SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2024, Vol. 67, Iss. 10, 202501:1–202501:14
https://doi.org/10.1007/s11432-023-3988-x

Automatically identifying imperfections and attacks in practical quantum key distribution systems via machine learning

Jiaxin XU[1,2], Xiao MA[1,2], Jingyang LIU[1,2], Chunhui ZHANG[1,2], Hongwei LI[3*], Xingyu ZHOU[1,2*] & Qin WANG[1,2*]

[1]*Institute of Quantum Information and Technology, Nanjing University of Posts and Telecommunications, Nanjing* 210003*, China;*
[2]*Telecommunication and Networks, National Engineering Research Center, Nanjing University of Posts and Telecommunications, Nanjing* 210023*, China;*
[3]*Henan Key Laboratory of Quantum Information and Cryptography, Strategic Support Force Information Engineering University, Zhengzhou* 450001*, China*

Received 17 August 2023/Revised 17 October 2023/Accepted 21 March 2024/Published online 10 September 2024

Abstract The realistic security of quantum key distribution (QKD) systems is currently a hot research topic in the field of quantum communications. There are always defects in practical devices, and eavesdroppers can make use of the security risk points of various devices to obtain key information. To date, current types of security analysis tend to analyze each security risk point individually, thereby posing great challenges for the overall security evaluation of QKD systems. In this paper, for the first time, we employ machine learning algorithms to identify the defects of different devices and certain attacks in real time, with an accuracy of 98%. It provides a novel solution for the practical security evaluation of QKD systems, thereby addressing the bottleneck problem of multiple risk points being difficult to address simultaneously in QKD systems, thus paving the way for the future large-scale application of quantum communication networks.

Keywords quantum key distribution, device imperfections, eavasdropping attacks, real-time detection, machine learning algorithms

1 Introduction

As the central core of quantum communication, quantum key distribution (QKD) [1], which relies mainly on the law of quantum mechanics [2,3], can then provide remote users with unconditional secure keys and thus, have attracted worldwide attention since the first BB84 protocol was proposed [4]. However, there is a wide gap between theoretical security and the practical QKD implementation [5,6] due to imperfect devices and the threat of potential eavesdropping attacks.

Numerous studies have been conducted, focusing on the various imperfections in the practical QKD systems, such as intensity fluctuation [7], state preparation flaws [8,9], intensity modulator flaws [10], detection efficiency mismatches [11–13], dead time [14,15], and after-pulse effect [16]. These imperfections may then lead to corresponding errors and side-channel information leakage and may even result in the situation being fully manipulated by the eavesdropper (Eve). Several quantum hacking attacks exploiting such realistic security loopholes have been reported, such as phase-remapping attack [17,18], distinguishable decoy states [19], laser-damage attacks [20,21], a wavelength-dependent attack [22], an intercept-resend attack [23], and a time-shift attack [12].

In most previous studies, the flaws were routinely considered individually using different models. To address this problem, Sun et al. [24] and Chen et al. [25] both recently proposed a systematic performance analysis that addresses both source and detection imperfections. Although these methods analyze the errors induced by device flaws effectively, they need to calibrate each error separately, which will not only

* Corresponding author (email: lihow@ustc.edu.cn, xyz@njupt.edu.cn, qinw@njupt.edu.cn)

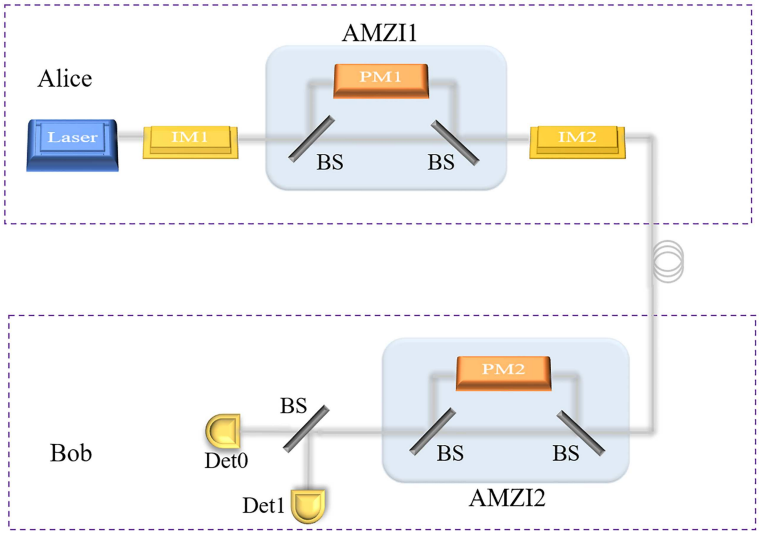

Figure 1 (Color online) Device diagram based on time-bin phase coding. IM: intensity modulator, PM: phase modulator, AMZI: asymmetric Mach-Zehnder interferometer, BS: beam splitter, Det: detector. Only the main devices are shown in the figure.

take a lot of time but also result in increasing system complexity and additional consumption. Therefore, realizing the identification and calibration of system flaws in real time is of great importance for the practical implementation of QKD systems.

Machine learning is, in essence, a general data processing technology, which includes a large number of required learning algorithms [26–29]. Due to their advantages in data processing, machine learning schemes can be considered a candidate to replace traditional detection schemes. Huang et al. [30, 31] proposed using machine learning methods to detect quantum attacks in continuous-variable QKD (CV-QKD), providing a useful tool for the detection of attacks in CV-QKD systems.

In this paper, we propose a universal real-time detection method to identify device imperfections and eavesdropping attacks via machine learning. We construct a general mathematical model for practical QKD systems by incorporating device imperfections and taking potential eavesdropping attacks into account. Then, we implement machine learning algorithms, e.g., the random forest algorithm, to identify device imperfections and attacks in real time. Then, one can either give corresponding feedback control to depress imperfections of QKD systems and reduce the influence of Eve's attacks or take those imperfections into account in the security analysis to maintain their subsequent security.

2 Modeling

In this context, we employ the commonly used time-bin phase-coding BB84 QKD system as an illustration. Indeed, our present method can also apply to some other coding schemes and QKD protocols.

Figure 1 shows the schematic of a typical time-bin phase-coding BB84 QKD system consisting of intensity modulators (IMs), asymmetric Mach-Zehnder interferometers (AMZIs), and phase modulators (PMs). IM1 is used herein to realize the modulation of different intensities. AMZIs are fabricated by splicing two beam splitters (BSs) end to end to form a long and a short arm, and the PM is integrated into just one of AMZI's arms. After passing through the AMZI1, the quantum light is split into front and rear pulses. IM2 performs the chopping operation on the Z basis or passes the front and rear pulses with equal intensity on the X basis. PM1 and PM2 realize phase modulation and demodulation, respectively. The modulated pulses interfere with the BS, and the output results are detected by two threshold detectors, namely, Det0 and Det1.

Next, we introduce the specific modeling process for three parties: the sender, the receiver, and the attacker.

