TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 65, Issue III, September, 2022

IMPROVING NIST TEST SUITE 800-22 REV.1A BY ADDING VARIOUS CORRECTIONS ON THE TESTS AND OTHERS GOODNESS-OF-FIT TESTS TO CHECK THE UNIFORMITY IN SECOND-LEVEL TESTS

Elena-Iuliana GINGU (BOTEANU)

Abstract: *This paper goes over some of the most significant considerations for choosing and testing random number generators. The testing process is applied to random number generator output sequences, with the purpose of determining if the random sequences behave statistically inconspicuously. The bitstream's randomness is tested using the evaluation report suggested by NIST Test Suite 800-22 Rev.1a. Several investigations on the NIST randomness test suite's dependability have been published, with certain tests requiring corrections. By applying numerous adjustments to the tests, we review the NIST Statistical Test Suite in this study. Furthermore, a more precise interval of acceptable proportions was defined for the proportion of passing sequences. In the second level test, two more Goodness of Fit tests (Kolmogorov-Smirnov and Anderson-Darling) are implemented in order to improve the uniformity testing methodology. The results of the studies presented in this paper show that the new testing approach improves detectability and reliability under the same or different test conditions.*

Key words: *NIST tests, statistical analysis, corrections, Kolmogorov-Smirnov test, Anderson-Darling test.*

1. INTRODUCTION

Random Number Generators (RNGs), be it True Random Number Generators (TRNGs) or Pseudo-Random Generators (PRNGs), are critical components in a wide range of cryptographic applications. The statistical tests of random number generators are discussed in this study in order to discover deviations from randomness in a binary sequence. Randomness variations can be caused by a poorly constructed generator or anomalies in the binary sequence. It is required to conduct statistical tests in order to provide assumptions about the system's behavior, to build, confirm, and adjust, and finally to comprehend the source of randomness generated by the RNG.

The National Institute of Standard and Technology (NIST) offered a set of 15 statistical tests in [1,2] that were used to assess the randomness of bitstreams. The publications became known as SP 800-22. The latest version is revision 1a. Unfortunately, the publication SP 800-22 rev1a gives us only few directions on how to understand the NIST STS (Statistical Tests Suite) results; the explanations are either inadequate or offer only approximate values [3]. As a result, a number of corrections and improvements to statistical tests must be proposed in order to improve the interpretation of the results. Several papers have been already published in the literature.

Structure of the paper. In this article, we focus on the interpretation of the results provided by NIST STS. In order to evaluate the accuracy of the approximation for statistical test P-values, this study will investigate the NIST tests and make modifications and upgrades. The remainder of this study is structured as follows: Section 2 provides a brief overview of the NIST statistical test suite, testing methodology, and results interpretation. In Sections 3 and 4, we present a methodology for evaluating and interpreting NIST tests that we believe is necessary. We examine various corrections and improvements published in the literature and suggest a testing approach for improving the proportion of sequences passing a test and the

Received: 30.07.22; **Similarities**: 11.08.22: **Reviewed**: 23.08./30.08.22: **Accepted**:17.09.22.

uniformity in second-level randomness test. Then, in Section 5, we run some tests and compare the results obtained using various corrections from the literature with the findings obtained using the original NIST test suite. Finally, in Section 6, we wrap up and address future development possibilities.

2. TESTING METHODOLOGY PROPOSED BY NIST TEST SUITE 800-22

The technique for evaluating and interpreting NIST testing is provided below for the reader's convenience [2]:

1) *Choose a Generator and a set of sequence blocks with a length of N*

When the NIST statistical test suite is run, and the required bit stream length, N, is chosen, a list of generator options displays (underlined in Table 1). A binary sequence of 0 and 1 of length N should be generated by the generator or provided from outside.

Table 1

Test Code Generator Options for Running

The option's number	**Generator options**	**The option's number**	**Generator options**
[00]	Input File	[01]	Linear Congruential
[02]	Quadratic Congruential I	[03]	Quadratic Congruential II
[04]	Cubic Congruential	[05]	XOR
[06]	Modular Exponentiation	[07]	Blum-Blum-Shub
[08]	Micali-Schnorr	[09]	G Using SHA-1

2) *Binary sequence generation*

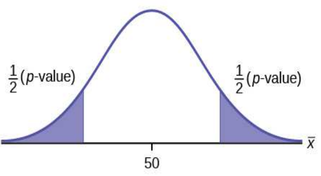
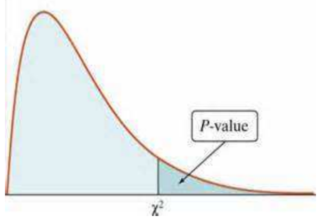
A collection of binary sequences m is produced for a fixed sequence of length N and a pre-selected generator or an input file, and the sequences are recorded in a file.

