IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 16, 2020, accepted August 30, 2020, date of publication September 7, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022098

Testing Randomness Using Artificial Neural Network

YULONG FENG AND LINGYI HAO

State Key Laboratory of Advanced Optical Communication, Systems and Networks, Department of Electronics, and Center for Quantum Information Technology, Peking University, Beijing 100871, China

Corresponding author: Yulong Feng (fengyulonglx@gmail.com)

ABSTRACT The vital characteristic of randomness is unpredictability. Thus any regularity will compromise the application of random numbers. Quantum random number generators (QRNGs) can provide intrinsic unpredictable randomness based on the nature of quantum physics, while pseudo-random number generator can not due to the origination of deterministic algorithms or physical processes. However, commonly used traditional test suits are rigorously to test the statistical properties, and the unpredictability of random numbers is still difficult to test. To verify which sources of random numbers are truly unpredictable, in this paper, we propose a new randomness testing method with the artificial neural network (ANN). Random number sequences generated by four different kinds of sources are tested, which are the natural number π, the linear congruence generator (LCG) pseudo-random algorithm, the Mersenne Twister (MT) pseudo-random algorithm and the QRNG based on vacuum noise, respectively. The testing results indicate that the random sequences from natural number π and LCG fail to pass our randomness test, while the other two kinds of sequences pass the test successfully due to the relatively simple structure of our ANN. As the complexity of the ANN and the amount of computing power involved increasing, this test method has potential to predict the MT random numbers, and it is also expected that the quantum random numbers can not pass the test with unpredictability no matter how complex the ANN is.

INDEX TERMS Randomness, test, quantum random number generator, pseudo-random, artificial neural network.

I. INTRODUCTION

Random numbers play an irreplaceable role in many fields, such as biology [1], [2], economics [3], [4], and computer science [5]. Even in the fields like cryptography [6], [7], unpredictable random sequences are an indispensable part to ensure the security of the entire crypto system [8], [9]. In the broadest sense, random numbers are generated in two ways. One is complex algorithms based on the computer, which aims at utilizing pseudo-random algorithms to extend short randomness seed. However, this method faces theoretical predictability due to the determinacy in the algorithms. The other is based on the physical processes in which the uncertainty is attempted to be discovered and used for generating random numbers and is usually called physical random number generator (PRNG). Furthermore, according to the intrinsic properties of physical processes, PRNG can be divided into two categories: classical-physics-based methods and quantum-physics-based methods [10]. Classical-physics-based random number generators, such as [11]–[14], are always based on complex classical physical processes, which are in principle predictable if we know all details of the system and thereby are also deterministic. On the contrary, quantum-physics-based random number generators, which are also called Quantum random number generators (QRNGs) [15], [16], are based on quantum physics processes, such as position and momentum of single photon [17]–[19], laser phase noise [20]–[23], and vacuum shot noise [24]–[27], which are the genuine unpredictable processes to our knowledge. Therefore, it is believed that this kind of random number generator can provide true randomness.

Besides the generation of randomness, randomness tests are also significantly important to verify the quality of the random numbers. However, due to the fact that there is no rigorous definition of randomness at present, the randomness tests employed by people are still constituted by deterministic algorithms, such as DIEHARD [28], and NIST-STS [29].

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

They can be very effective to assess the randomness from algorithms, while the purpose of which is to estimate whether the randomness generated by the pseudo-random algorithm satisfies a certain statistical distribution and application. So, conventional testing methods are not enough to objectively understanding the ideal randomness and optimizing the generation of random sequences, especially for the random sequences generated based on the principles of quantum mechanics. Even if the device independent (DI) method can quantify ''quantumness'', there is still no valid way to distinguish random numbers generated by quantum physical processes and other processes.

With the development of computer science and information theory, people have proposed artificial neural network (ANN) as a powerful computing tool [30]–[33]. From the viewpoint of information processing, ANN establishes a simple model based on the human brain neural network and forms different networks according to different connection methods aiming to solve discrete classification problems. It plays an important role in the fields of computer science [34] and biology [35]–[37] as a powerful computing tool. It is also used for data processing of quantum information [38], [39]. Furthermore, based on the universal approximation theorem [40], the pseudo-random sequence can be recognized theoretically by training ANN to simulate the characteristics of pseudo-random sequence algorithms. Thus, we try to employ ANN to approximate pseudo-random algorithms to realize the prediction of pseudo-random sequences.

In this article, we use a four layers ANN based on the Levenberg-Marquart back propagation algorithm to test four different kinds of random sequences. Statistical fluctuations calculated from the predicted success rate are employed as the reference indicators and limits in the testing [41], [42]. The experimental results illustrate that ANN can approximate some algorithms after it is trained using corresponding random sequences generated by the algorithm. However, for some complex pseudo-random algorithms, ANN also needs complex construction to be effective. In addition, it is proved that there is no fixed pattern and commonality in quantum random numbers based on ANN testing.

II. WHY USING ARTIFICIAL NEURAL NETWORKS

According to the universal approximation theorem, the standard multilayer feed-forward networks with a single hidden layer which contains finite hidden neurons and arbitrary activation function are universal approximators for $C(R_m)$. Theorem universal approximation theorem can be expressed in mathematical form:

Let φ be an arbitrary activation function and $X \subseteq R^m$ and X is compact. The space of continuous function on X is denoted by $C(X)$. Then $\forall f \in C(X), \forall \varepsilon > 0 : \exists n \in N,\ a_{ij}, b_i, w_i \in R,\ i \in \{1 \ldots n\},\ j \in \{1 \ldots m\}$:

$$(A_n f)(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{n} w_i \varphi \left(\sum_{j=1}^{m} a_{ij} x_j + b_i \right). \quad (1)$$

An approximation of the function f means:

$$\|f - A_n f\| < \varepsilon. \quad (2)$$

In the notation $A_n f$, n represents the number of hidden neurons.