2.1 Sender

Alice firstly prepares and emits one of the four states $|\phi_{0Z}\rangle, |\phi_{1Z}\rangle, |\phi_{0X}\rangle, |\phi_{1X}\rangle$. Here, $|\phi_{j\alpha}\rangle$ denotes Alice's state with the bit j (0 or 1) in the α basis (Z or X). The concrete expressions of prepared states

are given by [32]:

$$\begin{aligned}
|\phi_{0Z}\rangle &= \cos\left(\frac{\delta_1}{2}\right)|0_z\rangle + \sin\left(\frac{\delta_1}{2}\right)|1_z\rangle,\\
|\phi_{1Z}\rangle &= \sin\left(\frac{\delta_2}{2}\right)|0_z\rangle + \cos\left(\frac{\delta_2}{2}\right)|1_z\rangle,\\
|\phi_{0X}\rangle &= \sin\left(\frac{\pi}{4}+\frac{\delta_3}{2}\right)|0_z\rangle + \cos\left(\frac{\pi}{4}+\frac{\delta_3}{2}\right)\mathrm{e}^{\mathrm{i}\theta_1}|1_z\rangle,\\
|\phi_{1X}\rangle &= \sin\left(\frac{\pi}{4}+\frac{\delta_4}{2}\right)|0_z\rangle + \cos\left(\frac{\pi}{4}+\frac{\delta_4}{2}\right)\mathrm{e}^{\mathrm{i}(\pi+\theta_2)}|1_z\rangle,
\end{aligned} \tag{1}$$

where $\delta_1, \delta_2, \delta_3, \delta_4$ denote the state preparation errors caused by the finite extinction ratio in IM. θ_1, θ_2 denote the phase errors caused by the deviation of the operating voltage in PM.

In addition, the intensity of each laser pulse is modulated with a certain probability for three different intensity values, μ_1, μ_2, μ_3 ($\mu_1 > \mu_2, \mu_3 = 0$), representing the signal, decoy, and vacuum state, respectively. Due to the instability of the bias voltage in IM, the pattern effect [33], and the non-ideal characteristics of the device itself, the actual intensity of the output of the weak coherent source (WCS) has a random fluctuation. Therefore, the actual value of intensities is set as $\mu_1(1 \pm \lambda_1)$, $\mu_2(1 \pm \lambda_2)$, $\mu_3(1 \pm \lambda_3)$, where $\lambda_1, \lambda_2, \lambda_3$ are the intensity fluctuations.

2.2 Receiver

On Bob's side, we consider the case of two threshold detectors (Det0 and Det1), which can cause detector efficiency mismatches, whose efficiencies are defined as η_0 and η_1, respectively. Moreover, the after-pulse and dead time effects are inevitable for off-the-shelf detectors. For simplicity, we define that only one detector clicking event corresponds to a valid event. Bob measures the received pulses with a probability of 50% in the Z basis and 50% in the X basis.

Here, we define the measurement probability as $P^{\mu}_{\xi_A^a \xi_B^b}$, where $\mu \in \{\mu_1, \mu_2, \mu_3\}$, $a, b \in \{0, 1, 2, 3\}$, which represents the four quantum states sent by Alice (see details in (1)) and received by Bob. Next, we describe the parameter $P^{\mu}_{\xi_A^a \xi_B^b}$ in detail. It is the probability when the corresponding measurement outcome is b after Bob uses the basis measurement $\xi \in \{Z, X\}$ under the condition that Alice prepares the quantum state $\xi_A^a \in \{\phi_{0Z}, \phi_{1Z}, \phi_{0X}, \phi_{1X}\}$, and its expression form is

$$P^{\mu}_{\xi_A^a \xi_B^b} = \sum_{n=0}^{\infty} P_n(\mu) \sum_{j=0}^{n} C_n^j \eta^j (1-\eta)^{n-j} \left(|\langle \xi_A^a | \xi_B^b \rangle|^2\right)^j F(j), \tag{2}$$

where $P_n(\mu)$ is the distribution of the WCS, $F(j)$ is the probability of effectively detection events under the condition of j photon state,

$$F(j) = \begin{cases} 1-d, & j > 0, \\ d(1-d), & j = 0, \end{cases} \tag{3}$$

and d is the dark count rate of the detector. When Bob's measurement result is bit 0, Det0 clicks; whereas Det1 clicks, the detection efficiency corresponds to η_0 and η_1, respectively. That is to say, according to different measurement results, η in (2) is replaced by η_0 or η_1.

When the after-pulse effect [16] is considered,

$$F(j) = \begin{cases} (1-d)(1+P_{\mathrm{ap}}), & j > 0, \\ d(1-d), & j = 0, \end{cases} \tag{4}$$

where P_{ap} is the after-pulse probability [16]. Detail simplification steps of $P^{\mu}_{\xi_A^a \xi_B^b}$ is addressed in Appendix A.

Based on the above analysis, we can calculate the count according to the probability $P^{\mu}_{\xi_A^a \xi_B^b}$ under the corresponding conditions through the total number of pulses and the different selection of the sender and receiver. The specific expression is as follows:

$$M^{\mu}_{\xi_A^a \xi_B^b} = \frac{1}{2} \times N \times P_{\mu} \times P_{\xi_A} \times P_{\xi_B} \times c_{\mathrm{DT}} \times P^{\mu}_{\xi_A^a \xi_B^b}, \tag{5}$$

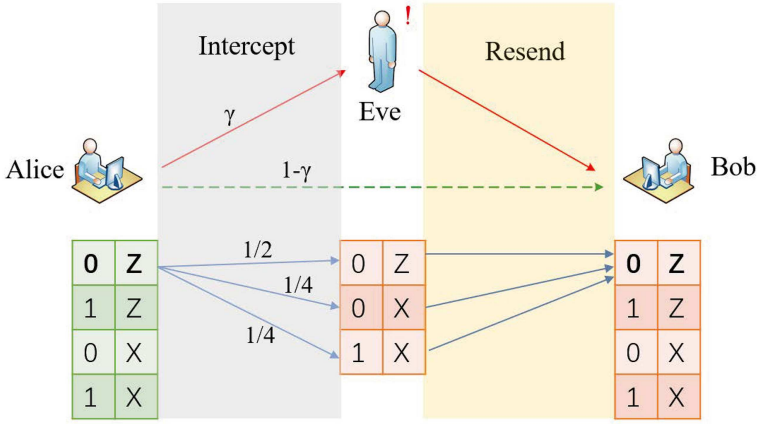

Figure 2 (Color online) Schematic diagram of intercept-resend attack.

where N represents the total number of pulses, $P_\mu \in \{P_{\mu 1}, P_{\mu 2}, P_{\mu 3}\}$ represents the probability that Alice sends different pulse intensity, P_{ξ_A} and P_{ξ_B} represents the possibility of the basis chosen by Alice and Bob. c_{DT} represents the correction coefficient caused by detector dead time τ_{dt} [34], which can be expressed as

$$c_{\mathrm{DT}} = \frac{1}{1 + \zeta \times P^{\mathrm{tot}}_{\xi_A \xi_B} \times \tau_{\mathrm{dt}}}, \tag{6}$$

where ζ represents the system repetition frequency and $P^{\mathrm{tot}}_{\xi_A \xi_B}$ denotes the sum of those probabilities over all intensities μ_1, μ_2, μ_3.

In the above equation, the values of the counts can be directly measured in practical experiments and they will be used as input feature data for the following machine learning models, which will be described in detail in the next Section 3.

2.3 Partial intercept-resend attack

In an intercept-resend attack, Eve intercepts and measures the quantum state that Alice sends to Bob, then reproduces the quantum state based on the measurement result, and sends it to Bob. When Eve launches an intercept-resend attack, a 25% error will be introduced which is easily found by Alice and Bob, see Figure 1 [23].