3) *The Statistical Test Suite is run*

The NIST statistical test suite consists of 15 empirical tests grouped into two categories: binomial and chi-square distribution-based tests [2]. The tests are used to analyze binary sequences, and they are briefly mentioned in Table 2.

Table 2

Distribution of the test statistic values

Name	**Distribution**
Frequency (Monobit)	Binomial 
Runs	
Discrete Fourier Transform (Spectral)	
Maurer's" Universal Statistical"	
Random Excursions Variant	
Frequency Test within a Block	Chi – Square 
Longest Run of Ones in a Block	
Binary Matrix Rank	
Non-overlapping Template Matching	
Overlapping Template Matching	
Linear Complexity	
Serial	
Approximate Entropy	
Cumulative Sums (Cusums)	
Random Excursions	

Two alternatives exist:

- To perform every test in a series, press key 1, and
- to run a single test, press key 0.

For instance, follow these steps to utilize the Overlapping Template Matchings test:

123456789111111
000000001000000

The appropriate sample size is then specified in a question: How many bit sequences ought to be generated?

4) *Examining p-value variables / Assessing the test: Passed / Failed*

Two techniques are recommended by NIST [2]:

The first-level test. Each statistical test will require intermediate data, such as test statistics and p-value variables, which the test suite will deliver in an output file. A conclusion about the sequence quality can be reached based on these p-value factors. Each statistical test in the NIST test suite examines the null hypothesis (H_0). According to the null hypothesis, the sequence is random. The alternative hypothesis (H_a), which is related to the null hypothesis, states that this sequence is not random. The decision to accept the null hypothesis - that is, whether or not the generator produces random values - is made next. For each test, a statistical test value is produced based on the data. The crucial value ($\alpha = 0.01$) is used to compare this statistical value. The null hypothesis is rejected if the statistical value is greater than the crucial value. If not, the null hypothesis is rejected and the alternative hypothesis is accepted.

To improve the reliability of statistical tests, the **second-level test** has been proposed. There are two methods for estimating the distribution of *N p-values*:

- Proportion of Sequences Passing a Test.
- Uniform Distribution of P-values.

Proportion of Sequences Passing a Test. NIST uses the normal distribution as an approximate representation of the binomial distribution. By counting the number of blocks with P-values equal to or higher than α, the passing ratio is determined. If the ratio falls within the following confidence interval:

$$\hat{p} \pm c \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{m}},$$

where, $m = simple\ size$, $\alpha = 0.01$, $\hat{p} = 1 - \alpha$, then, the first-level test is successfully passed. The formula is based on a binomial distribution estimate that is quite accurate for many tested sequences ($m \geq 1000$). According to NIST, $c = 3$ is the preferred value.

Uniform Distribution of P-values. The interval $[0, 1)$ is subdivided into k equal sub-intervals: $[0.0;\ 0.1), [0.1;\ 0.2), \dots, [0.9;\ 1.0)$. Then, a count of *P-values* inside each sub-interval is made. The NIST SP 800-22 test suite treats k as being equal to 10.

On these k integers, a Chi-Square test known as the Goodness-of-Fit Distributional Test is carried out using the presumptive uniform distribution. The following χ^2 value is computed as a result of this investigation:

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - \frac{s}{10})^2}{\frac{s}{10}}$$

where F_i is the quantity of *P-values* in sub-interval $[(i-1) \cdot 0.1, i \cdot 0.1]$, and s is the quantity of sequences (the sample size). After that it is generated a new *P-value* p_T:

$$p_{valueT} = igamc(\frac{9}{2}, \frac{\chi^2}{2})$$

If p_T is equal to or higher than another significance level $\alpha_T = 0.0001$ in the NIST SP 800-22 test suite, the second-level test is regarded as passing.

3. CORRECTIONS TO THE NIST SP 800-22 TEST BATTERY DESCRIBED IN THE LITERATURE

Discrete Fourier Transform (Spectral) Test. Assuming that $x_i = 2\varepsilon_i - 1$, for $1 \leq i \leq n$ and $X = x_1, x_2, \dots, x_n$ are true, we have x_i. Apply a Discrete Fourier Transform (DFT) to X to produce a sequence of n complex variables. $f = DFT(X)$ represents the periodic elements of a set of bits at various frequencies. Let mod_j be the complex modulus of f_j (let $f_1, f_2, \dots, f_{\frac{n}{2}}$ be the first $\frac{n}{2}$ components in *f*) for $1 \leq j \leq \frac{n}{2}$. A confidence interval can be calculated based on the values of mod_j under the assumption that X is random. More specifically, 95% of the values mod_j should be fewer than $T = \sqrt{nlog(\frac{1}{0.05})}$. Let $N_0 = \frac{0.95n}{2}$ be the expected theoretical number of peaks and let N_1 be the number of mod_j less than T. Following the conventional normal distribution, the statistic test is:

$$d = \frac{N_1 - N_0}{\sqrt{\frac{n \cdot p \cdot q}{c}}} = \frac{N_1 - 0.95\frac{n}{2}}{\sqrt{\frac{n \cdot 0.95 \cdot 0.05}{c}}} \sim N(0,1)$$

NIST advises using the value $c = 4$. (Proposed by Kim et al. in [5]).