The theorem implies when the network is sufficiently complex (i.e. contains enough hidden neurons), it can approximate any algorithm based on finite continuous function with arbitrary precision, which means that we can use ANN to reach the limit of an algorithm, theoretically. Therefore, we tend to utilize ANN to approximate pseudo-random algorithms and distinguish the random sequences generated by pseudo-random algorithms and quantum physical processes and even realize the prediction of random sequences generated by pseudo-random algorithms.

In addition, when we tend to predict the random numbers generated by the pseudo-random algorithms, we actually perform the inverse operation of the algorithm and solve some challenging mathematical problems. Such reverse problem is not only difficult to find the answer, but also requires significant computing resources. However, based on ANN, the pseudo-random algorithm is actually approached in the forward direction. The difficulty of the problem depends on the complexity of the pseudo-random algorithm and the computing power of the ANN. Currently, ANN is one of the most powerful computing tools. By adjusting the number of layers, the number and weight of nodes, the transfer functions, a large amount of computation and complexity can be associated with ANN. Therefore, using ANN to test random sequences can theoretically detect random sequences generated by different complexity algorithms.

Finally, one of the major properties of the random sequences is unpredictability, which means that the prior probability should be equal to the posterior probability. For a binary sequence, the prior probability is 0.5. If the posterior probability exceeds the confidence interval of 0.5, the unpredictability condition could not be established and the random number is not safe. In this paper, the prediction of random sequences generated by quantum physics process is implemented based on the principle and verifying the prior probabilities and posterior probabilities of the prediction are both 0.5 with ANN, which indicates the unpredictability of the sequence. Furthermore, as one of the foremost methods of supervising the learning process in machine learning, ANN provides the right and wrong indication in the process of machine learning. The ANN continuously reduces the error between the inferred data and the real data and simulate the precise function or algorithm to achieve the purpose of prediction or classification. Therefore, the ANN can judge if the prior probability and the posterior probability of the test

sequence are consistent with the ideal sequence and realize the test for the random sequence.

Consequently, a four-layer ANN is established to test the randomness. On the one hand, based on the universal approximation theorem, we hope the ANN can be used to approximate the pseudo-random algorithm to recognize the difference between the random sequence generated by QRNG and the pseudo-random algorithms and even realize the prediction of the random sequence generated by the pseudo-random algorithm. On the other hand, with formidable computing power and analytical ability of ANN, we expect that it can single out the random sequence generated by quantum physics and verify their ideal randomness.

III. RANDOM NUMBER GENERATORS

The experiment is carried out using four kinds of random sequences generated by nature number π, linear congruence generator (LCG) pseudo-random algorithm, Mersenne Twister (MT) pseudo-random algorithm and vacuum shot noise, respectively. LCG and MT are both famous and wildly used pseudo-random algorithms.

A. PSEUDO-RANDOM NUMBER GENERATOR

Random numbers generated from two of the most representative pseudo-random number algorithms LCG and MT are utilized to test the method. In the experiment, The C programming language is actually used to generate random sequences based on the LCG pseudo-random algorithm and MATLAB is used to generate random sequences based on the MT pseudo-random algorithm.

Based on the factorization problem, the LCG algorithm represents a class of random number generators based on linear congruence algorithm, which is mainly used in various programming languages such as C language, C++, and java to produce random seeds. The basic principle of LCG is the formula:

$$X_{n+1} = (aX_n + c) \bmod m, \tag{3}$$

where $X = \{X_1, X_2, \ldots, X_n\}$ are the random sequence. X_0 $(0 \leq X_0 < m)$ is the random seed of LCG. m $(0 < m)$ is called the modulus. a $(0 < a < m)$ is called the multiplier. And c $(0 < c < m)$ is called increment. If $c = 0$, the random number generator is also called a multiplicative congruential generator. If $c \neq 0$, it is also called a mixed congruential generator. LCG is extremely sensitive to the selection of parameters a and m. There are generally 3 types of parameter selection: (1) m is prime, (2) m is power of 2 and $c = 0$, (3) $c \neq 0$. Here we employed the C programming language to generate random numbers and its parameters showed in Table. 1.

MT is based on the twisted generalised feedback shift register, which is widely used in MATLAB, MATHMATIC, EXCEL and PYTHON, representing a class of algorithms based on linear feedback shift. It is the pseudo-random algorithm widely used in experiments in various scientific fields. Its name derives from the fact that its period length is chosen to be the Mersenne prime and the most commonly used version of the Mersenne Twister algorithm is based on the Mersenne prime $2^{19937} - 1$. The standard implementation of that called MT19937. In a 32-bit system, each random number generated by it is 32-bit stored and is called MT19937-32. In a 64-bit system, each number of a random sequence is 64-bit stored and is called MT19937-64.

TABLE 1. Parameters employed by LCG and MT to generate random sequences.

PRNG	Parameter	Value
LCG	m	2^{32}
	a	22695477
	c	1
MT	(w, n, m, r)	$(64, 312, 156, 31)$
	a	$B5026F5AA96619E9_{16}$
	(u, d)	$(29, 5555555555555555_{16})$
	f	6364136223846793005
	(s, b)	$(17, 71D673FEDA60000_{16})$
	(t, c)	$(37, FFF7EEE0000000000_{16})$
	l	43

The algorithm for MT is based on matrix linear recursion in a binary finite field F_2. The basic idea is to define a series $\{x_i\}$ through a simple recurrence relation and then output numbers x_iT, where T is an invertible F_2 matrix called a tempering matrix.

The first step is to define $\{x_i\}$. MT implements a random sequence of length n which is input by a simple recursive algorithm. Here we could assume each number in the sequence is stored in w-bits and it is necessary to give an ideal random number x_0 in advance as a seed. Furthermore, we can get the following numbers $\{x_i; i = 1, 2, \ldots, n-1\}$ as our input according to the recursive formula.

$$x_i = f \times (x_{i-1} \oplus (x_{i-1} \rightarrow (w-2))) + i, \tag{4}$$

where constant f is a parameter for MT generator. $\rightarrow$ denotes the bits of random number is shifted to the right and $\oplus$ denotes the XOR operation.