However, if Eve only intercepts and resends the quantum state sent by Alice with a very small probability (γ), the bit error rate will not increase significantly which is difficult to be found by both communication parties.

As is shown in Figure 2, in the case of Alice sending the bit 0 in the Z basis, Eve reproduces the quantum state based on the measurement results and sends it to Bob. Since there is a 50% chance that the Z basis is chosen for the measurement, Eve has a 50% chance of receiving the state $|\phi_{0Z}\rangle$ (abbreviated as $|0\rangle$) sent by Alice, and a 50% chance of choosing the X basis for the measurement, so Eve has a 25% chance of receiving the state $|\phi_{0X}\rangle, |\phi_{1X}\rangle$ (abbreviated as $|2\rangle, |3\rangle$) sent by Alice, respectively. Therefore, in the event of an intercept-resend attack, the expression of the measurement possibility is shown as

$$P^{\mu}_{\xi^0_A \xi^0_B, \gamma} = \frac{1}{2} P^{\mu}_{\xi^0_A \xi^0_B} + \frac{1}{4} P^{\mu}_{\xi^2_A \xi^0_B} + \frac{1}{4} P^{\mu}_{\xi^3_A \xi^0_B}. \tag{7}$$

In the whole system, the number of pulses sent by Alice is N, in which γN photons are intercepted and resent by Eve, and the remaining $(1-\gamma)N$ photons are transmitted safely in the secure channel; then the measurement probability of the whole system is defined as

$$P^{\mu'}_{\xi^0_A \xi^0_B} = (1-\gamma) \times P^{\mu}_{\xi^0_A \xi^0_B} + \gamma \times P^{\mu}_{\xi^0_A \xi^0_B, \gamma}. \tag{8}$$

When intercept-resend attacks are considered, $P^{\mu}_{\xi^a_A \xi^b_B}$ in (5) should be replaced by $P^{\mu'}_{\xi^0_A \xi^0_B}$, and the expression of the counts becomes

$$M^{\mu}_{\xi^a_A \xi^b_B} = \frac{1}{2} \times N \times P_\mu \times P_{\xi_A} \times P_{\xi_B} \times c_{\mathrm{DT}} \times P^{\mu'}_{\xi^a_A \xi^b_B}. \tag{9}$$

Table 1 System parameters

d	e_d	N	η	L
10^{-8}–10^{-6}	0–0.1	10^8–10^{10}	0.1–0.9	0–100 km

Table 2 Definition of imperfections

Symbol	Considered imperfection	Range
$\delta_1, \delta_2, \delta_3, \delta_4$	State preparation error	0–0.15
θ_1, θ_2	Phase error	0–$\pi/20$
$\lambda_1, \lambda_2, \lambda_3$	Intensity fluctuation	-15%–15%
$P_{\rm ap}$	After-pulse possibility	0–0.1
$\tau_{\rm dt}$	Dead time	10^{-8}–10^{-6}
$\Delta\eta$	Detector efficiency mismatch	0.1–0.8
γ	Intercept-resend attack	0–0.1

It is worth noting that in our method we use machine learning algorithms to detect device imperfections and attacks through different measurement counts $M^{\mu}_{\xi^a_A \xi^b_B}$. In principle, it can also cover other imperfections and attacks that might not change the counting rates of detectors, but have an influence on other observable parameters, e.g., spectrum, or current, by being assisted with other measuring instruments and data. Second, even for those imperfections or attacks which can lead to the rise of quantum bit error rate (QBER), if we neglect them and simply calculate the key rate with QBER, it might cause security problems under certain circumstances. Besides, for a QKD system with device imperfections, e.g., for a polarized passive-basis-choice QKD system [35], Eve can apply the intercept-resend attack and threaten its security. We cannot simply calculate the key rate with the QBER through the standard way.

In the actual system, the number of pulses emitted by Alice is limited. In the actual security analysis, the influence of statistical fluctuation on parameter estimations should be considered. Here, the Gaussian analysis method is adopted for statistical fluctuation analysis. A detailed description of statistical fluctuation can be found in Appendix B.

3 Detection method based on machine learning

In this section, we use machine learning algorithms to detect device imperfections and attacks through different measurement counts $M^{\mu}_{\xi^a_A \xi^b_B}$. Here, we use the classification algorithms to classify the type of device imperfections or attacks in the actual QKD system, and the values of the imperfections are given through the prediction algorithms. Finally, the results are analyzed and discussed.

3.1 Data preprocessing

The chosen system parameters listed as follows are the most crucial ones in calculating the key generation rate for a specific QKD protocol: dark count rate of the detector (d), misalignment error rate (e_d), numbers of pulses (N), efficiency of single photon detectors (η), and transmission distance (L). According to the configuration and performance of practical QKD systems [36, 37], the parameters are generated within an appropriate range (see Table 1).

We unify these system parameters into a vector $\boldsymbol{s}$, $\boldsymbol{s} := \{d, e_d, N, \eta, L\}$. Within the range, we evenly select 4, 3, and 3 values within the range of L, e_d, and η, respectively, and 3 values of d and N with an equal logarithmic space of base 10.

Furthermore, we model the practical QKD system with potential imperfections in devices including IMs, PMs, detectors, and potential eavesdropper attacks. Here, we quantify the effects of various device imperfections and eavesdropping attacks on the QKD system. In Table 2, we list all the imperfections and their symbols, and set a reasonable range for different device defects and attack parameters.

To simplify the calculation, we assume that δ_1 and δ_2 have the same value, and δ_3 and δ_4 have the same value. In addition, we assume that the values of $\lambda_1, \lambda_2, \lambda_3$ are equal. For the imperfection parameters in Table 2, we take 100 evenly spaced values for each parameter in the corresponding range. We unify these imperfection parameters into a vector $\boldsymbol{i}$, $\boldsymbol{i} = \{\delta_1, \delta_3, \theta_1, \theta_2, \lambda, P_{\rm ap}, \tau_{\rm dt}, \Delta\eta, \gamma\}$. By bringing the system parameters in Table 1 and the imperfection parameters in Table 2 into the simulation model, we can calculate the measurement counts of single-photon detectors $M^{\mu}_{\xi^a_A \xi^b_B}$ in different cases. Here, given that

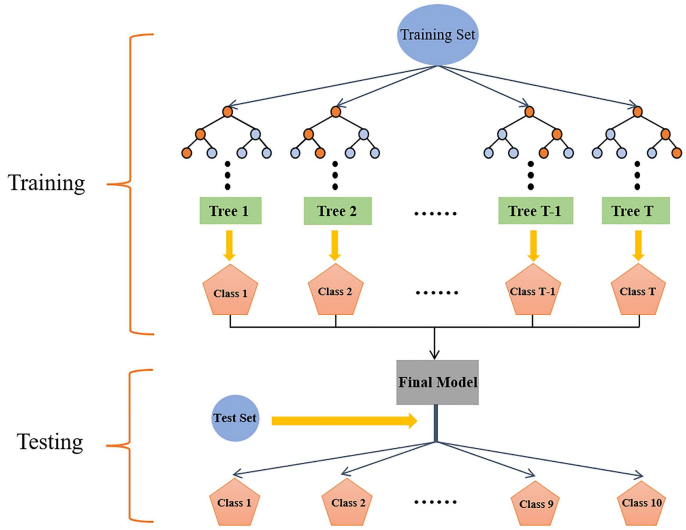

Figure 3 (Color online) Random forest classification algorithm.