Iwasaki [6] first established the theoretical value of $c = 3.7903$ using Parseval's theorem, followed by experimentation and discussion of the validity of the result. The equation is written as follows:

$$c = (2 \cdot 0.05 \cdot 0.95) / \left(\left(1 - \frac{T^2}{2m^2}\right)^{m-1} - \left(1 - \frac{T^2}{2m^2}\right)^{2m-2} + (m-1) \left\{ \left(1 - \frac{T^2}{2m^2}\right)^{m-1} - \left(1 - \frac{T^2}{2m^2}\right)^{2m-2} \right\} \right);$$

$$m = \frac{n}{2}.$$

Overlapping Test. Early versions of NIST used precise probabilities, but subsequently shifted to Polya-Aeppli probabilities. When they changed the numbers π_i in the source code, they neglected to comment or remove the instructions for calculating the real probabilities (lines 40–44) [7,8] (more precisely, in Overlapping Template Matching.c). As a result, precise probabilities were used in place of the Polya-Aeppli probabilities.

For greater accuracy, the probability π_i values were computed to 20 decimal places. These probabilities were calculated for the additional 9-bit and 10-bit patterns (by simulation).

pi[6] =

0.36409105321672786245
0.18565890010624038178
0.13938113045903269914
0.10057114399877811497
0.07043232634639844974
0.13986544587282249192

The formulas for calculating these probabilities are as follows: [5]:

$$\pi_i = \frac{T_i(n)}{2^n}$$

where $0 \leq i \leq 4$

$$T_0(n) = \begin{cases} 1, & n = -1 \\ 1, & n = 0 \\ 2T_0(n-1), & 1 \leq n \leq m-1 \\ 2T_0(n-1) - T_0(n-m-1), & n \geq m \end{cases}$$

$$T_1(n) = \begin{cases} 0, & n \leq m-1 \\ 1, & n = m \\ 2, & n = m+1 \\ \sum_{j=1}^{n-m-1} T_0(j) T_0(n-m-2-j), & n \geq m+2 \end{cases}$$

$$T_\alpha(n) = T_{\alpha-1}(n-1) + \sum_{j=1}^{n-2m-\alpha} T_0(j) T_{\alpha-1}(n-m-2-j)$$

$$\pi_5 = 1 - \sum_{i=0}^{4} \pi_i$$

Linear Complexity Test. The values of the probabilities π_i with a higher number of decimals were determined for the Linear Complexity test.

$\boldsymbol{pi}[7]$ =

$$\frac{1}{96}, \frac{1}{32}, \frac{1}{8}, \frac{1}{2}, \frac{1}{4}, \frac{1}{16}, \frac{1}{48}.$$

Random Excursions** and **Random Excursions Variant Test. Due to programming errors in calculating the uniformity of *P-values* in the STS software package, there are discrepancies in the computation of p_{valueT} from the third decimal point for Random Excursions and Random Excursions Variant tests:

- the intended value is of type integer. When multiplying two numbers (one integer and one double) by 10, the result loses the decimal part;
- when computing the expected value, the programming environment reacts differently depending on the order of the operands.

4. RESEARCH TESTING METHODOLOGY FOR IMPROVING THE PROPORTION OF SEQUENCES PASSING A TEST AND UNIFORMITY SECOND-LEVEL RANDOMNESS TEST

4.1 Proportion of sequences passing a test

The proportion test level correction entails revising the allowable proportion range using a more precise constant of $c = 2.6$. NIST's method is based on a binomial distribution approximation that is reasonably accurate for many of the sequences examined ($m = 1000$). The likelihood of the sequence of random sequences passing in the computed range is 99.73%, which equates to a type I error probability of 0.27%. As a result, if the range of the permissible fraction is computed using the following formula, the probability of a type I error will be closer to 1%:

$$1 - \alpha \pm 2.6 \cdot \sqrt{\frac{\alpha(1-\alpha)}{m}},\ \text{where}\ \alpha\ =\ 0.01$$

Thus, using the test parameters recommended by NIST, $n\ =\ 10^6$ or $n = 2^{20}$ and $N\ =\ 1000$, it will be obtained that if $p - value\ \in$ $[0.981819291;\ 0.998180709]$ then the test will be considered PASSED [9].