The second step is calculating the matrix T. MT produces an output of the twisted generalized feedback shift register (twisted GFSR, or TGFSR). For a number of w-bit, the MT will produce a random number in the range $[0, 2^{w-1}]$. The series x is defined as a series of w-bit quantities with the recurrence relation.

$$x_{k+n} := x_{k+m} \oplus \left(\left({x_k}^u || {x_{k+1}}^l\right) A\right) \quad k = 0, 1, \ldots, \tag{5}$$

where n is degree of recurrence. m $(1 \leq m < n)$ is called middle word and is an offset used in the recurrence relation defining the series. $||$ denotes concatenation of bit vectors (with upper bits on the left). $\oplus$ denotes XOR. ${x_k}^u$ means the upper $w-r$ bits of x_k and ${x_{k+1}}^l$ means the lower r bits of x_{k+1}. The twist transformation A is defined in rational normal form

as:

$$A = \begin{pmatrix} 0 & I_{\omega-1} \\ a_{\omega-1} & (a_{\omega-2}, \ldots, a_0) \end{pmatrix}, \tag{6}$$

where $\{a_i; i = 0, 1, \ldots, \omega\}$ are coefficients of the rational normal form twist matrix and I_{n-1} is a $(n-1) \times (n-1)$ identity matrix.

The tempering is defined in the case of Mersenne Twister as:

$$y := x \oplus ((x \to u) \,\&d), \tag{7a}$$
$$y := y \oplus ((x \leftarrow s) \,\&b), \tag{7b}$$
$$y := y \oplus ((y \leftarrow t) \,\&c), \tag{7c}$$
$$z := y \oplus (y \to l), \tag{7d}$$

where x is the next value from the series. y is a temporary intermediate value. z is the value returned from the algorithm with $\leftarrow$ and $\to$ denotes the bitwise left and right shifts. & denotes the bitwise AND. b and c is called tempering bit-masks. b and d is called tempering bit shifts. (u, d, l) is called additional Mersenne Twister tempering bit shifts or masks. For the property of TGFSR, $s + t \geq \left\lceil \frac{\omega}{2} \right\rceil - 1$ is required to reach the upper bound of uniform distribution for the upper bits.

Our experiment employ a random sequence generated by 64-bit MATLAB under a 64-bit system, namely MT19937-64. Its parameters are shown in Table. 1.

B. QUANTUM RANDOM NUMBER GENERATOR

Vacuum shot noise comes from vacuum fluctuations is an quantum phenomenon. The vacuum state can be represent in the quadrature:

$$|0\rangle = \int_{-\infty}^{\infty} \psi(x) |x\rangle \, dx, \tag{8}$$

where $|x\rangle$ are the amplitude quadrature eigenstates $\left(\langle x \mid x' \rangle = \delta\left(x - x'\right)\right)$ and $\psi(x)$ is the ground-state wavefunction, which is a Gaussian function centred around $x = 0$. The measurement of the amplitude quadrature collapses the wavefunction into quadrature eigenstates, and the associated outcomes being unpredictable but biased according to the Gaussian probability function $|\psi(x)|^2$. Then we can use the measurement to produce random numbers by dividing the Gaussian distribution as equal parts [24].

Since the vacuum shot noise is small, it is generally measured by a balanced homodyne detector which can amplify the quadrature of a state and its structure is shown in Fig. 1. The CW beams emitted by the laser diode enter one input port of the 50:50 BS. The other input port of the BS is blocked to provide the vacuum state. And a following measurement operation is realized by a homodyne detector and an ADC. The measurement result is finally processed by a randomness extractor to extract the final random bits.

Homodyne detector is the core of the system. Taking the annihilation operator $\hat{a}$ as an example, the variation process of components in homodyne detection can be calculated.

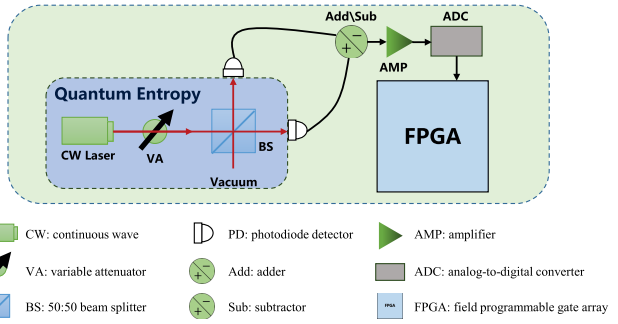


FIGURE 1. Experimental illustration of real-time optical QRNG based on vacuum fluctuation. A 1550-nm fiber-coupled laser (The model is NKT Basic E15 and linewidth is 100 Hz) serves as the local oscillation (LO) and is connected to one input port of the 50:50 beam splitter, while the other input port is blocked to provide the vacuum state. Two output ports of the beam splitter are optically coupled to two input ports of a balanced homodyne detector (The model is Thorlabs PDB480C and measurement bandwidth are limited to 1GHz with low-pass filter). The measurement results of the balanced homodyne detector are finally sampled by a 12-bit ADC (The model is ADS5400 and sampling frequency is 1 GHz and input voltage range is 1.5 V_{PP}) to acquire the raw data in real-time. A following randomness extractor based on the optimized algorithm is used to perform extraction simultaneously with raw data acquiring.

Assuming that the annihilation operator of the vacuum state is $\hat{a}$, the annihilation operator of the intrinsic light is $\hat{a}_{LO}$. After the beam splitter, the two annihilation operators are respectively:

$$\hat{a}_1 = \hat{a} - \hat{a}_{LO}, \tag{9a}$$
$$\hat{a}_2 = \hat{a} + \hat{a}_{LO}. \tag{9b}$$

And the operators $\left(\hat{a}, \hat{a}^\dagger\right)$ can be expressed as mechanical quantity operators $\left(\hat{x}, \hat{p}\right)$:

$$\hat{x} = \frac{\left(\hat{a} + \hat{a}^\dagger\right)}{\sqrt{2}}, \tag{10a}$$
$$\hat{p} = \frac{-i\left(\hat{a} - \hat{a}^\dagger\right)}{\sqrt{2}}. \tag{10b}$$

After the detection, the difference between the two signals reflects the difference in the number of photons:

$$\Delta\hat{n} = \hat{n}_2 - \hat{n}_2 = \hat{a}_2^\dagger \hat{a}_2 - \hat{a}_1^\dagger \hat{a}_1. \tag{11}$$

Note that the number of photons is proportional to the photocurrent. Then bring (9) and (10) into (11), we can get:

$$j \propto \Delta\hat{n} = \frac{|a_{LO}|}{\sqrt{2}} \left(\hat{x}\cos\theta + \hat{p}\sin\theta\right). \tag{12}$$

Note that calculations about creation operator are similar. Therefore, a signal positively correlated with the vacuum fluctuation can be obtained by Homodyne detection, and the random sequence can be obtained.

