Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

Implementation of High-Speed Pseudo-Random-Number Generator with Chaotic and Random Neural Networks

Hitoaki Yoshida
Iwate University
hitoaki@iwate-u.ac.jp

Haruka Fukuchi
Iwate University
e0216029@iwate-u.ac.jp

Takeshi Murakami
Iwate University
mtakeshi@iwate-u.ac.jp

Abstract

Chaotic and random time series generated from improved chaotic and random neural network (CRNN) afford statistically appropriate pseudo-random number series for information security. Randomness of outputs of CRNN is empirically validated in detail, and control methods of an appropriate ratio of chaotic character and randomness in the time series for PRNG is reported. The rate of random number generation has reached 2.8530×10^{12} b/s. In future, the generator may play an important role on implementing applications for protecting personal information on the Internet.

1. Introduction

Chaotic time series from the artificial neural networks (for example CNN in Figure.1) are useful for a pseudo-random number generator (PRNG) for stream cipher. CNN consists of 4 conventional artificial neurons [1-6]. It is useful for protecting private data and keeping safety on the Internet.

Recently, we have reported pseudo-random number series from CNN with fix-point arithmetic for applying the cipher to embedded systems [5,6]. The fix-point arithmetic is a simple Q5.26 without a carry and it allows overflow and underflow of variables. APLF (Asymmetric Piecewise-Linear-Function) [3-6] has been used as an activation function for the networks (Figure 2). The preliminary study suggests that the time series has both chaotic and random property; therefore the network is called CRNN (Chaotic and Random Neural Network) hereinafter.

In this paper, we report experimental validation of randomness in CRNN outputs, and report control methods of an appropriate ratio of chaotic character and randomness in the time series for PRNG, and also report a novel fast and secure PRNG. It is expected that information security applications using CRNN can be applied to IoT device or embedded systems which do not support double-precision-floating-point arithmetic, and also smartphones with GPU.

2. Properties of CRNN

2.1. Iterations of CRNN and the extraction method of pseudo-random number

The network that composed of 4 neurons in the discrete-time system has been used for a chaos generator (Figure 1). I_j is an external input of jth neuron. A total value of inputs in jth neuron at time t (t = 0, 1, 2, ...) is defined as equation 1. w_{ij} is a synaptic weight and x_i is an input from ith neuron at time t. An output from jth neuron at time t+1 is defined as equation 2 with the asymmetric piecewise-linear-function (APLF) f (Figure 2). The time series generated from CRNN can separate into 2 independent subseries; α series and β series [5,6]. In other words, 2 subseries are simultaneously generated.

Computer generated chaotic time series is *eventually* periodic by the calculation with the finite precision within our knowledge. A period of chaotic time series from CRNN changes with a different external input value I. Therefore, a perturbation I_D is added to an external input I at odd discrete time t = 1, 3, 5, ... where perturbation I_D is a small additional value. 2 subseries don't always reach different periodic orbits, the perturbation I_D, however, leads the 2 subseries to different periodic orbits [5,6]. Generally, I_D can be decided randomly. Result on α series is shown in the following unless otherwise mentioned.

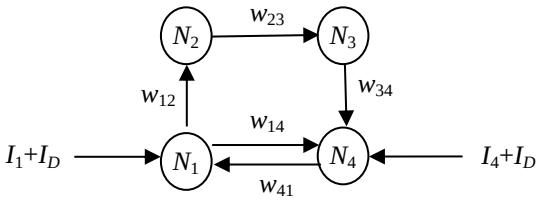


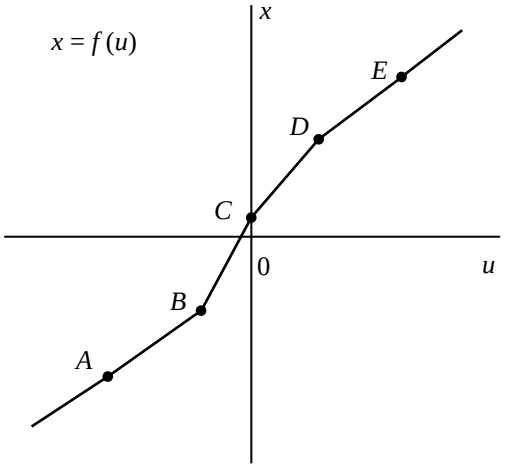
Figure 1. Neural networks consist of 4 neurons having cyclic structure (C4-nn).
CNN: C4-nn with double-precision-floating point arithmetic.
CRNN: C4-nn with fix-point arithmetic (Q5.26).

URI: https://hdl.handle.net/10125/64528
978-0-9981331-3-3

$$u_j(t) = \sum_{i=1}^{n} w_{ij} x_i(t) + I_j \qquad (1)$$

$$x_j(t+1) = f(u_j(t)) \qquad (2)$$

Pseudo-random number series were extracted from outputs of CRNN by the method shown in Figure 11(a) until now because the chaotic time series itself is not uniform as shown in Figure 3.