4.2 Uniformity of the distribution of the p-value variable

Two further GOF tests, the Kolmogorov-Smirnov and Anderson-Darling tests, will be employed to measure uniformity.

It is advised to do a GOF test under the uniform distribution hypothesis using a test like the Kolmogorov-Smirnov test (rather than the Chi-Square test) on the complete distribution of P-values [7].

The Kolmogorov-Smirnov test uses the CDF (cumulative distribution function) of U [0, 1] to examine the uniformity of the random integers.

Compared to the Chi-squared test, the Kolmogorov-Smirnov test has the following advantages:

- There are no intervals needed;
- It was developed for continuous data, like values sampled from a Uniform [0,1] random variable;
- The Kolmogorov-Smirnov test is a precise measurement.
- For the approximations to be valid, the Chi-square GOF test requires a sufficient sample size.

The Kolmogorov-Smirnov GOF uniformity test entails using the formula to get the Kolmogorov distribution [10,11]:

$$K(n, D_n)\ =\ Prob(D_n\ <\ x),$$

where

$$D_n\ =\ max\left(A_1 - \frac{0}{n}, A_2 - \frac{1}{n} \dots, A_n - \frac{n-1}{n}, \frac{1}{n} - A_1, \frac{2}{n} - A_2, \dots, \frac{n}{n} - A_n\right)$$

with $A_1 < A_2, \dots\ < A_n$ a set of n independent uniform random variables [0,1) sorted in ascending order and returns the variables $p - value = K(n, D_n)$.

The Kolmogorov-Smirnov Test for Uniformity is based on the following algorithm [11,12]:

Step 1. Evaluate $P\ [D_n <\ d]$;

Step 2. Write $d = \frac{k-h}{n}$, with k a positive integer and $0 \leq h < 1$;

Step 3. Apply the Durbin matrix formula:

$$K(n, d) = P\ [D_n \leq d] = \frac{n!}{n^n} t_{k,k}\ ,$$

where $t_{k,k}$ is k, k is the matrix element $T = H^n$, H is a matrix $m \times m$, $m = 2k - 1$;

Step 4. Rank the P-values in ascending order, $A_1, A_2, \dots, A_n$, where n is the size of the sample;

Step 5. Calculate

$$D^+\ =\ \max_{1<i<n} \left\{\frac{i}{n} - A_i\right\};$$

Step 6. Calculate

$$D^-\ =\ \max_{1<i<n} \left\{A_i - \frac{i-1}{n}\right\};$$

Step 7. Calculate

$$D\ =\ \max_{1<i<n} \{D^+, D^-\};$$

Step 8. The probability transformation $P\ =\ K(n, D_n)$ changes the randomly generated D_n to a uniform (0,1) variate (P-value);

Step 9. If $K(n, D) = p - value \geq 0.0001$, then the sequences can be considered to be uniformly distributed.

Moreover, many analysts consider Anderson-Darling GOF test to be more powerful than the original Kolmogorov-Smirnov test because it gives greater weight to the tails. The Anderson-Darling test is used to determine whether a set of data came from a population with a particular distribution.

The Anderson-Darling test for uniformity is putting a collection of ostensibly random data in ascending order and applying the method to calculate the statistic test [13]:

ADtest $=\ -N - \frac{1}{N} \cdot [ln(A_1 \cdot z_1) + 3 \cdot ln(A_2 \cdot z_2 + \dots + (2 \cdot N - 1) \cdot ln(A_N \cdot z_N)]$, where $z_1 = 1 - A_N,\ z_2 = 1 - A_{N-1}, \dots, z_N = 1 - A_1$

then find $v = adinf\ (a)$ and return the p-value variables associated with the observed variables $p = v + errfix\,(v)$, which should be uniform on [0,1).

The Anderson-Darling Test for Uniformity is based on the following procedure [13,14]:

Step 1. Arrange in ascending order $A_1, A_2, \dots, A_N$ the vector to be tested for uniformity and return the p-value variables associated with the Anderson-Darling test using $adinf$ () and $errfix$ ();

Step 2. Calculate $adinf\ (z)$

For $0 < z < 2$, max |error| $<$ 0.000002

$$adinf(z) = \exp(-\frac{1.2337141}{\mathrm{z}})/\sqrt{\mathrm{z}} \cdot (2.00012 + (0.247105 - (0.0649821 - (0.0347962 - (0.011672 - 0.00168691 \cdot \mathrm{z}) \cdot \mathrm{z}) \cdot \mathrm{z}) \cdot \mathrm{z}) \cdot \mathrm{z})$$