The random sequence we used in the experiment can pass the NIST-STS test packet, but the test packet is mainly for random sequences generated based on pseudo-random algorithms to assess whether the randomness is acceptable. We also employed other three kinds of random numbers

to implement the test which constructed by ANN to verify whether it can cognize the difference between random sequences generated based on quantum physical processes and pseudo-random algorithms, include these can pass the NIST-STS test.

IV. EXPERIMENT SCHEME

In this article, an ANN is employed to predict four different kinds of random sequences and we utilize statistical fluctuations about prediction success rate to be the test standard. Four kinds of random sequences come from the earliest randomness source natural number π, two kinds of pseudo-random number generator which are most widely known but based on different principles, i.e, LCG which is realized by C++ language and MT which is realized by software MATLAB, and the quantum random number generator based on vacuum fluctuation noise.

The ANN we employ here is a four layers Levenberg-Marquart back propagation algorithm network with the configuration of 6-30-20-2 based on Pytorch. We use ReLU as the activation function, calculate the error by CrossEntropy-Loss, use Adam as the optimizer, construct a classification problem by using the first six of the sequence as input, and use the Softmax function to process the output result to obtain the probability distribution determination result. The length of the test sequence is chosen to be seven, which covers the different signal segments resulted from the sampling of ADC of QRNG. Through tests and adjustment, the learning rate is set to 0.001 ensuring that the network will not be unstable due to the high learning rate and will not converge because the learning rate is too small. The number of training repetitions is 40, which also ensures the stability of the network.

We intend to predict the seventh number from its precedent consecutive six numbers by setting the input as the precedent consecutive six numbers and the output as the prediction value. The ANN actually outputs the scores about 0 and 1 that can be translate to the probability by formula $p(1) = \frac{e^a}{e^a+e^b}$ and $p(0) = \frac{e^b}{e^a+e^b}$. If we assume the number 1 gets a score and number 0 gets b score, the one with the highest score (predicted probability) would be chosen as the final output.

A. PRELIMINARY EXPERIMENT FOR RANDOM NUMBER GENERATED FROM π

For the natural number π, which is actually one of the first randomness sources utilized by people, each decimal is binarized with a threshold of 5 to obtain a binary sequence P, which is shown in Fig. 2. Seven digits are one training or test instance. In order to ensure parameters of the network are proper, we conduct a preliminary test of the sequence P.

The 1st to 40006th digits of P are used as the 40,000 instances for network training, and then 100,001th to 1,000,006th digits of P are used as 900,000 instances as Test 1 data set, in which each 100,000 cases are a test subset. The 999,001th to 9,999,006th digits of P are 900,000 instances for Test 2 data set, where each 1,000,000 instances are a test subset. The prediction success rate is the number of successes

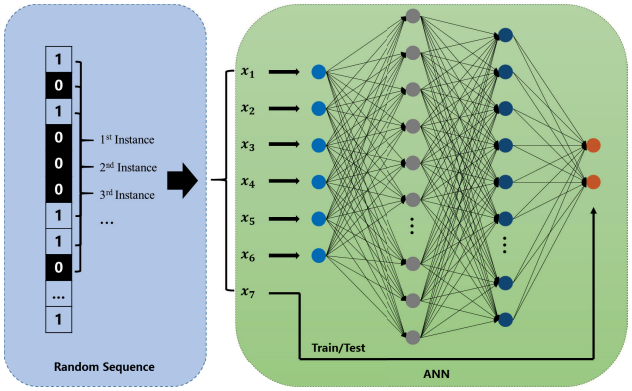


FIGURE 2. The method obtaining the training and test instances and the structure of ANN. Each training and test instance contains seven random numbers, which are intercepted from the original random sequence without overlapping. Among them, six digits are used as the input of the network and the last digit is used as the real output of the network, which is compared with the predicted output of the network to obtain the prediction success rate of the network.

divided by the total number of trials. In order to ensure the stability of the network and the reliability of the results, the experiment is repeated ten times, and the results are shown in Fig. 4.

For the test results, we can use 3σ and 5σ bound to verdict them. A random variable $Y = \frac{\sum_{k=1}^{n} X_k}{n}$ can be defined in the test, where the random variable $X_k = 1$ when the prediction is correct, and otherwise $X_k = 0$. Ideally, the probability of success for each prediction is 50% and is not related to each other. So, the mean of Y can be derived from following formula:

$$\begin{aligned} E(Y) &= \sum_{i=0}^{n} C_n^i \left(\frac{1}{2}\right)^n \left(\frac{i}{n}\right) \\ &= \sum_{i=1}^{n} \frac{(n-1)!}{(n-i)!(i-1)!2^n} \\ &= \sum_{i=0}^{n-1} \frac{C_{n-1} i - 1}{2^n} \\ &= \frac{1}{2}, \end{aligned} \tag{13}$$

and its variance is:

$$\begin{aligned} E\left(Y^2\right) &= \sum_{i=0}^{n} \left(\frac{i}{n}\right)^2 \frac{C_n}{2^n} \\ &= \sum_{i=1}^{n} \frac{i(n-1)!}{2^n n(i-1)!(n-i)!} \\ &= \sum_{i=1}^{n} \frac{(n-1)!}{2^n n(i-1)!(n-i)!} + \sum_{i=2}^{n} \frac{(n-1)(n-2)!}{2^n n(i-2)!(n-i)!} \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{C_{n-1} i - 1}{2^n} + \frac{n-1}{n} \sum_{i=2}^{n} \frac{C_{n-2} i - 2}{2^n} \\ &= \frac{1}{4n} + \frac{1}{4}. \end{aligned} \tag{14}$$