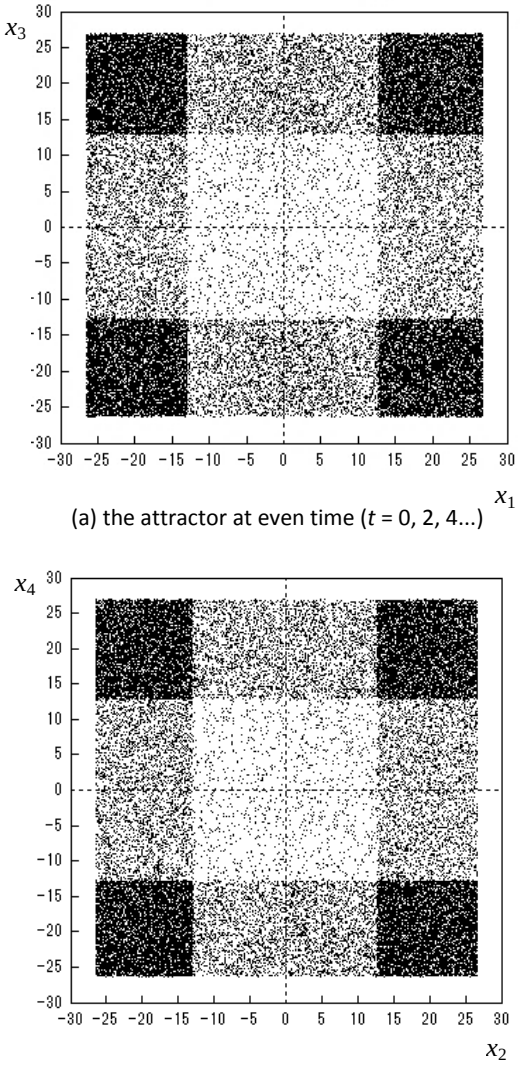
	APLF1(f_1)		APLF3(f_3)	
	u	*x*	*u*	*x*
A	-21.0001	-21.0	-31.0001	-31.001
B	-4.980101	-12.9899	-7.9811	-8.29999
C	0.0	0.499012	0.0001	0.500012
D	4.980101	12.6891	7.981101	8.6901
E	21.0002	21.0	31.0002	31.00999

Figure 2. An activation function *f* (APLF) for CRNN using fix-point arithmetic.

The value of the points, *A*-*E* can be decided randomly, APLF1 (f_1) and APLF3 (f_3) are used in this work.

2.2. Randomness in outputs of CRNN

Randomness in outputs of CRNN is observed on the attractor as shown in Figure 3. The distribution, however, is not uniform, because APLF1 has 3 fixed points; 2 attractive fixed points near (20.999784, 20.999784) and (-20.999900, -20.999900), and the repelling fixed point near (-0.292066, -0.292066). The interval of possible values on Q5.26 is $\boldsymbol{X}$ = [-32, 32-2^{-26}], so initial values of CRNN can be selected in the Cartesian product, $\boldsymbol{X} \times \boldsymbol{X}$. Points in $\boldsymbol{X} \times \boldsymbol{X}$ are attracted to the attractor shown in Figure 3; the interval of attractor is about $\boldsymbol{X}'$ = [-26.5, 26.7]. The volume of $\boldsymbol{X} \times \boldsymbol{X}$ is diminished to $\boldsymbol{X}' \times \boldsymbol{X}'$ in 2D phase space. The Lyapunov spectrum [7] of CRNN has positive value (1.664, 0.796) under the standard condition. Therefore it implies that the time series involves considerable randomness besides the chaotic properties. More evidence on randomness is mentioned in the following.



(a) the attractor at even time (t = 0, 2, 4...)

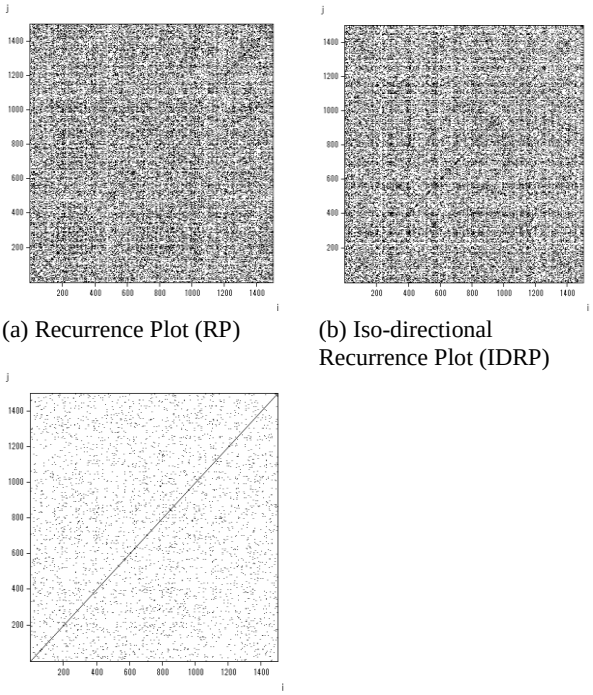
(b) the attractor at odd time (t = 1, 3, 5...)