For z ≥ 2, max |error| < 0.0000008

$$adinf(z) = \exp\,(-\exp\,(1.0776 - (2.30695 - (0.43424 - (0.082433 - (0.008056\ - 0.0003146 \cdot \mathrm{z})\ \cdot \mathrm{z}) \cdot \mathrm{z}) \cdot \mathrm{z}) \cdot \mathrm{z}))$$

Step 3. Calculate $errfix(N, A)$

For $A > 0.8$

$$errfix(N, A) = (-130.2137 + (745.2337 - (1705.091 - (1950.646 - (1116.360 - 255.7844 \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}))/\mathrm{N}$$

$c = 0.01265 + \frac{0.1757}{N}$;

For $A < c$, $t = \frac{A}{c}$

$$t = \sqrt{t} \cdot (1 - t) \cdot (49 \cdot t - 102)$$

$$errfix(N, A) = \mathrm{t} \cdot \frac{\frac{0.0037}{\mathrm{N \cdot N}} + \frac{0.00078}{\mathrm{N}} + 0.00006}{\mathrm{N}}$$

For $c \leq A < 0.8$

$$t = \frac{A - c}{0.8 - c}$$

$t = -0.00022633 + (6.54034 - (14.6538 - (14.458 - (8.259 - 1.91864 \cdot t) \cdot t) \cdot t) \cdot t) \cdot t$;

$$errfix(N, A) = \mathrm{t} \cdot \frac{0.04213 + \frac{0.01365}{\mathrm{N}}}{\mathrm{N}}$$

Step 4. Calculate $AD(N, z)$

$$A = adinf(z)$$

For $A > 0.8$

$$v = errfix(N, A) = ((-130.2137 + (745.2337 - (1705.091 - (1950.646 - (1116.360 - 255.7844 \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}) \cdot \mathrm{A}))/N$$

$$AD(N, z) = A + v$$

$c = 0.01265 + \frac{0.1757}{N}$;

For $A < c$, $v = \frac{A}{c}$

$$v = \sqrt{v} \cdot (1 - v) \cdot (49 \cdot v - 102)$$

$$AD(N, z) = \mathrm{A} + \mathrm{v} \cdot \frac{\frac{0.0037}{\mathrm{N \cdot N}} + \frac{0.00078}{\mathrm{N}} + 0.00006}{\mathrm{N}}$$

For $c \leq A < 0.8$

$$v = \frac{A - c}{0.8 - c}$$

$v = -0.00022633 + (6.54034 - (14.6538 - (14.458 - (8.259 - 1.91864 \cdot v) \cdot v) \cdot v) \cdot v) \cdot v$;

$$AD(N, z) = \mathrm{A} + v \cdot \frac{0.04213 + \frac{0.01365}{\mathrm{N}}}{\mathrm{N}}$$

Step 5. Calculate $ADtest(N, A)$

$$t = A[i] \cdot (1 - A[n - 1 - i]))$$

$$z = z - (i + i + 1) \cdot log(t)$$

$$ADtest(N, A) = AD(N, -N + \frac{z}{N})$$

Step 6. If $ADtest(N, A) = p - value \geq 0.0001$ then the sequence can be considered to be evenly distributed.

5. EXPERIMENTAL RESULTS

In this paper, following [12,14,15], we evaluated NIST statistical tests in order to examine the accuracy of the *P-values* approximation.

The following parameters are used to test the battery:

- Sample file (Input data): random1e9.dat, a binary file [16] as an example
- File size: 125000000 bytes
- Significance Level $\alpha = 0.01$
- Proportion constant $c = 2.6$
- Sample size $N = 10^3$

- Frequency (Monobit): the bit stream length $n = 2^{20}$
- Runs: the bit stream length $n = 2^{20}$
- Discrete Fourier Transform: the bit stream length $n = 10^6$
- Maurer's" Universal Statistical": the bit stream length $n = 10^6$
- Random Excursions Variant: the bit stream length $n = 2^{20}$
- Frequency within a Block: the bit stream length $n = 2^{20}$; (Sub)block=128
- Longest Run of Ones in a Block: the bit: stream length $n = 2^{20}$; (Sub)block = 10000
- Binary Matrix Rank: the bit stream length $n = 2^{20}$
- Non-overlapping Template Matching: the bit stream length $n = 2^{20}$; (Sub)block = 8, Patterns size = 9, Patterns numbers=148
- Overlapping Template Matching: the bit stream length $n = 10^6$; Patterns size = 9, Patterns numbers=1
- Linear Complexity: the bit stream length $n = 2^{20}$; (Sub)block = 500
- Serial: the bit stream length $n = 2^{20}$; (Sub)block = 16
- Approximate Entropy: the bit stream length $n = 2^{20}$; (Sub)block = 10
- Cumulative Sums (Cusums): the bit stream length $n = 2^{20}$
- Random Excursions: the bit stream length $n = 2^{20}$.

