y Y, Wang Y J, Huang W T, Qin H, Huang D and Guo Y 2020 *Phys. Rev. A* **101** 062320
[30] Luo H S, Zhang L, Qin H, Sun S H, Huang P, Wang Y J, Wu Z J, Guo Y and Huang D 2022 *Phys. Rev. A* **105** 042411
[31] Mao Y Y, Huang W T, Zhong H, Wang Y J, Qin H, Guo Y and Huang D 2020 *New J. Phys.* **22** 083073
[32] Du H W and Huang D 2022 *Photonics* **9** 177
[33] Liu W Q, Huang P, Peng J Y, Fan J P and Zeng G H 2018 *Phys. Rev. A* **97** 022316
[34] He Z, Wang Y J and Huang D 2020 *J. Opt. Soc. Am. B* **37** 1689
[35] Li S M, Yin P Z, Zhou Z H, Tang J H, Huang D and Zhang L 2023 *Entropy* **25** 499
[36] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2020 *Commun. ACM* **63** 139
[37] Odena A, Olah C and Shlens J 2017 *Proceedings of Machine Learning Research* **70** 2642
[38] Ravanbakhsh M, Nabi M, Sangineto E, Marcenaro L, Regazzoni C and Sebe N 2017 *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 1577–1581
[39] Jolliffe I T and Cadima J 2016 *Philos. Trans. R. Soc. A* **374** 20150202
[40] Cook N R 2007 *Circulation* **115** 928
[41] Liao Q, Xiao G, Zhong H and Guo Y 2020 *New J. Phys.* **22** 083086
[42] Shelhamer E, Long J and Darrell T 2017 *IEEE Trans. Pattern Anal.* **39** 640