Figure 3. The attractor of CRNN with APLF1 and synaptic-weight-set A.

The parameters are I_D = 0.119725, I = 0.000123, and synaptic-weight-set A (w_{12} = -12.60001, w_{23} = 5.951, w_{34} = -4.7004, w_{14} = 4.511, w_{41} = -7.345007).

The determinism of the time series (the outputs of CRNN) is analyzed by the iso-directional neighbors plot (IDNP) which is the product set of RP and IDRP. IDNP shows the set of points that keep neighbors and similar moving directions [8]. The result on the time

series from CRNN with APLF1 is shown in Figure 4. The determinism of the time series is low because the cardinality of the set IDNP is small. The result also supports high randomness of the outputs of CRNN with APLF1.



(a) Recurrence Plot (RP)

(b) Iso-directional Recurrence Plot (IDRP)

(c) IDNP (RP∩IDRP)

Figure 4. The determinism analysis of CRNN with APLF1.

Furthermore, the number-of-points dependency on the Lyapunov spectra was observed (Figure 5). Lyapunov exponents increase with the number of points. The dependency is particularly notable if time series involves larger dynamical noises [7]. In this case, a local versus global (LVG) plot of Lyapunov spectra is a possible method of analysis [7,8].

The result of LVG plot of CRNN with APLF1 is shown in Figure 6. The values of Lyapunov exponent change with the number of neighbors but they show no flat region. The precise value of Lyapunov exponent would not be determined by the method, too, the tendency, however, is characteristic of the chaotic time series with large dynamic noises.

Entropy of a dynamic system is known to distinguish chaotic, random, and regular motion (equation 3) [10,11]. In particular, entropy of chaotic system is discussed on the basis of the invariant measure of the strange attractor (chaotic attractor). That is, the invariant measure of the chaos attractor (μ_i) gives p_i (= μ_i), and the entropy S, which is considered as a scale of randomness.

$$S = -\sum_{i=1}^{np} p_i \log_2 p_i \qquad (3)$$

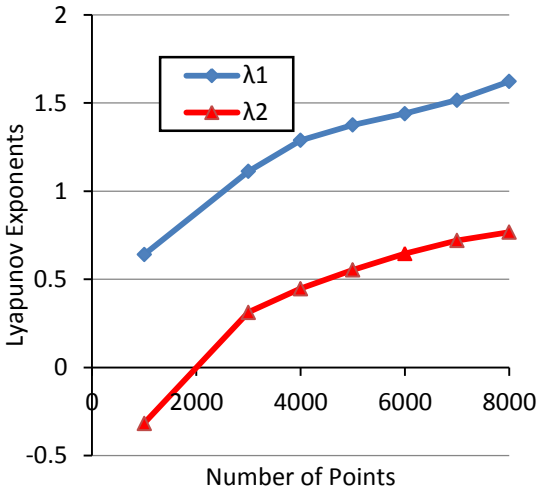


Figure 5. The number-of-points dependency on Lyapunov Spectra of CRNN outputs with APLF1.

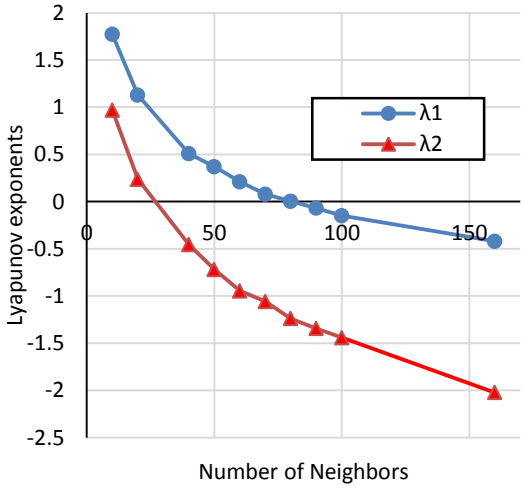


Figure 6. A local versus global plot of Lyapunov spectra of CRNN outputs with APLF1.

The entropy of CRNN outputs with APLF1 is determined by using the invariant measure shown in Figure 3. The number of partition (np) and the number of points (N) are determined as S approaches a maximum value in the system, that is, np = 4096, and $N = 10^5$ in this work. As for random time series (or uniform distribution), $p_i = 2^{-12}$ and then entropy in a bit unit should be S = 12 b, and 24 b for a whole network.