The updated NIST test suite yielded the following results (Figs. 1,2,3):

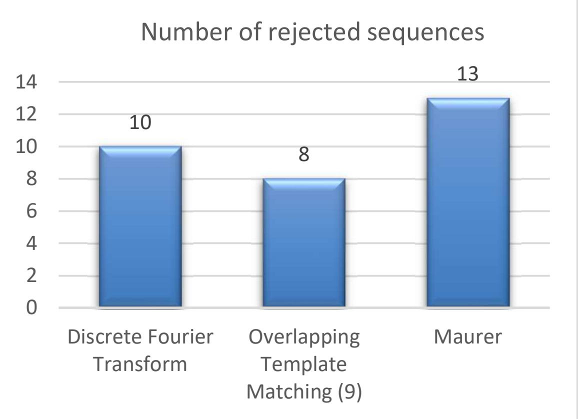


Fig. 1. The number of rejected sequence (from a total of 1000 sequences)

The maximum number of rejected sequences of these tests is in the range [3.71, 16.29].

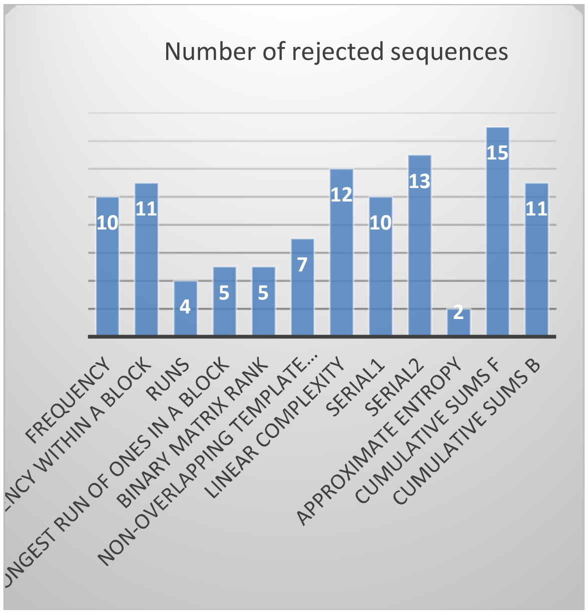


Fig. 2. The number of rejected sequence (from a total of 953 sequences)

The maximum number of rejected sequences of these tests is in the range [0.00, 15.67].

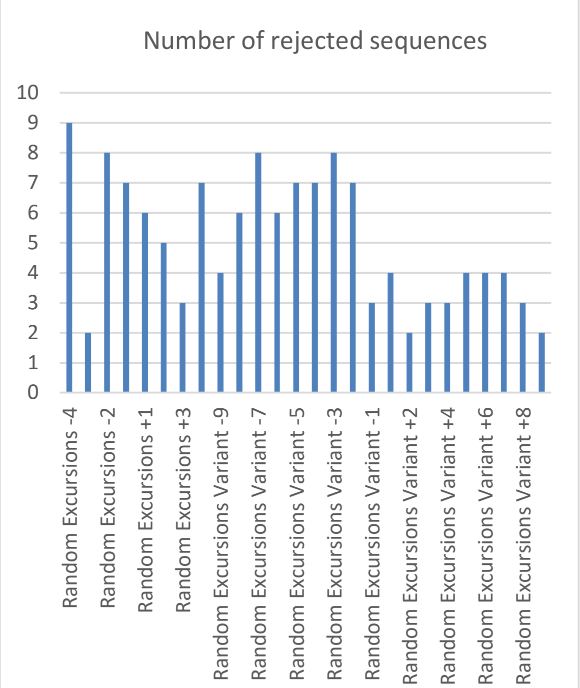


Fig. 3. The number of rejected sequence (from a total of 606 sequences)

The maximum number of rejected sequences of these tests is in the range [0.00, 10.96].

It can be observed (Fig. 4), that fewer sequences are rejected after applying modifications when comparing the improved NIST tests battery to the previous implementation of the NIST tests suite (sts-2.1.2).