$\mu_3 = 0$, we only consider 32 different sets of counts, namely $M^{\mu_1}_{\xi^a_A \xi^b_B}$ and $M^{\mu_2}_{\xi^a_A \xi^b_B}$. We unify the 32 sets of the counts into a vector $\boldsymbol{m}$, $\boldsymbol{m} = \{M^{\mu}_{\xi^0_A \xi^0_B}, M^{\mu}_{\xi^0_A \xi^1_B}, M^{\mu}_{\xi^0_A \xi^2_B}, M^{\mu}_{\xi^0_A \xi^3_B}, \ldots, M^{\mu}_{\xi^3_A \xi^3_B}\}$. $\boldsymbol{s}$ and $\boldsymbol{m}$ consist of the inputs of our classification model:

$$\boldsymbol{x} = \{\boldsymbol{s}, \boldsymbol{m}\}. \tag{10}$$

For each scenario with a specific x, we use discrete values to label the corresponding imperfection and normal types. These labels are the outputs of our classification model, denoted as

$$\boldsymbol{y}_c = \{\boldsymbol{y}_{\delta_1}, \boldsymbol{y}_{\delta_3}, \boldsymbol{y}_{\theta_1}, \boldsymbol{y}_{\lambda}, \boldsymbol{y}_{P_{\rm ap}}, \boldsymbol{y}_{\tau_{\rm dt}}, \boldsymbol{y}_{\Delta\eta}, \boldsymbol{y}_{\gamma}, \boldsymbol{y}_{\rm normal}\}, \tag{11}$$

where the first eight labels correspond to the types of imperfections in vector $\boldsymbol{i}$, and $\boldsymbol{y}_{\rm normal}$ represents the normal type without imperfections. In Appendix B, we introduce the definition of the normal class in detail.

3.2 Random forest algorithm

The random forest algorithm [38] is a popular and widely used machine learning technique for addressing classification and prediction problems. It is an ensemble-based learning method that works by constructing multiple decision trees, where each tree is first trained on a subset of the training data and a subset of the input features. The predictions of the individual trees are then combined to obtain a final prediction.

The random forest algorithm embodies outstanding performance in handling high-dimensional and complex datasets efficiently, avoiding overfitting by using multiple trees. It works by generating an initial large number of decision trees and selecting the best subset of trees based on their observed performance on the training data. To generate each decision tree, the algorithm uses a technique called "bagging" (bootstrap aggregating), which first involves resampling the training data with replacement to create multiple "bootstrap" samples. Each decision tree is subsequently trained on one of these bootstrap samples. Once all the decision trees have been constructed, the algorithm will predict by aggregating the predictions of all the individual trees. As is shown in Figure 3, in undertaking classification tasks, the most common aggregation technique is majority voting, where each tree's prediction is treated as a vote for a particular class, and the final prediction is the class that receives the most votes.

To evaluate the actual performance of the algorithm, the whole data set is typically split initially into two sets: a training set, which is used to train the individual decision trees, and a test set, which is deployed to evaluate the performance of the model. The algorithm can also be further tuned by adjusting hyperparameters such as the total number of trees, the maximum depth of each tree, and the size of the resultant random feature subset.

When training the model in the training set, input features and labels must be used, and the trained model is then tested on the test set. When the input characteristics of the test set are known, the machine learning model will generate an output label. In the classification model, the label represents the types

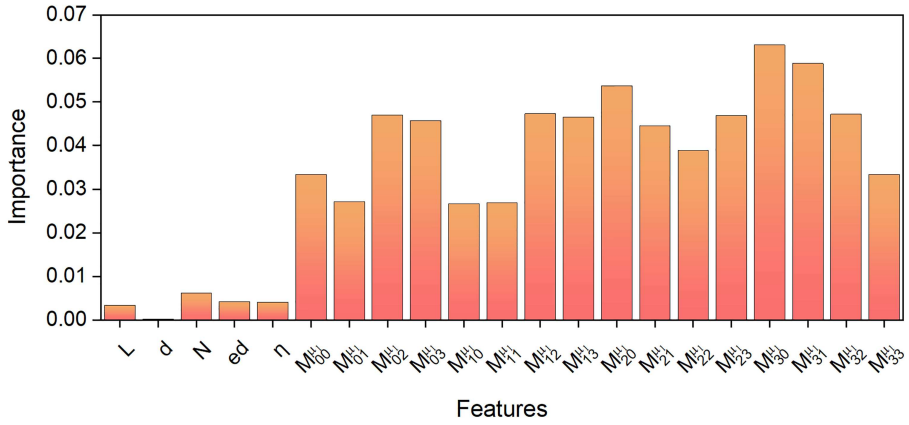

Figure 4 (Color online) Feature importance.

of imperfections and attacks in the QKD system, and in the prediction model, the label represents the specific value of the imperfections or attacks. By comparing those labels generated by the machine learning model with the actual labels in the test set, parameters such as accuracy can then be derived.

3.3 Model training and evaluations

In this subsection, we discuss how to construct the detection model based on the feature vectors.

Initially, we deploy the random forest classification algorithm. Our target here is to derive an output $\boldsymbol{y}$ according to the input $\boldsymbol{x}$ by first constructing a classifier, which is represented by a function f_1: $\boldsymbol{x} \rightarrow \boldsymbol{y}$. The classifier is constructed based on multiple training iterations on a training set. The input and output vectors of the model, therefore, constitute the whole data set, 80% of which is included as the training set and the remaining 20% as the test set.

As is shown in Figure 4, the abscissa represents the input features of our model, and the ordinate represents each feature's importance. Considering that the influence of M^{μ_2} is less than that of M^{μ_1}, we only list the influence of the system parameters and M^{μ_1} here. Feature importance is a measure of the contribution of each feature to the performance of a machine learning model. Generally, the higher the feature importance score, the more the feature contributes to the model's predictions. Through the deployed random forest algorithm, $M_{30}^{\mu_1}, M_{31}^{\mu_1}$, and $M_{20}^{\mu_1}$ are identified as the top three most significant features for predicting outputs. Moreover, the feature importance can help us understand how the model is making its predictions, which is important for both building trust and explaining the model's behavior. Finally, feature importance can be used to detect those outliers or anomalies that have a significant impact on the model's performance, which may then indicate errors in the data or problems with the model's assumptions.

A confusion matrix, which is also known as an error matrix, is a commonly used tool in the field of machine learning to evaluate the performance of classification models. It is a two-dimensional table where the rows represent the true classes, and the columns represent the predicted classes. It summarizes all the classification results and gives an intuitive view of the classifier's performance, and it can be implemented further to calculate various evaluation metrics, such as accuracy, precision, recall, and F1-score. These metrics can help us evaluate the performance of a classifier more thoroughly.

Precision measures the proportion of true positives (TP) among all predicted positives (TP + false positives, FP). It can be interpreted as the model's ability to accurately identify positive instances. A high precision means that the model is making very few false positive predictions,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{12}$$

where TP denotes the actual class is positive, the model predicts it as positive, and FP denotes the actual class is negative but the model predicts it as positive.