The entropy of the first line in Table 1 was determined by the invariant measure in Figure 3. The value, 21.60 b is larger than entropy of the chaotic time series from CNN which is only 4 b. It also suggests high randomness in the time series shown in Figure 3.

3. Pseudo-random numbers extracted from CRNN outputs

Pseudo-random numbers had been extracted by the method in Figure 11(a) until now. Generally upper bits of chaotic time series almost always include a fractal property; therefore the distribution is not uniform. And lowest 3 bits sometimes include calculation errors in double-precision-floating-point arithmetic.

The extracted pseudo-random number series are tested by NIST SP800-22 test suite [12,13]. Appropriate random number series for information security applications can be selected, because as for CNN the fail rates of the most NIST tests are below 1% [4,14]. The fail rate is the number of fails over 100-1000 times of the NIST tests.

As for CRNN with APLF1 and synaptic-weight-set A, results of the NIST tests are almost the same as results of CNN except the overlapping template matching test (OT test). The fail rate of OT test was 0.64% as an average of the test repeated 5000 times. The rate is higher than other tests which are normally about 0.1-0.3%. In order to investigate the cause of the results, a correlation between entropy and the fail rate is studied.

A new synaptic-weight-set has been designed to make various invariant sets corresponding to smaller entropy values, that is, synaptic-weight-set B: w_{12} = -1.60001, w_{23} = 5.0, w_{34} = -7.004, w_{14} = 4.511, w_{41} = -0.345007. The input-output characteristic of CRNN with synaptic-weight-set A is shown in Figure 7 and with synaptic-weight-set B is shown in Figure 8. The entropy under various conditions is shown in Table 1 and Table 2.

Table 1. Entropy of the time series.

Time Series			Entropy / b		
SWS [a]	APLF	I	x_1-x_3 [c]	x_2-x_4 [d]	total
A	APLF1	0.000123	10.81	10.79	21.60
B [b]	APLF1	-18.22696599	8.41	10.78	19.20
B [b]	APLF1	-9.00651951	9.34	10.58	20.15
A	APLF3	0.000123	11.88	11.88	23.76
Random Series [e]			12.00	12.00	24.00

a) Synaptic-weight-set A or B.
b) Synaptic-weight-set B corresponding to the time series shown in Figure 8.
c) Entropy at even time. d) Entropy at odd time.
e) Theoretical values on uniformly distributed series.

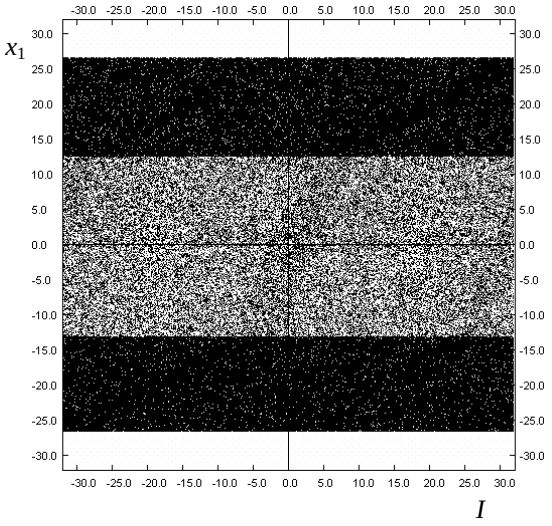


Figure 7. The Input-output characteristic of CRNN with APLF1 and synaptic-weight-set A.
The abscissa is I and the ordinate is x_1. The parameters are I_D = 0.119725, and synaptic-weight-set A.

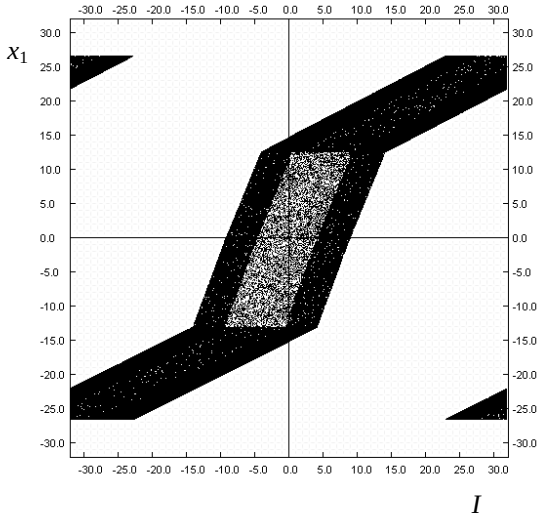


Figure 8. The input-output characteristic of CRNN with APLF1 and synaptic-weight-set B.
The abscissa is I and the ordinate is x_1. The parameters are I_D = 0.119725, and synaptic-weight-set B which is a special and rare set for the comparable study.