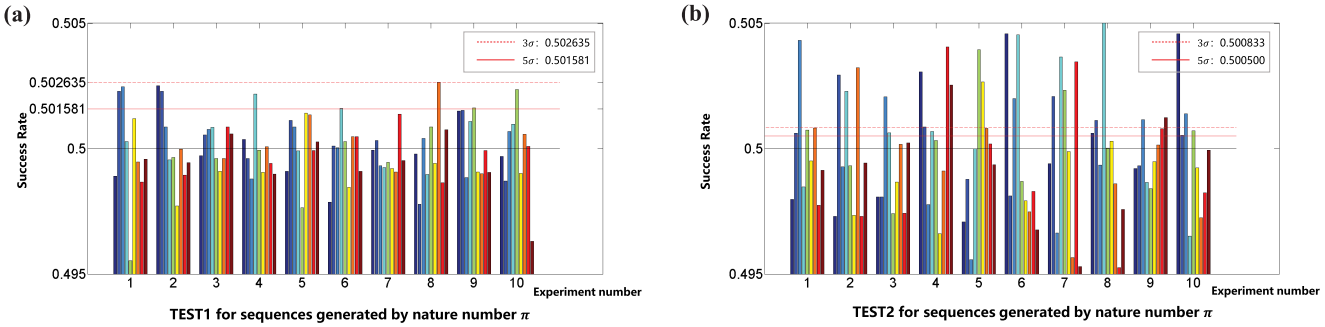


FIGURE 3. Preliminary test results for the sequence P generated by random source natural number π. (a) Test results for the TEST 1. The abscissa is the repeated number of experiments and the ordinate is the success rate. The red lines are 3σ and 5σ bound respectively. Each experiment contains nine test subsets, which is comprised of 100,000 cases and one case includes 7 digits. Each set of experiments is repeated 10 times to ensure the stability of the network prediction results. The prediction success rate is obtained by dividing the number of successful predictions by the total number of experiments. (b) Test results for the TEST 2. It includes nine test subsets, which consist of 1,000,000 cases. It can be seen that with the increase of test data, the prediction success rate gradually exceeds the 3σ and 5σ bound, which also means that the randomness of the random sequence is deficient.

According to the central limit theorem, the probability of success after n predictions should follow the Gaussian distribution $\left(0.5, \frac{1}{2\sqrt{n}}\right)$. Above 3σ bound indicates the prediction success rate exceeds $0.5 + 3\sigma$, which is a small probability event and means we have a 99.73% confidence that the ANN learned something from the training set. The sequence is considered to be non-random and can be predicted.

It is not difficult to figure out from the above results that most of the subsets of the P sequence TEST 1 exceed 3σ, some exceed 5σ, and most of the subsets of TEST2 exceed 5σ. Therefore, it can be reasonably considered that the ANN is valid for the sequence P and 1,000,000 is a reasonable test set sample size.

B. EXPERIMENT FOR PRNG AND QRNG

After that, we perform the similar test on the random numbers generated by LCG, MT and QRNG. The first 1,000,000 instances are used as the training set and after 50,000,000 bits, every 1,000,000 instances are used as one test set. Ten test sets are selected consecutively. The experiment is repeated ten times. The results are shown in Fig. 4.

The prediction success rate of LCG test sets greatly exceeds 3σ and 5σ bounds. LCG is a well-known and widely used pseudo-random number generator and mathematical problems involved in its inverse solution is difficult to resolve. However, its generation algorithm is not complex. Thus, we use the ANN to approximate its algorithm and finally get a fairly high prediction success rate, which is much greater than the ideal random sequence.

Only one MT test data set exceeds the 3σ bound indicating that it contains enough randomness to satisfy the test. The current ANN can not effectively "learn" something from the data generated by MT and simulate the MT pseudo-random algorithm. This may result from the ANN we employed is not complicated enough and the network transfer function is not efficient enough and the amount of computation power involved is also not enough to cover the complexity of the algorithm which can break the MT pseudo-random number generator, so there are no major fluctuations about the prediction success rate.

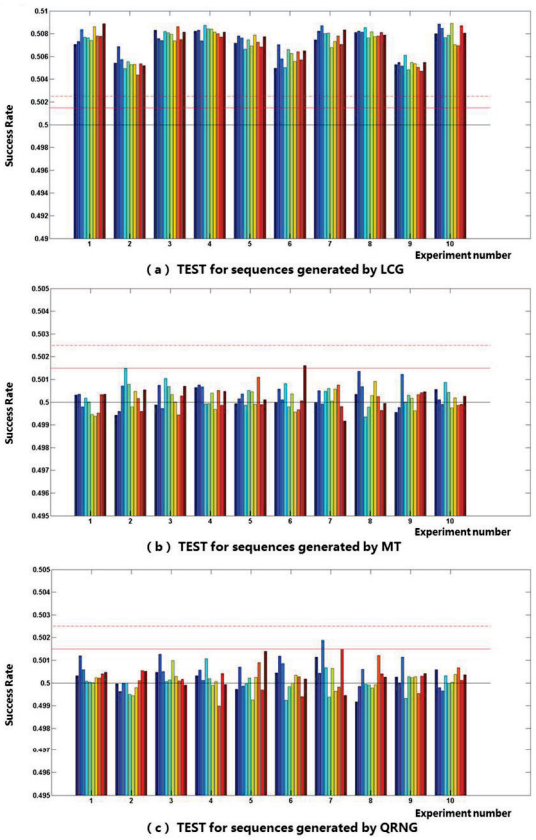


FIGURE 4. Test results for LCG, MT, and QRNG. (a) Test results for LCG. The abscissa is the repeated number of experiments and the ordinate is the success rate. Each experiment contains nine test subsets, which is comprised of 1,000,000 cases and each case includes 7 digits. The experiment is repeated 10 times. The red lines are 3σ and 5σ bound respectively. It can be the prediction success rate for the LCG is all above the 5σ bound, which means that the randomness generated by the algorithm is the worst among the four kinds of random sequence. (b) Test results for MT. One set of data has a prediction success rate of more than 5σ bound, and another set of data have a prediction success rate of more than 3σ bound. (c) Test results for QRNG. Only one set of data had a prediction success rate greater than 5σ bound.