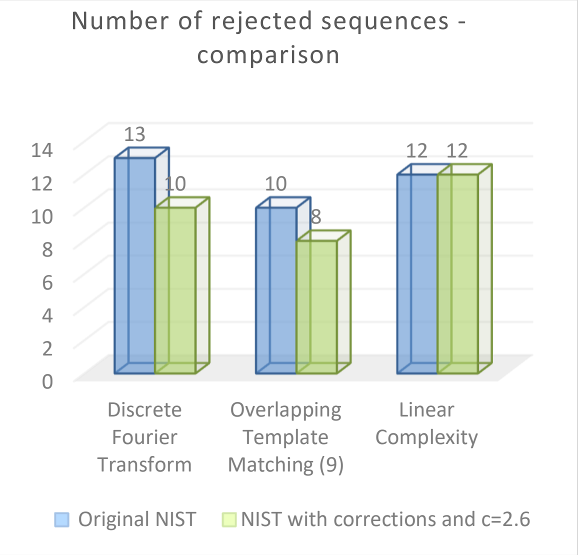


Fig. 4. Comparison between the original NIST and the new implementation

The Anderson-Darling test yielded superior findings than Kolmogorov-Smirnov or Chi-Square because the results are more concentrated in the center of the interval [0,1], which is indicative of the uniformity of the distribution of the p-values (Figs. 5, 6).

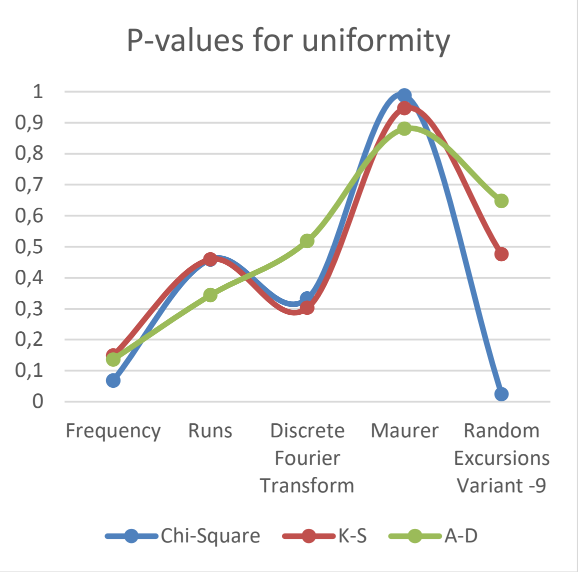


Fig. 5. The p-values for the tests that followed the binomial distribution

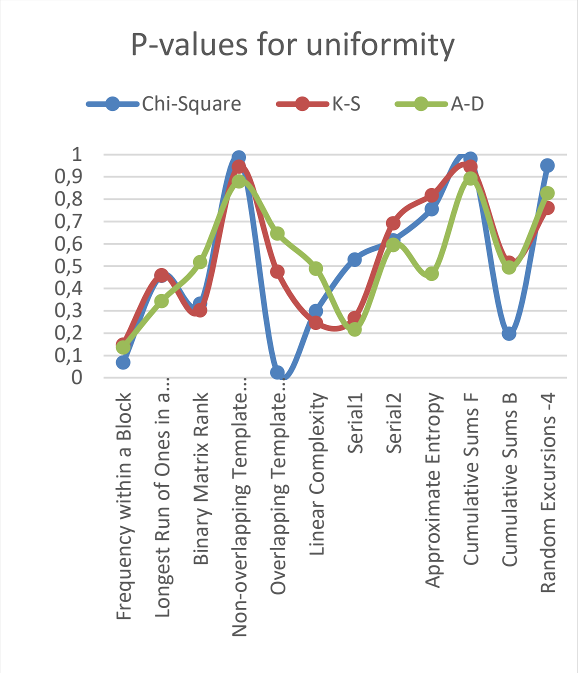


Fig. 6. The p-values for the tests that followed the Chi-Square distribution

The experiments demonstrate that the new testing methodology improve the detectability and reliability under the same or other test parameters.

6. CONCLUSIONS AND FURTHER DEVELOPMENT

This research has shown that the NIST test suite have to improve its testing method. In this study, NIST tests were used to detect deviations from randomness in a binary sequence. The project's first goal was to investigate how NIST STS data were interpreted. The second goal of the project was to provide various computation-friendly approximations and revisions in order to improve the results' interpretation. The project's final purpose was to use experimental results to investigate the proportion of passing sequences and the uniformity of P-value distribution. Experiments show that the new testing methodology boosts detectability and reliability in the same tests and similar parameters. The study's main contribution was to provide a review of the NIST Statistical Test Suite utilizing case studies with more precise values and uniformity tests in the second-level tests, all of which were then implemented in C-code.

Future Work. In order to establish an optimum approach for testing random numbers, our forthcoming projects will compare the benefits of several batteries or suites that have been researched in the literature, such as Diehard, Dieharder, Cryp-X, or ENT. Examining the NIST IR 6390 and NIST IR 6483 methodologies used to test the statistical properties of block algorithms would be another interesting extension to examine in the future. The samples generated by these methodologies will be directly subjected to statistical testing with the NIST SP 800-22 battery.