The selected results of the NIST tests on CRNN with synaptic-weight-set B are shown in Table 2; test fail rates corresponding external inputs Is, periods of time series, and entropy values of the invariant measure in a bit unit.

Table 2. Selected results of NIST SP800-22 test. [a)]

	Test Fail Rate (%) [b)]							
I	**FR**	**RU**	**RK**	**OT**	**AE**	**LC**	**Period**	***S*** [c)]
-19.30944918	1	0	0	0	0	0	2.0×10^9	19.23
-19.19931289	0	0	0	0	0	1	1.2×10^9	19.23
-18.22696599	0	0	1	3	0	0	4.6×10^9	19.20
-17.41411053	0	0	0	1	0	0	2.6×10^9	19.22
-17.38534870	0	0	0	0	0	0	2.6×10^9	19.23
-15.67024619	0	0	0	0	0	0	2.3×10^9	19.19
-14.64446048	0	0	1	0	0	0	3.3×10^9	19.23
-12.88730720	0	0	0	2	0	0	1.6×10^9	19.47
-11.78719633	0	0	0	1	1	1	3.7×10^9	19.65
-11.07755968	0	0	0	0	0	0	1.3×10^9	19.85
-9.95751086	0	0	2	1	0	1	6.3×10^9	20.09
-9.92205388	0	0	0	0	0	0	4.4×10^9	20.09
-9.00651951	0	0	0	1	0	1	1.3×10^9	20.15
11.69055082	0	0	1	1	1	0	3.8×10^9	19.72
12.77303401	0	0	0	1	0	0	1.7×10^9	19.54
13.58588947	1	0	0	2	0	0	2.5×10^9	19.32
16.29794228	0	0	0	2	0	0	3.5×10^9	19.33
16.35553952	0	0	0	1	0	0	3.8×10^9	19.32
17.22436138	0	0	0	1	0	0	5.4×10^9	19.27
18.11269280	0	0	0	1	0	0	2.0×10^9	19.34
19.21280367	0	2	0	2	0	0	3.5×10^9	19.27

a) Representative results on the examination of the proportion of sequences passing a test in the NIST tests. Abbreviations of test names: FR: Frequency test, RU: Run test, RK: Binary Matrix Rank Test, OT: Overlapping Template Matching Test, AE: Approximate Entropy Test, LC: Linear Complexity Test.

b) The test fail rate is an average of repeated 100 times of the NIST tests.

c) Total entropy of the invariant measure in a bit unit.

3. New activation-function APLF3 for improved PRNG

Next, the result in Table 2 is analyzed in order to determine whether the additional randomness should be increased or decreased.

The results of the NIST tests corresponding to various entropy values showed nearly the same tendency (Table 2). The fail rate of OT test corresponding to an external input was higher value 0-3%.

The fail rates of the most NIST tests as an average of the test repeated 2600 times over 26 external inputs were about 0.1-0.3%, and that of OT test was higher value 0.96%, too. The statistical correlation analysis between the test fail rates of OT test and the entropy values affords no correlation against expectations; a correlation coefficient, R = -0.140. It suggests that additional randomness has no effect on the statistical properties of the lower bits which extracted by the method in Figure 11(a) in this work. The slight increase of the test fail rate of OT may be caused by fix-point arithmetic, but a detailed mechanism is still unclear.

According to the result, a new activation-function APLF3 is designed to increase randomness as much as possible, because it is expected to hide the information of an activation function; for example existence of fixed-points, and a range of an attractor.

The attractor of the time series which generated from the CRNN with APLF3 is shown in Figure 9. The points of the attractor extend to the dynamic range, the distribution, however, isn't uniform. The Lyapunov spectrum of the time series has positive value (1.153, 0.338) under the standard condition (the number of neighbors is 20). Although a precise value of the Lyapunov spectrum is hard to decide, sensitive dependence on initial conditions is clearly observed. That is, 2 initial points of CRNN at shortest distance ($=2^{-26}$) exponentially apart to the attractor size within several iterations (*vide infra*).