Similarly, for the quantum random numbers, only one group of testing data exceeds 3σ bound, which also indicates that it contains randomness of high quantity. There are no significant common features and modes between the training data sets and the testing data sets in the random sequence. This may be due to the fact that there is no significant correlation between the sequences of quantum random numbers and no algorithm can simulate the generation process of its random sequences. Even with more complex networks and stronger computations, it may be still impossible to find the connection between random numbers and effectively predict it. We may figure out a valid way to verify the random sequences generated by quantum physics with ANN. Furthermore, we may also take a deeper insight about the uncertainties in quantum mechanics using machine learning methods.

V. CONCLUSION

In this paper, we propose a new randomness test method based on ANN and utilize it to assess randomness of four different kinds of random numbers. In the experiment, the test can discern the pseudo-random algorithm to some extent and the ANN we employed in this paper can single out the random sequences generated by the natural number π and LCG, while it can not distinguish the random numbers generated by MT and QRNG because of the simple construction of the ANN. With the increase of the complexity of the pseudo-random algorithm, for example, for the random sequence generated by pseudo random generator MT, it may also require a large amount of computing resources to design the ANN to distinguish the randomness of its output and approach its generation algorithm. Furthermore, the test proves that the random sequences generated by vacuum fluctuation noise contain randomness of high quantity. We expect to utilize more complex ANN which could involve more computing resources to effectively approximate more complex pseudo-random algorithms and single out the random sequences generated by quantum physics processes. We believe the test based on the ANN has the potential to detect random numbers generated by chaotic processes.

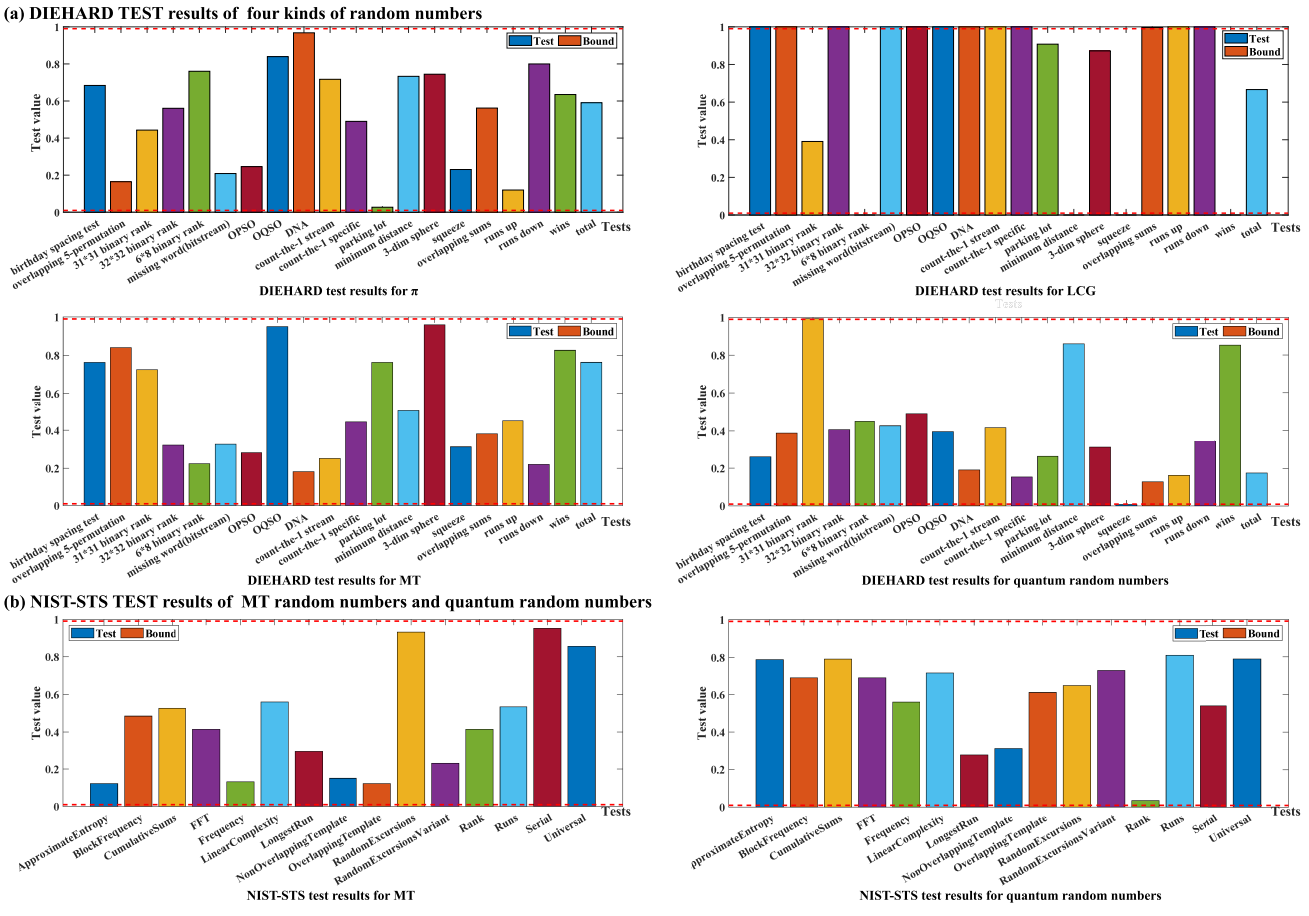


FIGURE 5. DIEHARD and NIST-STS test results. (a) DIEHARD test results of four kinds of random numbers. The x-coordinate is the tests contained in the DIEHARD test package and the y-coordinate is the test value for the random numbers. The red dotted line is the bound. When the test value exceeds the bound, it means that the set of random numbers does not pass the test. After passing all 20 tests, the set of random numbers is considered to pass the DIEHARD test package. (b) NIST-STS test results of MT random numbers and quantum random numbers. The x-coordinate is the tests contained in the NIST-STS test package and the y-coordinate is the test value for the random numbers. The red dotted line is the bound. NIST-STS test package includes 15 tests. If the random numbers pass all 15 tests, it can be considered to pass the NIST-STS test package.