Recall rate, another evaluation indicator, is used to quantify the proportion of true positives among all actual positives (TP + false negatives, FN). It can be interpreted as the model's ability to identify all positive instances. A high recall rate means that the model is correctly identifying a large proportion of

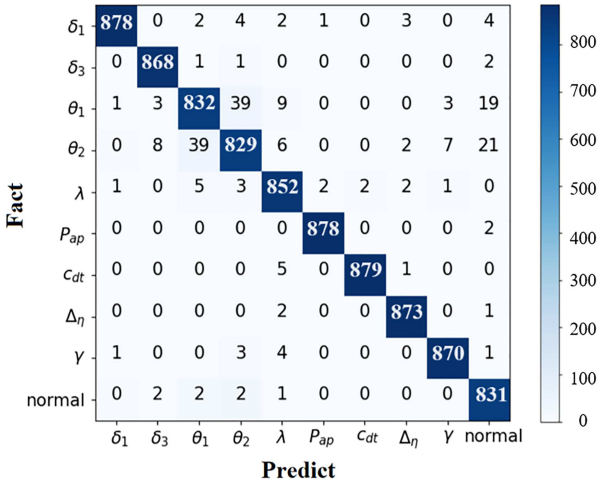

Figure 5 (Color online) Confusion matrix of random forest.

positive instances,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{13}$$

where FN denotes the actual class is positive but the model predicts it as negative. The F1-score is a harmonic mean of precision and recall, and is computed as

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{14}$$

It is a metric that takes both precision and recall rate into account, and is useful when we want to balance the trade-off between precision and recall rate. A high F1-score means that the model has both high precision and high recall rate. In summary, precision measures the accuracy of positive predictions, recall rate measures the completeness of positive predictions, and F1-score is a balance between precision and recall rate.

With a total of 291384 data sets, among them, 80% is for training, 20% is for testing, and the final testing accuracy is higher than 98%. In addition, Table 3 lists precision, recall rate, and F1-score for all imperfect types using the random forest algorithm.

4 Discussion and results

4.1 Classification

In addition to the random forest (RF) algorithm, we also employ other classification algorithms, such as the decision tree (DT) algorithm [28] and the K-nearest neighbors (KNN) algorithm [29], to detect imperfections and attacks in QKD systems; corresponding results are listed in Table 4.

The total number of the data set is 291384, 20% of which is the test set. Table 4 enumerates the accuracy of different algorithms and the time spent using them, among which the test time is the total time spent on all data in the test set. The RF algorithm has the highest accuracy, followed by the DT algorithm, but the RF algorithm spends more time on model training and testing. The DT algorithm is slightly less accurate than the RF algorithm but takes less time. Figures 5 and 6 show the confusion matrix of the three different algorithms. In general, the DT algorithm and the RF algorithm show better performance, whereas the KNN algorithm is the worst.

It is worth noting that once the model is trained, the time required for the test set is so short that it is almost negligible. If accuracy is the most concerning indicator, one can sacrifice a little training time and choose the RF algorithm, and if high speed is emphasized more, the DT algorithm can be chosen. Both the RF algorithm and the DT algorithm are tree structures. The RF is an integrated algorithm that synthesizes multiple decision trees and can avoid the overfitting of a single decision tree. In the following discussion, we will use the RF algorithm by way of illustration.

Table 3 Classification report

	Precision	Recall rate	F1-score
δ_1	1.00	0.99	0.99
δ_3	1.00	0.99	1.00
θ_1	0.91	0.95	0.93
θ_2	0.91	0.95	0.93
λ	0.99	0.99	0.99
$P_{\rm ap}$	1.00	1.00	1.00
$\tau_{\rm dt}$	1.00	1.00	1.00
$\Delta\eta$	1.00	1.00	1.00
γ	1.00	0.99	0.99
Normal	0.99	0.94	0.96

Table 4 Comparison of different algorithms (device information: 13th Gen Intel(R) Core (TM) i5-13400F 2.50 GHz)

	Training time (s)	Test accuracy	Test time (s)
RF	677.169	0.979	1.950
DT	46.910	0.975	0.027
KNN	245.677	0.961	62.650

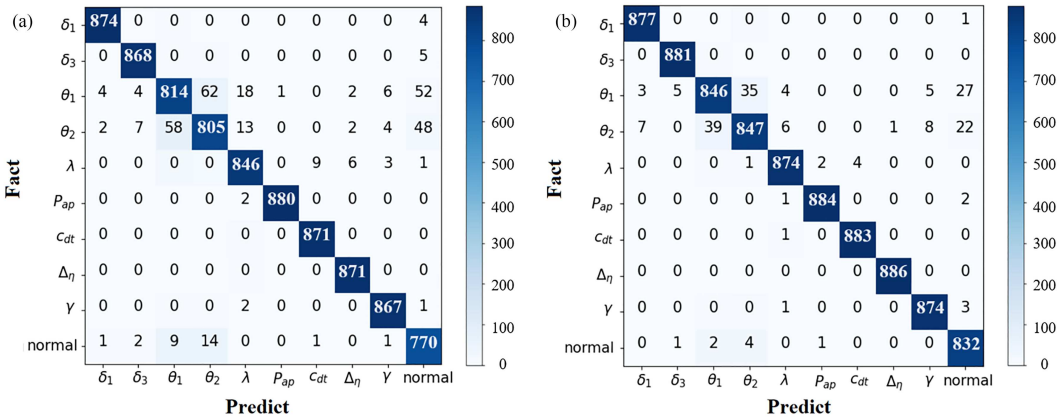

Figure 6 (Color online) Confusion matrices of (a) DT and (b) KNN.

4.2 Prediction

After judging the category of imperfections in the actual QKD system, to further inform the user about the imperfection detail and guide them to conduct a tighter security analysis and improve the security, we constructed an error prediction model using the RF algorithm f_2.

Similar to other classification models, $\boldsymbol{s}$ and $\boldsymbol{m}$ are the inputs of our prediction model. For each scenario with a specific x, we label the corresponding imperfection values. These labels are, therefore, the outputs of our prediction model, denoted as

$$\boldsymbol{y}_i = \boldsymbol{i}. \tag{15}$$

The prediction results are shown in Figure 7. We calculate the residuals of the values predicted by the RF algorithm and the true values in the test set and then visualize them. The abscissa in Figure 7 represents the value of the imperfection predicted by the RF algorithm and the ordinate represents the residuals of the value of the imperfection as predicted by the RF algorithm and the true value of the imperfection. The smaller the residuals, the more accurately the model predicts the true value. In addition, because we take into account the effects of statistical fluctuations, where the imperfection of the system is very small, it may be just the effects of normal channel statistical fluctuations.