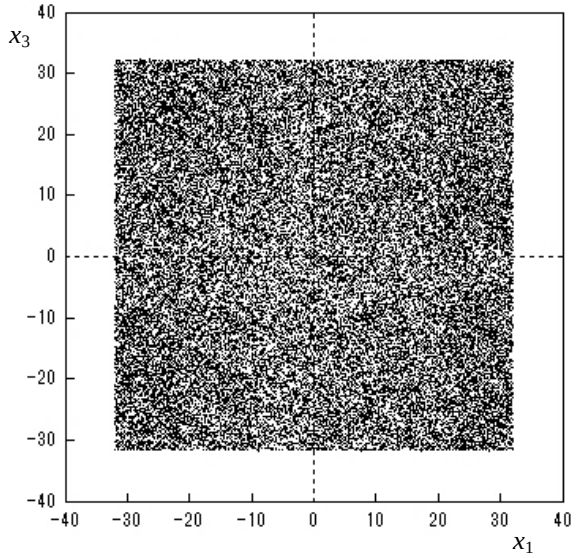
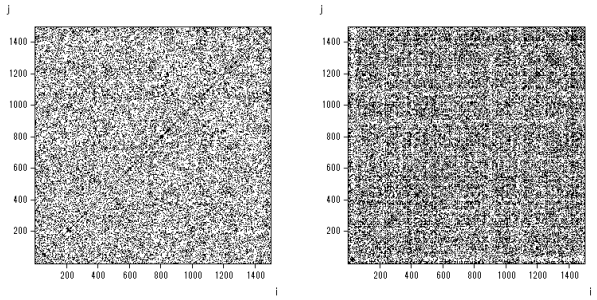


Figure 9. The attractor of the time series generated from CRNN with APLF3.

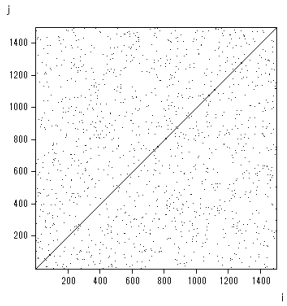
Therefore, approximation of the initial points (or outputs of CRNN) is no meanings on the time series from CRNN, because points in the neighborhood extend to different orbits.

The result of determinism analysis on the time series from CRNN with APLF3 is shown in Figure 10. The determinism of the time series with APLF3 is lower than that with APLF1 because the cardinality of the set IDNP is smaller.



(a) Recurrence Plot (RP)

(b) Iso-directional Recurrence Plot (IDRP)



(c) IDNP (RP∩IDRP)

Figure 10. The determinism analysis of CRNN with APLF3 (α1 series).

4. The new extraction method for CRNN with APLF3

A fail rate of each NIST test is valuable for evaluating result, and useful to determine an extraction method of random numbers from time series. In our work a test fail rate is estimated by repeated NIST tests and an overall result is judged by the test fail rate: less than 1.0% for normal test [14].

The fail rate of NIST tests on CRNN outputs with APLF3 is shown in Table 3 as an average of repeated 5000 tests over 50 external inputs. Only selected results are shown in Table 3 for simplicity.

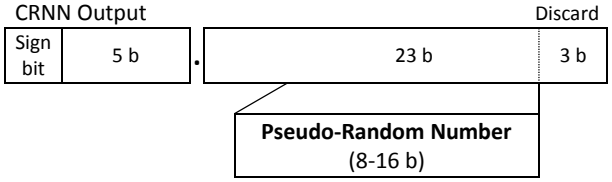
It suggests that the lower bits do not need discarding, and the border of higher bits which needs to discard is about 4-6 bits including a sign bit. The result is very interesting because APLF3 extends the extractable PRN from 8-16 b to 26-28 b (Method B in Figure 11(b)). The rate of PRN generation is expected to be faster.

The statistical property of the time series extracted by the new method (Method B) was confirmed as appropriate for information security applications with the NIST test.

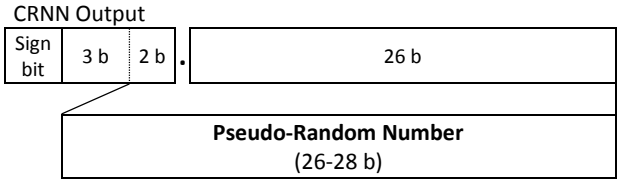
Table 3. Averaged fail rate (%) of 5000 NIST tests for the discarded number of bits. [c]

High [a]	0	1	2	4	6	8	8	8	8
Low [b]	0	0	0	0	0	0	1	2	3
FR	92	35	0.44	0.16	0.24	0.12	0.18	0.18	0.06
BF	5.4	10	11	0.06	0.08	0.06	0.04	0.10	0.06
CS	88	31	0.36	0.17	0.17	0.12	0.17	0.11	0.06
RU	100	100	0.76	1.9	0.16	0.14	0.28	0.12	0.10
LR	0.56	0.18	0.08	0.16	0.32	0.40	0.32	0.38	0.22
RK	0.10	0.10	0.20	0.16	0.20	0.16	0.10	0.12	0.16
OT	24	1.3	0.70	0.48	0.40	0.64	0.68	0.72	0.58

a) The discarded number of higher bits.
b) The discarded number of lower bits.
c) Abbreviations **of** test names: FR: Frequency test, BF: Frequency Test within a Block, CS: Cumulative Sums Test, RU: Run test, LR: Linear Complexity Test, RK: Binary Matrix Rank Test, OT: Overlapping Template Matching Test.