In the development process of human history, people's recognition of randomness is also constantly improved. However, there has been no clear definition of randomness so far, which also makes testing randomness difficult. Although this paper only proposes a new method of randomness testing, what it really wants to illustrate is that, compared with the pseudo-random number generated by the algorithm, the quantum random number generated by the quantum physical process cannot be described by the fixed process, which means that it should not be calculated. So, the question is that can the sequence be considered truly random, if it cannot be computed using infinite computational forces. As the most powerful computing tool available to human beings at present, ANN can adapt to different problems by adjusting its own parameters, which is guaranteed by the universal approximation theorem. It may be an appropriate tool for us to calculate randomness. Combining randomness with ANN is a very interesting study. On the one hand, it may be possible to make a new definition of randomness by calculating, which needs further research. On the other hand, different degree of randomness may also become the standard to measure the effectiveness of ANN.

Based on ANN, which is one of the most powerful computational tools, and according to the universal approximation theorem, we intend to approximate the pseudo-random algorithms and pick out random sequences generated by quantum physical processes by calculating the statistical fluctuations of the prior probability and posterior probability of random sequences. In order to reach this goal, we test 4 kinds of random sequences generated by the natural number π, pseudo-random algorithm LCG, pseudo-random algorithm MT and vacuum noise, respectively. The test results show that the randomness from the original random source π and the pseudo-random algorithm LCG can be distinguished by the ANN. However, randomness from the widely used pseudo-random algorithm MT and quantum physics process vacuum fluctuation can not be effectively distinguished. We expect to complete the test method with more powerful computing resources and more sophisticated ANN networks and algorithms. On the one hand, if the MT algorithm can eventually be excluded from the true random sequence, most of the existed pseudo-random algorithms would lose their security and it would also help us further understand the randomness of quantum random sequences and uncertainty of quantum mechanics. On the other hand, if the true random sequence can be excluded, it indicates that the device we use to generate the true random number is not reliable, which helps to improve our research and understanding of the performance of the quantum random number generator, and also shows that it is better to verify randomness from a random source rather than its random sequence results.

APPENDIX
DIEHARD AND NIST-STS TEST RESULTS

In the Appendix, we add DIEHARD and NIST-STS test results of four kinds of random numbers, which are shown in the Fig. 5. By comparing the test results of different random numbers, we can roughly infer how much randomness is contained in the four kinds of random numbers. However, these tests are originally constructed for pseudo-random numbers produced by the algorithm. They are tested primarily on the basis of statistical distribution rather than the fundamental property of randomness, which is unpredictability.

Due to insufficient data, we test four kinds of random numbers with DIEHARD and two kinds of random numbers with NIST-STS. From the Fig. 5, we can see that MT random numbers and true random numbers have basically passed the DIEHARD and NIST-STS tests, which indicates that these two kinds of random numbers have good randomness. However, the random number generated by π and LCG algorithm contains less randomness.

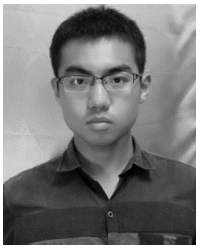
ACKNOWLEDGMENT

The authors are very grateful to their tutor for his advice and guidance during the experiment. They would also like to thank their colleagues for their help.

REFERENCES

[1] S. J. Phillips, R. P. Anderson, and R. E. Schapire, "Maximum entropy modeling of species geographic distributions," *Ecol. Model.*, vol. 190, nos. 3–4, pp. 231–259, Jan. 2006.
[2] S. J. Phillips and M. Dudík, "Modeling of species distributions with maxent: New extensions and a comprehensive evaluation," *Ecography*, vol. 31, no. 2, pp. 161–175, Apr. 2008.
[3] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, "A theory of power-law distributions in financial market fluctuations," *Nature*, vol. 423, no. 6937, pp. 267–270, May 2003.
[4] R. Tamblyn, R. Laprise, J. A. Hanley, M. Abrahamowicz, S. Scott, N. Mayo, J. Hurley, R. Grad, E. Latimer, R. Perreault, P. McLeod, A. Huang, P. Larochelle, and L. Mallet, "Adverse events associated with prescription drug cost-sharing among poor and elderly persons," *Jama-J. Amer. Med. Assoc.*, vol. 285, no. 4, pp. 421–429, JAN. 24. 2001.
[5] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Adv. Phys.*, vol. 51, no. 4, pp. 1179–1187, JUN. 2002.
[6] C. H. Bennett and G. Brassard, "Quantum cryptography and its application to provably secure key expansion, public-key distribution, and coin-tossing," in *Proc. IEEE Int. Symp. Inf. Theory*, Dec. 1983, p. 91.
[7] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Phys. Rev. Lett.*, vol. 67, no. 6, pp. 661–663, Aug. 1991.
[8] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Modern Phys.*, vol. 74, no. 1, pp. 145–195, Mar. 2002.
[9] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Rev. Mod. Phys.*, vol. 81, no. 3, pp. 1301–1350, Sep. 2009.
[10] I. V. Volovich, "Randomness in classical mechanics and quantum mechanics," *Found. Phys.*, vol. 41, no. 3, pp. 516–528, Mar. 2011.
[11] P. Xu, Y. L. Wong, T. K. Horiuchi, and P. A. Abshire, "Compact floating-gate true random number generator," *Electron. Lett.*, vol. 42, no. 23, pp. 1346–1347, Nov. 2006.
[12] C. S. Petrie and J. A. Connelly, "A noise-based IC random number generator for applications in cryptography," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 5, pp. 615–621, May 2000.
[13] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, "A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC," *IEEE Trans. Comput.*, vol. 52, no. 4, pp. 403–409, Apr. 2003.
[14] R. Hamza, "A novel pseudo random sequence generator for image-cryptographic applications," *J. Inf. Secur. Appl.*, vol. 35, pp. 119–127, Aug. 2017.
[15] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators," *Rev. Modern Phys.*, vol. 89, no. 1, Feb. 2017, Art. no. 015004.