Besides, we use R^2 [39] to evaluate the predictive model. R^2 (R squared, coefficient of determination) reflects the degree of interpretation of the predicted value to the actual value, and it characterizes the quality of the fit. It is closer to 1, the stronger the explanatory power of the variables of the equation to

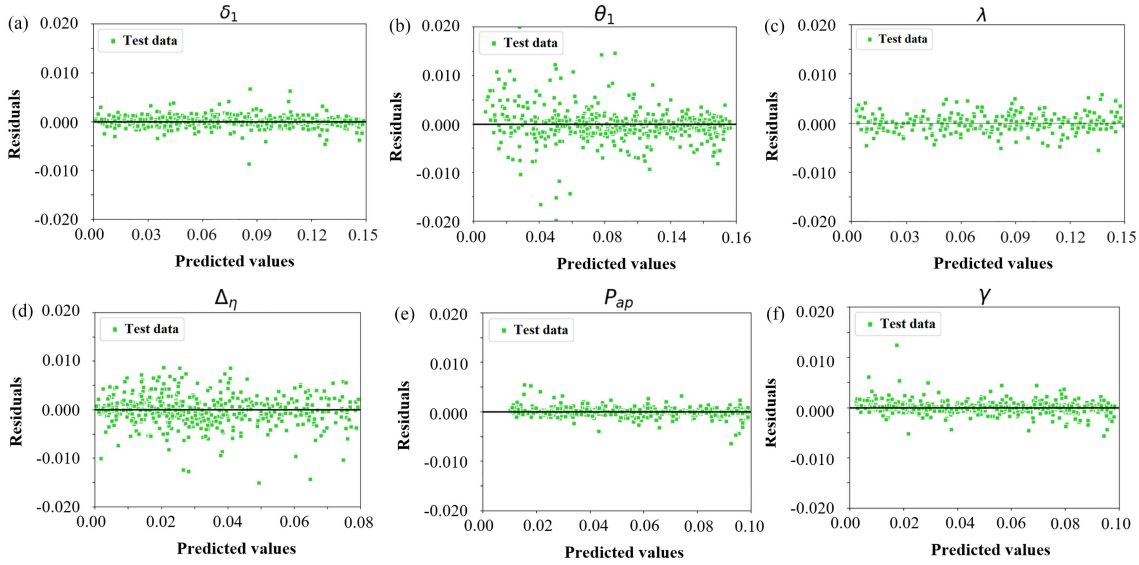

Figure 7 (Color online) Residuals of (a) state preparation error, (b) phase error, (c) intensity fluctuation, (d) detector efficiency mismatch, (e) after-pulse possibility, and (f) intercept-resend attack possibility.

Table 5 Parameter type

IM	PM	Detector	Attack
δ_1, δ_3, λ	θ_1, θ_2	$P_{\rm ap}$, $\tau_{\rm dt}$, $\Delta\eta$	γ

y, and the better the model fits the data,

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_i')^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{16}$$

where N is the number of samples, y_i is the actual value, y_i' is the predicted value, and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. In our model, R^2 of the phase error is about 98%, and R^2 of the other errors is greater than 99%, indicating that it is feasible to predict the value of the systematic error by using the RF algorithm, whose effect is very desirable.

The above discussion focuses on certain parameter errors, not so intuitive and visible for users. In the following, more attention is paid to locating the imperfect devices or eavesdropping, as shown in Table 5. Here, $\delta_1, \delta_2, \delta_3, \delta_4$, and λ are related to the IM, θ_1, θ_2 correspond to the PM, $P_{\rm ap}$, c_{DT}, and $\Delta\eta$ refer to the detector, and γ is located to the attack.

4.3 Multi-fault detection

In the above discussion, we only consider detecting the single fault scenario. Next, detecting multi-fault scenarios is taken into account. Instead of using one output, we use four outputs $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4\}$, representing IM, PM, Detector, and Attack, respectively, where $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4\} \in \{\text{Normal}, \text{Imperfection}\}$. We use the MultiOutputClassifier algorithm combined with the random forest algorithm. Our target is to derive an output $\boldsymbol{y}$ according to the input $\boldsymbol{x}$ by constructing a classifier, which is represented by a function $f_1' : \boldsymbol{x} \to \boldsymbol{y}$, where $\boldsymbol{y} \in \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4\}$. The confusion matrix of IM, PM, Detector, and Attack are shown in Figure 8.

Through this way, we can detect multiple faults simultaneously. The total amount of our data sets is 340038, among which 20% are the test set, with an accuracy of 98% on the test set. In other words, after detection by our classification model, 66410 groups are correctly detected in all four categories. There are 1469 groups with three categories of correct detection and one category of incorrect detection. There are only 126 groups with two categories of correct detection and 3 groups with one category of correct detection.

The Precision, Recall rate, and F1-score of these four classes are shown in Table 6.

We use the principal component analysis (PCA) [40] method to reduce the input features of the classifier as described in (10). The PCA is a commonly used dimensionality reduction technique that

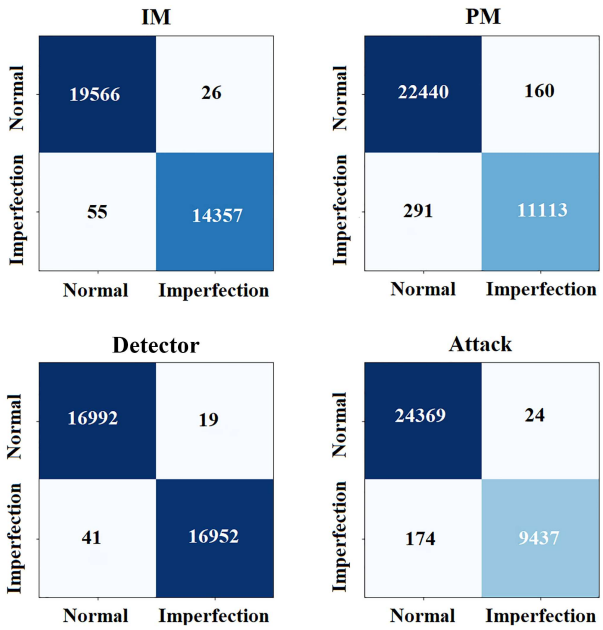

Figure 8 (Color online) Confusion matrices of of IM, PM, Detector, and Attack.

Table 6 Classification report

	Precision	Recall rate	F1-score
IM	1.00	1.00	1.00
PM	0.99	0.97	0.98
Detector	1.00	1.00	1.00
Attack	1.00	0.98	0.99

converts a high-dimensional dataset into a lower-dimensional representation while preserving the most important information. The PCA achieves this by projecting the original features onto a set of new orthogonal features known as principal components. Each principal component is a linear combination of the original features but ordered in terms of their importance in explaining the variance in the data. After calculation and comparison, the optimal number of principal components found in our model is 11; that is, if the number of principal components is reduced, the accuracy of the test set will be less than 97%. In contrast, if the number of principal components is increased, the accuracy of the model will not increase significantly, but rather, the time complexity will increase.

After using the PCA algorithm and the MultiOutputClassifier algorithm, the training set takes only 177.76 s, which is several times faster than not using the PCA method; the test set takes only 2.38 s, and the accuracy can then be as high as 98%.

5 Conclusion and outlook

In conclusion, for the first time, we propose to identify the imperfections of different devices and certain attacks in QKD systems and assess the value of these imperfections simultaneously using machine learning algorithms. It is very important to identify the imperfections of different devices and particular attacks because then legitimate users can either re-calibrate their devices or provide corresponding security analysis and then get the secure keys. First, we establish a mathematical model by taking into account both typical device imperfections and eavesdropper attacks. Second, we carry out corresponding numerical simulations and identify the imperfections of different devices and particular attacks in real time, with an accuracy of 98%.