(a) Method A: an extraction method for APLF1 [5,6].



(b) Method B: a new extraction method for APLF3.

Figure 11. The extraction method of a pseudo-random number block from a CRNN output.

5. Implementation of an ultra-high-speed pseudo-random-number generator by GPU

Before implementation of a high-speed PRNG by GPU, the bit operation should be installed to prevent side-channel attacks. The outputs of CRNN are kept secret as internal states, the possibility of side-channel attacks would not be denied. Therefore, the 7-bit-

rotate-left instruction has been introduced before APLF mapping [5-6].

In the same way, the 7-bit-rotate-left instruction is introduced before APLF3 mapping at N_1 and N_2 also in this work. The statistical properties were confirmed by the NIST test.

Next, CRNNs have been implemented with CUDA 8.0 on PC mounted with a GPU [15] (NVIDIA Tesla P100, 3584 CUDA cores). The rate of pseudo-random number (PRN) generation has been extremely accelerated by the GPU (Figure 12). The rate of PRNG on CRNN with APLF3 and Method B has been superior to the previous rate (Method A) [6] and reached 2.85 Tb/s (10^{12} b/s) as shown in Figure 12 and Table 4.

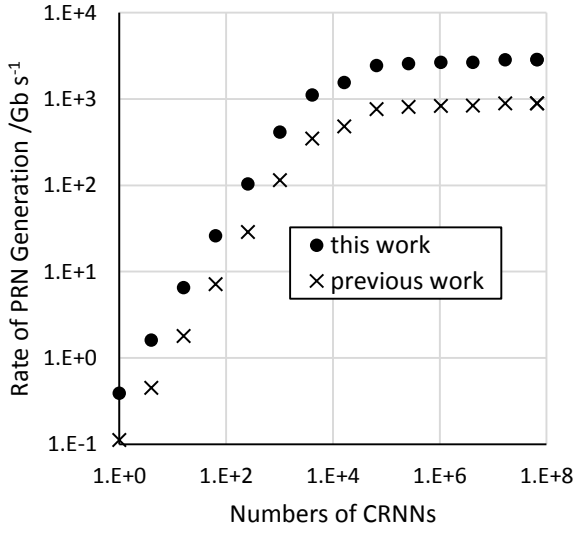


Figure 12. Rate of PRN generation from CRNNs.

Table 4. Rate of PRN generation by GPU.

Extraction	Number of Threads	Max Rate of PRN Generation /Tb s^{-1}
Method A [a)]	6.7108864×10^7	1.8570 [6]
Method A [b)]	6.6060288×10^7	1.7785 [6]
Method B [c)]	6.6060288×10^7	2.8530 (this work)

a) CRNN with APLF1.
b) CRNN with APLF1 and with 7-bit-rotate-left instruction.
c) CRNN with APLF3 and with 7-bit-rotate-left instruction.

The number of threads (the number of CRNNs) was optimized to realize the maximum rate. As for CRNN the number of blocks is 64512, the number of threads/blocks is 1024, that is, 6.6060288×10^7 CRNNs with different external inputs work in parallel on P100.

The simple structure of CRNN probably makes the huge number of threads possible.

The rate of PRN generation by GPU with the number of discarded upper bits is shown in Table 5. The rate corresponding to discarding 6 b is slower than others due to efficiency at coding. The number of discarded upper bits can be selected 4 or 8 b depending on the situation.

Table 5. Rate of PRN generation by GPU with the number of discarded upper bits.

Number of Discarded Upper Bits /b	Maximum Rate of PRN Generation /Tb s^{-1}
4	2.853
6	2.056
8	2.466

6. Predictable terms of chaotic time series.

Chaotic time series are characterized as long term unpredictability. A limit of a predictable term (T_c) is estimated by the following equation 4 [7,16]. ε is a difference of initial values, L is an attractor size, and K is a constant. In other words, the time T_c is the critical time between a predictable term and an unpredictable term. If $t > T_c$, the discrepancy grows to the order of the magnitude of the attractor size over the limits of the predictability of the system.

$$T_c = \frac{1}{K}\log_2\frac{L}{\varepsilon} \qquad (4)$$

T_c was estimated as an average of repeated 10^5 times experiments in this work (as for outputs of neuron 1 under randomly selected 10^5 initial values) (Table 6). The experimental value on the Logistic map is also shown as an example of a chaotic time series.

Table 6. Experimental result on predictable terms of time series.