[16] X. Ma, X. Yuan, Z. Cao, B. Qi, and Z. Zhang, "Quantum random number generation," *NPJ Quantum Inf.*, vol. 2, no. 1, p. 16021, Nov. 2016.
[17] M. Stipc and B. M. Rogina, "Quantum random number generator based on photonic emission in semiconductors," *Rev. Sci. Instrum.*, vol. 78, no. 4, 2007, Art. no. 045104, doi: 10.1063/1.2720728.
[18] A. J. Martino and G. M. Morris, "Optical random number generator based on photoevent locations," *Appl. Opt.*, vol. 30, no. 8, pp. 981–989, MAR. 10. 1991, doi: 10.1364/AO.30.000981.
[19] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A fast and compact quantum random number generator," *Rev. Sci. Instrum.*, vol. 71, no. 4, pp. 1675–1680, Apr. 2000, doi: 10.1063/1.1150518.
[20] H. Guo, W. Tang, Y. Liu, and W. Wei, "Truly random number generation based on measurement of phase noise of a laser," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, no. 5, May 2010, Art. no. 051137, doi: 10.1103/PhysRevE.81.051137.
[21] X. Ma, F. Xu, H. Xu, X. Tan, B. Qi, and H.-K. Lo, "Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction," *Phys. Rev. A, Gen. Phys.*, vol. 87, no. 6, Jun. 2013, Art. no. 062327.
[22] H. Zhou, X. Yuan, and X. Ma, "Randomness generation based on spontaneous emissions of lasers," *Phys. Rev. A, Gen. Phys.*, vol. 91, no. 6, Jun. 2015, Art. no. 062316.
[23] J. Liu, J. Yang, Z. Li, Q. Su, W. Huang, B. Xu, and H. Guo, "117 Gbits/s quantum random number generation with simple structure," *IEEE Photon. Technol. Lett.*, vol. 29, no. 3, pp. 283–286, Feb. 1, 2017, doi: 10.1109/LPT.2016.2639562.
[24] C. Gabriel, C. Wittmann, D. Sych, R. Dong, W. Mauerer, U. L. Andersen, C. Marquardt, and G. Leuchs, "A generator for unique quantum random numbers based on vacuum states," *Nature Photon.*, vol. 4, no. 10, pp. 711–715, Oct. 2010.
[25] T. Symul, S. M. Assad, and P. K. Lam, "Real time demonstration of high bitrate quantum random number generation with coherent laser light," *Appl. Phys. Lett.*, vol. 98, no. 23, Jun. 2011, Art. no. 231103.
[26] B. Xu, Z. Chen, Z. Li, J. Yang, Q. Su, W. Huang, Y. Zhang, and H. Guo, "High speed continuous variable source-independent quantum random number generation," *Quantum Sci. Technol.*, vol. 4, no. 2, Apr. 2019, Art. no. 025013.
[27] Z. Zheng, Y. Zhang, W. Huang, S. Yu, and H. Guo, "6 GBPS real-time optical quantum random number generator based on vacuum fluctuation," *Rev. Sci. Instrum.*, vol. 90, no. 4, Apr. 2019, Art. no. 043105, doi: 10.1063/1.5078547.
[28] G. Marsaglia. (1996). *Diehard: A Battery of Tests of Randomness*. [Online]. Available: http://www.stat.fsu.edu/pub/diehard/
[29] NIST. (2010). *STS: A Statistical Test Suite for the Validation of Random Number Generators and Pseudo-Random Number Generators for Cryptographic Applications*. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/rng/
[30] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the kohn-sham equations with machine learning," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, Oct. 2017.
[31] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[34] H.-C. Shin, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.
[35] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1240–1251, May 2016.
[36] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.
[37] H. Chen, Y. Zhang, Y. Chen, J. Zhang, W. Zhang, H. Sun, Y. Lv, P. Liao, J. Zhou, and G. Wang, "LEARN: Learned experts' assessment-based reconstruction network for sparse-data CT," 2017, *arXiv:1707.09636*. [Online]. Available: http://arxiv.org/abs/1707.09636
[38] N. D. Truong, J. Y. Haw, S. M. Assad, P. K. Lam, and O. Kavehei, "Machine learning cryptanalysis of a quantum random number generator," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 403–414, Feb. 2019.
[39] A. D. Tranter, H. J. Slatyer, M. R. Hush, A. C. Leung, J. L. Everett, K. V. Paul, P. Vernaz-Gris, P. K. Lam, B. C. Buchler, and G. T. Campbell, "Multiparameter optimisation of a magneto-optical trap using deep learning," *Nature Commun.*, vol. 9, no. 1, Dec. 2018, Art. no. 4360.
[40] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, 1989, doi: 10.1007/BF02551274.
[41] J. Kelsey, K. A. McKay, and M. S. Turan, "Predictive models for min-entropy estimation," in *Cryptographic Hardware and Embedded Systems*, T. Gãneysu and H. Handschuh, Eds. Berlin, Germany: Springer, 2015.
[42] F. Fenglei and W. Ge, "Learning from pseudo-randomness with an artificial neural network-does god play pseudo-dice?" *IEEE Access*, vol. 6, pp. 22987–22992, 2018, doi: 10.1109/ACCESS.2018.2826448.



YULONG FENG received the B.S. degree in electronic science and technology engineering from Naikai University, Tianjing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronics and the Center for Quantum Information Technology, Peking University. His research interests include quantum information, quantum optics, applied mathematics, machine learning, and medical imaging.

LINGYI HAO received the B.S. degree from the Harbin Institute of Technology, Weihai, China, in 2018. He is currently pursuing the M.S. degree with the Beijing University of Posts and Telecommunications. His current research interests include computer vision, deep learning, and edge computing.