Given that the accuracy of our present machine learning algorithm is not 100%, but 98%, this means that there is a 98% chance that it can be judged correctly and a 2% possibility that it can be misjudged in one test. We can implement the following strategy to ensure system security: we can continue monitoring with our present machine learning-based method N times, e.g., $N = 10$, where all 10 machine learning

judgments are independent. Then, we only retain those cases where all 10 machine learning judgments are the same, and the probability of this being the case is $(0.98^{10} + 0.02^{10} \approx)$ 81.7%, and the probability of all misjudgments is $(\frac{0.02^{10}}{0.98^{10}+0.02^{10}} \approx) 1.25 \times 10^{-17}$. The cost of reducing the misjudgment probability is to reduce the efficiency to 81.7% so that the security of the system can be improved. Of course, the above strategy is only an example, and it might actually make the whole process more complicated in real applications, so perhaps a more sophisticated strategy is required. That can constitute our future research work.

In all, our present work possesses the merits of resources, time-saving, and low cost. Most importantly, it does not need to interrupt the key transmission, compared with other classical security test methods, including using QBER testing, and other spectral or current testing with additional instruments. Therefore, our work is a more universal and useful method, which can provide a new solution for the practical security evaluation of QKD systems and, thus, pave the way for large-scale applications of quantum communication networks in the future.

It should be noted that here, we only use the phase-coding BB84 protocol as an example for illustration; however, this method is also applicable to other QKD protocols and coding schemes. In addition, our method, in principle, can also cover other imperfections and attacks that might not alter the counting rates of detectors but influence other observable parameters, e.g., spectrum, current, and light reflectivity, with the assistance of other measuring instruments and data. We need to obtain additional measurement data with current or voltage measuring devices. For example, for those side-channel attacks, such as time-shift attacks, fake-state attacks, and detector-blinding attacks [41–43], we need to obtain additional measurement data with current or voltage-measuring devices. For those distinguishable decoy states that are caused by imperfect modulation devices, we need to use a spectral measurement device to obtain the spectral information. That is to say, to give a more general model, we need to obtain additional data for training and testing.

Acknowledgements This work was supported by Industrial Prospect and Key Core Technology Projects of Jiangsu Provincial Key R&D Program (Grant No. BE2022071), National Natural Science Foundation of China (Grant Nos. 12074194, 12104240, 62101285), Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0300701), Natural Science Foundation of Jiangsu Province (Grant Nos. BK20192001, BK20210582), Natural Science Foundation of the Jiangsu Higher Education Institutions (Grant No. 21KJB140014), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. KYCX23_1030).

References

1 Scarani V, Bechmann-Pasquinucci H, Cerf N J, et al. The security of practical quantum key distribution. Rev Mod Phys, 2009, 81: 1301–1350
2 Gisin N, Ribordy G, Tittel W, et al. Quantum cryptography. Rev Mod Phys, 2002, 74: 145–195
3 Weedbrook C, Pirandola S, García-Patrón R, et al. Gaussian quantum information. Rev Mod Phys, 2012, 84: 621–669
4 Bennett C H, Brassard G. Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, 1984. 175–179
5 Xu F H, Ma X F, Zhang Q, et al. Secure quantum key distribution with realistic devices. Rev Mod Phys, 2020, 92: 025002
6 Sun S H, Huang A Q. A review of security evaluation of practical quantum key distribution system. Entropy, 2022, 24: 260
7 Wang X B. Decoy-state quantum key distribution with large random errors of light intensity. Phys Rev A, 2007, 75: 052301
8 Tamaki K, Curty M, Kato G, et al. Loss-tolerant quantum cryptography with imperfect sources. Phys Rev A, 2014, 90: 052314
9 Pereira M, Curty M, Tamaki K. Quantum key distribution with flawed and leaky sources. npj Quantum Inf, 2019, 5: 62
10 Huang J Z, Yin Z Q, Wang S, et al. Effect of intensity modulator extinction on practical quantum key distribution system. Eur Phys J D, 2012, 66: 1–5
11 Makarov V, Anisimov A, Skaar J. Effects of detector efficiency mismatch on security of quantum cryptosystems. Phys Rev A, 2006, 74: 022313
12 Zhao Y, Fung C H F, Qi B, et al. Quantum hacking: experimental demonstration of time-shift attack against practical quantum-key-distribution systems. Phys Rev A, 2008, 78: 042333
13 Sajeed S, Chaiwongkhot P, Bourgoin J P, et al. Security loophole in free-space quantum key distribution due to spatial-mode detector-efficiency mismatch. Phys Rev A, 2015, 91: 062301
14 Rogers D J, Bienfang J C, Nakassis A, et al. Detector dead-time effects and paralyzability in high-speed quantum key distribution. New J Phys, 2007, 9: 319
15 Weier H, Krauss H, Rau M, et al. Quantum eavesdropping without interception: an attack exploiting the dead time of single-photon detectors. New J Phys, 2011, 13: 073024
16 Fan-Yuan G J, Wang C, Wang S, et al. Afterpulse analysis for quantum key distribution. Phys Rev Appl, 2018, 10: 064032
17 Fung C H F, Qi B, Tamaki K, et al. Phase-remapping attack in practical quantum-key-distribution systems. Phys Rev A, 2007, 75: 032314
18 Xu F H, Qi B, Lo H K. Experimental demonstration of phase-remapping attack in a practical quantum key distribution system. New J Phys, 2010, 12: 113026
19 Huang A, Sun S H, Liu Z, et al. Quantum key distribution with distinguishable decoy states. Phys Rev A, 2018, 98: 012330
20 Zhang G, Primaatmaja I W, Haw J Y, et al. Securing practical quantum communication systems with optical power limiters. PRX Quantum, 2021, 2: 030304

21 Ponosova A, Ruzhitskaya D, Chaiwongkhot P, et al. Protecting fiber-optic quantum key distribution sources against light-injection attacks. PRX Quantum, 2022, 3: 040307
22 Li H W, Wang S, Huang J Z, et al. Attacking a practical quantum-key-distribution system with wavelength-dependent beam-splitter and multiwavelength sources. Phys Rev A, 2011, 84: 062308
23 Curty M, Lütkenhaus N. Intercept-resend attacks in the Bennett-Brassard 1984 quantum-key-distribution protocol with weak coherent pulses. Phys Rev A, 2005, 71: 062301
24 Sun S, Xu F. Security of quantum key distribution with source and detection imperfections. New J Phys, 2021, 23: 023011
25 Chen Y, Huang C, Chen Z, et al. Experimental study of secure quantum key distribution with source and detection imperfections. Phys Rev A, 2022, 106: 022614
26 Breiman L. Random forests. Machine Learn, 2001, 45: 5–32
27 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
28 Quinlan J R. Induction of decision trees. Mach Learn, 1986, 1: 81–106
29 Altman N S. An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat, 1992, 46: 175–185
30 Mao Y, Huang W, Zhong H, et al. Detecting quantum attacks: a machine learning based defense strategy for practical continuous-variable quantum key distribution. New J Phys, 2020, 22: 083073
31 Luo H, Zhang L, Qin H, et al. Beyond universal attack detection for continuous-variable quantum key distribution via deep learning. Phys Rev A, 2022, 105: 042411
32 Wang J, Liu H, Ma H, et al. Experimental study of four-state reference-frame-independent quantum key distribution with source flaws. Phys Rev A, 2019, 99: 032309
33 Lu F Y, Lin X, Wang S, et al. Intensity modulator for secure, stable, and high-performance decoy-state quantum key distribution. npj Quantum Inf, 2021, 7: 75
34 Rusca D, Boaron A, Grünenfelder F, et al. Finite-key analysis for the 1-decoy state QKD protocol. Appl Phys Lett, 2018, 112: 171107
35 Fei Y Y, Meng X D, Gao M, et al. Quantum man-in-the-middle attack on the calibration process of quantum key distribution. Sci Rep, 2018, 8: 4283
36 Lim C C W, Curty M, Walenta N, et al. Concise security bounds for practical decoy-state quantum key distribution. Phys Rev A, 2014, 89: 022307
37 Ding H J, Liu J Y, Zhang C M, et al. Predicting optimal parameters with random forest for quantum key distribution. Quantum Inf Process, 2020, 19: 1–8
38 Ho T K. Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995. 1: 278–282
39 Chicco D, Warrens M J, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci, 2021, 7: e623
40 Pearson K. LIII. On lines and planes of closest fit to systems of points in space. London Edinburgh Dublin Philos Mag J Sci, 1901, 2: 559–572
41 Tan H, Zhang W Y, Zhang L, et al. External magnetic effect for the security of practical quantum key distribution. Quantum Sci Technol, 2022, 7: 045008
42 Lu F Y, Ye P, Wang Z H, et al. Hacking measurement-device-independent quantum key distribution. Optica, 2023, 10: 520–527
43 Ye P, Chen W, Zhang G W, et al. Induced-photorefraction attack against quantum key distribution. Phys Rev Appl, 2023, 19: 054052