Generator	ε	T_c [a)]	K [b)]
CRNN with APLF1	1.49×10^{-8}	6.75	9.57
CRNN with APLF3	1.49×10^{-8}	6.08	10.1
Logistic Map [c)]	1×10^{-15}	46.3	1.00
CNN	1×10^{-15}	43.9	1.08

a) T_c is an average of repeated 10^5 times experiments.
b) The unit is bits per cycle.
c) $y_{n+1} = 4y_n(1-y_n)$, where $n = 0,1,2...$

The result suggests that the predictable term is considerably reduced for CRNN, and especially CRNN

with APLF3. The detailed result on predictable terms will be discussed in further publication.

7. Conclusion

The randomness of outputs of CRNN has been validated by several methods. The result suggests that it is difficult to predict sequences in the outputs of CRNN because the outputs of CRNN have large sensitive dependence on initial conditions and lower determinism. And it also suggests that the predictable term is considerably reduced for CRNN, and especially CRNN with APLF3. The detailed result on the predictable term will be discussed in further publication.

The new activation-function APLF3 has been designed as extending randomness, and resulted in producing the better extraction method B (Figure 11(b)).

The secure and ultra-fast PRNG has been implemented on PC mounted with the GPU (NVIDIA Tesla P100), the speed has reached 2.85 Tb/s. It is 1.6 times faster than the previous PRNG.

In the previous study, the period of pseudo-random number generator consists of 4200 chaotic time series has estimated to be 10^{22432} [6]. It may be also possible for the new PRNG with APLF3. The period of the huge number of time series from the new PRNG will be confirmed in the next study.

As future work, we will apply the new PRNG and the new method to information security applications for IoT devices which don't support floating-point arithmetic and for also smartphones equipped with GPU.

8. Acknowledgment

Part of the experimental results in this research was obtained using supercomputing resources at Cyberscience Center, Tohoku University. This work was supported by KAKENHI Grant Numbers JP16K00180.

9. References

[1] H. Yoshida, K. Yoneki, Y. Tsunekawa, M. Miura, "Chaos Neural Network", *Proceedings of Papers, ISPACS'96*, vol. l of 3, pp.16.1.1-5, 1996.

[2] S. Kawamura, H. Yoshida, M. Miura, M. Abe, "Implementation of Uniform Pseudo Random Number Generator and Application to Stream Cipher based on Chaos Neural Network", *Proceedings of Papers, ICFS2002*, R-18, pp.4-9, 2002.

[3] H. Yoshida, T. Murakami, Z. Liu, "High-Speed and Highly Secure Pseudo-Random Number Generator based on Chaos Neural Network", *Proceedings of Papers*, *ICSSE 2015*, pp.224-237, 2015.

[4] H. Yoshida, T. Murakami, T. Inao, S. Kawamura, "Origin of Randomness on Chaos Neural Network", *Trends in Applied Knowledge-Based Systems and Data Science*, vol.9799, pp.587-598, 2016.

[5] H. Yoshida, Y. Kon and T. Murakami, "Chaos Neural Network for Ultra-Long Period Pseudo-Random Number Generator", *Proceedings of Papers, ITISE 2017*, vol.1, pp.102-113, 2017.

[6] H. Yoshida, Y. Akatsuka and T. Murakami, "Implementation of High-Performance Pseudo-Random Number Generator by Chaos Neural Networks using Fix-Point Arithmetic with Perturbation", *Proceedings of Papers, NOLTA* 2018, pp.46-49, 2018.

[7] K. Aihara, ed., "Basics and Application of Chaos Time Series Analysis", *Sangyo Tosho, Tokyo*, 2000.

[8] S. Horai, T. Yamada, K. Aihara, "Determinism Analysis with Iso-Directional Recurrence Plots", *IEEJ Transactions on Electronics, Information and Systems*, vol.122, pp.141-147, 2002.

[9] T. Ikeguchi and K. Aihara, "Lyapunov Spectral Analysis on Random Data", *International Journal of Bifurcation and Chaos*, Vol. 07, pp.1267-1282, 1997.

[10] E. Ott, "Chaos in Dynamical Systems", *Cambridge University Press, New York*, 1993.

[11] H. G. Schuster, "Deterministic Chaos: An Introduction", *Wiley, New York*, 1995.

[12] J. Soto, L. Bassham, "Randomness Testing of the Advanced Encryption Standard Finalist Candidates", *National Institute of Standards and Technology (NIST)*, 2000.

[13] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Vo. S. Dray, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST SP800-22 rev.1a, Revised: July 2015 (sts-2.1.1)", Lawrence E. Bassham III, 2015.

[14] H. Yoshida, T. Murakami, S. Kawamura, "Study on Testing for Randomness of Pseudo-Random Number Sequence with NIST SP800-22 rev. la", *Technical Reports of IEICE*, vol.110, pp.13-18, 2012.

[15] NVIDIA CUDA ZONE.
https://developer.nvidia.com/cuda-zone

[16] P. Gaspard, "Chaos, Scattering and Statistical Mechanics", Cambridge University Press, 2005.