7. ACKNOWLEDGEMENT(S)

The author would like to express his gratitude to Diana Maimuţ, Mariana Costiuc, George Teşeleanu and Tiberiu Zbîrnea for their insightful comments.

8. REFERENCES

[1]. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., & Barker, E. (2001). *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Booz-allen and hamilton inc mclean va.

[2]. Bassham III, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., ... & Vo, S. (2010). Sp 800-22 rev. 1a. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology.

[3]. Kenny, C., Mosurski, K.: *Random Number Generators: An Evaluation and Comparison of Random.org and Some Commonly Used Generators.* Tech. rep., Computer Science Department, Trinity College Dublin (2005)

[4]. Pareschi, F., Rovatti, R., & Setti, G. (2012). *On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution*. IEEE Transactions on Information Forensics and Security, 7(2), 491-505.

[5]. Kim, S., Umeno, K., Hasegawa, A.: *Corrections of the NIST Statistical Test Suite for Randomness*. IACR Cryptology ePrint Archive 2004/018, 18 (2004)

[6]. Iwasaki, A. (2019). *Deriving the variance of the discrete Fourier transform test using Parseval's theorem*. IEEE Transactions on Information Theory, 66(2), 1164-1170.

[7]. Haramoto, H., & Matsumoto, M. (2019). *Checking the quality of approximation of p-values in statistical tests for random number generators by using a three-level test*. Mathematics and computers in simulation, 161, 66-75.

[8]. https://csrc.nist.gov/projects/random-bit-generation - sts-2.1.2

[9]. Sýs, M., Říha, Z., Matyas V., Marton, K., Suciu, A., (2015). *On the interpretation of results from the NIST statistical test suite*, Romanian Journal of Information Science and Technology, **18**, 1, 18–32

[10]. Simard, R., & L'Ecuyer, P. (2011). *Computing the two-sided Kolmogorov-Smirnov distribution*. Journal of Statistical Software, 39(11), 1-18.

[11]. Marsaglia, G., Tsang, W. W., & Wang, J. (2003). *Evaluating Kolmogorov's distribution*. Journal of statistical software, 8(18), 1-4.

[12]. Marsaglia, G., Tsang, W. W., & Wang, J.: C program to compute Kolmogorov's distribution. https://www.jstatsoft.org/article/view/v008i18

[13]. Marsaglia, G., & Marsaglia, J. (2004). *Evaluating the Anderson-Darling distribution.* Journal of statistical software, 9(2), 1-5.

[14]. Marsaglia, G., & Marsaglia, J.: AnDarl.c: C code. https://www.jstatsoft.org/article/view/v009i02

[15]. Sýs, M., Říha, Z.: *Optimised implementation of NIST STS* (2014). https://github.com/sysox/NIST-STS-optimised

[16]. https://www.idquantique.com/resource_type/random-number-generation/

ÎMBUNĂTĂȚAREA SUITEI DE TESTE NIST 800-22 REV.1A PRIN ADĂUGAREA DE DIVERSE CORECȚII ASUPRA TESTELOR ȘI ALTE TESTE GOODNESS-OF-FIT PENTRU A VERIFICA UNIFORMITATEA ÎN TESTELE DE NIVELUL AL DOILEA

Rezumat. Această lucrare studiază unele dintre cele mai semnificative considerații pentru alegerea și testarea generatoarelor de numere pseudoaleatoare. Procesul de testare este aplicat secvențelor de ieșire ale generatorului de numere aleatoare, cu scopul de a determina dacă numerele aleatoare se comportă statistic discret. Aleatorismul fluxului de biți este testat folosind raportul de evaluare sugerat de NIST 800-22 rev. Au fost publicate mai multe investigații privind fiabilitatea suitei de teste ale aleatorismului NIST, și anumite teste necesită corecții. În această lucrare se prezintă o trecere în revistă a suitei de teste statistice NIST prin implementarea multor modificări la testele de evaluare. Mai mult, a fost definit un interval mai precis de proporții acceptabile în cadrul proporției de secvențe de trecere. În testul de nivel al doilea, sunt implementate încă două teste Goodness of Fit (Kolmogorov-Smirnov și Anderson-Darling) pentru a îmbunătăți metodologia de testare a uniformității. Rezultatele studiilor prezentate în această lucrare arată că noua abordare de testare îmbunătățește detectabilitatea și fiabilitatea în condiții de testare identice sau diferite.

Elena-Iuliana GINGU (BOTEANU), PhD, University ”Politehnica” of Bucharest, Romania and Advanced Technologies Institute, e-mail: iuliana_boteanu@yahoo.com.