Appendix A Simplification of $P^{\mu}_{\xi_A^a \xi_B^b}$

The details of (2) are shown in this part for better understanding and simplification. $P^{\mu}_{\xi_A^a \xi_B^b}$ is the probability when the corresponding measurement result is b after Bob uses the basis measurement $\xi_B \in \{Z, X\}$ under the condition that Alice prepares the quantum state $\xi_A^a \in \{\phi_{0Z}, \phi_{1Z}, \phi_{0X}, \phi_{1X}\}$. We rewrite (2) from the main text here,

$$P^{\mu}_{\xi_A^a \xi_B^b} = \sum_{n=0}^{\infty} P_n(\mu) \sum_{j=0}^{n} C_n^j \eta^j (1-\eta)^{n-j} (|\langle \xi_A^a | \xi_B^b \rangle|^2)^j F(j). \tag{A1}$$

Let us first think about the case where $a = 0$ and $b = 0$, 1, 2 or 3. That is, Alice sends the $|0\rangle$ state, and Bob gets $|0\rangle, |1\rangle$ state by Z-basis measurement or $|2\rangle, |3\rangle$ state by X-basis measurement.

$$\begin{aligned} P^{\mu}_{\xi_A^0 \xi_B^0} &= \sum_{n=0}^{\infty} P_n(\mu) \sum_{j=0}^{n} C_n^j {\eta_0}^j (1-\eta_0)^{n-j} (|\langle \xi_A^0 | \xi_B^0 \rangle|^2)^j F(j) \\ &= \sum_{n=0}^{\infty} P_n(\mu) \sum_{j=1}^{n} c_n^j \eta_0^j (1-\eta_0)^{n-j} \left(\cos^2 \frac{\delta_1}{2}\right)^j (1-d) \times (1+P_{\mathrm{ap}}) + \sum_{n=0}^{\infty} P_n(\mu)(1-\eta)^n d(1-d) \\ &= \mathrm{e}^{-\mu\eta_0}(1-d)\left(\mathrm{e}^{\mu\eta_0 \cos^2 \frac{\delta_1}{2}} - 1 + d\right)(1+P_{\mathrm{ap}}) - P_{\mathrm{ap}} d(1-d)\mathrm{e}^{-\mu\eta_0}, \end{aligned} \tag{A2}$$

$$P^{\mu}_{\xi_A^0 \xi_B^1} = \mathrm{e}^{-\mu\eta_1}(1-d)\left(\mathrm{e}^{\mu\eta_1 \cos^2 \frac{\delta_1}{2}} - 1 + d\right)(1+P_{\mathrm{ap}}) - P_{\mathrm{ap}} d(1-d)\mathrm{e}^{-\mu\eta_1}, \tag{A3}$$

$$P^{\mu}_{\xi_A^0 \xi_B^2} = \mathrm{e}^{-\mu\eta_0}(1-d)\left[e^{\frac{\mu\eta_0(\cos\frac{\delta_1}{2} + \sin\frac{\delta_1}{2})^2}{2}} - 1 + d\right] \times (1+P_{\mathrm{ap}}) - P_{\mathrm{ap}} d(1-d)\mathrm{e}^{-\mu\eta_0}, \tag{A4}$$

$$P^{\mu}_{\xi_A^0 \xi_B^3} = \mathrm{e}^{-\mu\eta_1}(1-d)\left[e^{\frac{\mu\eta_1(\cos\frac{\delta_1}{2} - \sin\frac{\delta_1}{2})^2}{2}} - 1 + d\right] \times (1+P_{\mathrm{ap}}) - P_{\mathrm{ap}} d(1-d)\mathrm{e}^{-\mu\eta_1}. \tag{A5}$$

Similarly, the other cases are the same.

Appendix B Statistical fluctuation

Let us assume $X_1, X_2, \ldots, X_N$ is a set of Bernoulli random variables that satisfy independent and identically distributed conditions, $P(X_i = 1) = p_i$ and the expected values and variances of $X := \sum_{i=0}^{N} X_i, X$ are $m := E(X), \sigma^2 := \mathrm{Var}[X]$. Record each observation result as x. When $N \to \infty$, $\frac{x-m}{\sigma}$ is close to the standard normal distribution $N(0,1)$.

The central limit theorem gives the upper and lower bounds of P_χ.

$$P_\chi(1-\beta_\chi) \leqslant P_\chi \leqslant P_\chi(1+\beta_\chi), \tag{B1}$$

where

$$\beta_\chi = \frac{r}{\sqrt{N_\chi P_\chi}}. \tag{B2}$$

In (B2), r represents the number of standard deviations and is directly related to the probability of failure in safety analysis. Both satisfy the following relation:

$$1 - \mathrm{erfc}\left(r/\sqrt{2}\right) = \varepsilon, \tag{B3}$$

where $\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2}\,\mathrm{d}t$ is error function. ε is failure probability, which we set to 10^{-7}. After calculation, $r \approx 5.3$ can be obtained.

To sum up, assuming that the quantum channel fluctuations conform to the Gaussian distribution, we can obtain observably measured $M^{\mu}_{\xi_A^a \xi_B^b}$ affected by statistical fluctuations, which have the upper bound $M^{\mu,U}_{\xi_A^a \xi_B^b}$ and the lower bound $M^{\mu,L}_{\xi_A^a \xi_B^b}$. They satisfy

$$M^{\mu,L}_{\xi_A^a \xi_B^b} < M^{\mu}_{\xi_A^a \xi_B^b} < M^{\mu,U}_{\xi_A^a \xi_B^b}. \tag{B4}$$

Under the ideal condition, that is, without any imperfections and attacks, we will get 32 groups of M_0^L and M_0^U by the influence of the quantum channel fluctuations. When there are imperfections and attacks, we will get 32 groups of M. If the 32 groups of M are all between M_0^L and M_0^U, we consider it normal. It can be understood that the imperfection in this situation is very small which can be attributed to normal channel statistical fluctuations.