Efficient Online Computation of Mean, Median, and Standard Deviation as Continuous Health Tests for HRNGs

Sara Malik and N.A. Ahmed
PakCrypt NPO, Islamabad, Pakistan
smk@pakcrypt.org, nvd@tuta.io

***Abstract*—**Hardware random number generators (HRNGs) underpin the security of cryptographic systems, yet their physical entropy sources are susceptible to degradation, environmental perturbation, and adversarial manipulation. Continuous health testing during operation is therefore mandated by all major certification frameworks, including NIST SP 800-90B and BSI AIS 31. This survey examines the feasibility and efficiency of employing three classical statistical measures—mean, median, and standard deviation—as lightweight online health indicators for HRNG output streams. We ground our analysis in the Hotelling–Solomons inequality $|\mu - m| \leq \sigma$, which establishes a distribution-free bound linking these three statistics. We survey efficient streaming algorithms—including Welford's online variance computation, two-heap sliding-window median structures, and approximate quantile sketches—that enable their computation under the strict throughput and memory constraints of embedded cryptographic modules. We further address numerical stability considerations for long-running deployments processing billions of samples. Our analysis demonstrates that the mean–median–standard deviation triplet, combined with the Hotelling–Solomons bound, provides a complementary health test layer that fills the gap between the minimal repetition count and adaptive proportion tests of SP 800-90B and the comprehensive but offline NIST SP 800-22 statistical test suite.

***Keywords*—**Hardware random number generator, true random number generator, continuous health test, entropy source, mean–median inequality, Welford's algorithm, sliding-window median, NIST SP 800-90B, AIS 31, numerical stability

I. INTRODUCTION

The integrity of cryptographic protocols rests fundamentally on the quality of the random numbers they consume. Hardware random number generators (HRNGs)—also termed true random number generators (TRNGs)—derive randomness from physical phenomena such as thermal noise, shot noise, metastability in flip-flops, or jitter in ring oscillators [1], [2]. Unlike deterministic pseudorandom number generators, whose security relies on computational hardness assumptions, HRNGs must contend with the possibility that their physical entropy source may degrade or fail during operation due to environmental changes, component aging, manufacturing defects, or active adversarial interference [3], [4].

To mitigate this risk, all major cryptographic certification frameworks mandate continuous health testing of entropy sources during operation. The U.S. standard NIST SP 800-90B [5] prescribes two mandatory tests: the repetition count test (RCT), which detects a "stuck" source by flagging consecutive identical outputs, and the adaptive proportion test (APT), which monitors the frequency of the most common symbol within a sliding window. Both achieve O(1) time and memory per sample. The German BSI's AIS 31 framework [6], [7] takes a complementary approach, requiring a stochastic model of the noise source from which test parameters are derived. The FIPS 140-2 standard [8], now superseded by FIPS 140-3 [9] referencing ISO/IEC 19790:2012, historically required only a comparison of consecutive output blocks.

While the RCT and APT are computationally minimal and effective at detecting catastrophic failures (complete source collapse or extreme bias), they may be insufficient for detecting subtler distributional anomalies—gradual entropy degradation, emerging correlations, or distributional skew—that precede total failure. At the other extreme, comprehensive statistical test suites such as NIST SP 800-22 [10] and TestU01 [11] provide powerful but computationally expensive batch-mode evaluations unsuitable for real-time deployment.

Contribution. This survey proposes and analyses the use of three classical statistical measures—the sample mean, sample median, and sample standard deviation—as a lightweight continuous health test layer for HRNG output. The key enabling insight is the Hotelling–Solomons inequality [12], [13], which guarantees that for any distribution with finite variance, the absolute difference between the mean and median is bounded by the standard deviation. We demonstrate that:

(a) The mean and variance can be computed exactly in O(1) time and memory per sample using Welford's algorithm [14], with provable numerical stability suitable for processing billions of samples without reset.

(b) The sliding-window median can be maintained exactly in O(log n) time using two-heap structures [15] or order-statistics trees [16], or approximately in near-constant time using quantile sketches [17], [18].

(c) The Hotelling–Solomons bound provides a distribution-free consistency check: a violation of $|\mu - m| \leq \sigma$ in the population statistics is mathematically impossible for any well-behaved source, so its apparent violation in sample statistics at high confidence indicates a health test failure.

Organisation. Section II reviews the regulatory landscape for HRNG health testing. Section III presents the mathematical foundation of the mean–median inequality. Section IV surveys efficient streaming algorithms for computing the required statistics. Section V addresses numerical stability for long-running tests. Section VI discusses implementation trade-offs for embedded and hardware deployments. Section VII surveys related work. Section VIII concludes.

II. REGULATORY FRAMEWORK FOR HRNG HEALTH TESTING

A. NIST SP 800-90B

NIST Special Publication 800-90B [5], published in January 2018 by Turan, Barker, Kelsey, McKay, Baish, and Boyle, constitutes the primary U.S. standard governing entropy source validation for random bit generation. The document defines a comprehensive validation methodology comprising start-up testing, continuous health testing, and entropy estimation.

The continuous health tests are specified in Sections 4.4.1–4.4.2 of the standard. The *Repetition Count Test* (RCT) detects a stuck noise source by maintaining a counter of consecutive identical output samples. When C consecutive identical values are observed, where $C = \lceil 1 + (-\log_2 \alpha)/H \rceil$ with H denoting the assessed min-entropy per sample and $\alpha \approx 2^{-20}$ denoting the false-positive probability, the test signals failure. The *Adaptive Proportion Test* (APT) monitors the frequency of the most common value within a sliding window of W samples ($W = 1024$ for binary sources, $W = 512$ for non-binary). Both tests require O(1) memory and constant time per sample, establishing the baseline for computational efficiency that any proposed health test must meet or closely approach.

The standard further specifies ten entropy estimation methods, including the most common value estimate, the

collision estimate, the Markov estimate, and several compression-based and prediction-based estimators [19]. These are designed primarily for offline validation rather than continuous monitoring.

B. BSI AIS 20/31

The German Federal Office for Information Security (BSI) maintains the AIS 20/31 evaluation methodology [6], [7], [20], which defines functionality classes PTG.1, PTG.2, and PTG.3 for physical TRNGs with escalating security assurance requirements. A distinguishing feature of the BSI approach, in contrast to the NIST methodology, is the explicit requirement for a *stochastic model* of the entropy source. Test parameters and acceptance thresholds are derived from this model rather than being fixed constants. Schindler [21], [22] laid the theoretical foundation for this model-based evaluation, demonstrating that statistical tests tailored to the specific noise source achieve superior detection power compared to generic tests applied uniformly.

C. FIPS 140-2/3 and ISO/IEC 19790

The Federal Information Processing Standard FIPS 140-2 [8] specified a minimal continuous random number generator test in Section 4.9.2, requiring comparison of each n-bit output block to its predecessor; identical consecutive blocks triggered an error state. FIPS 140-3 [9], which became mandatory for new Cryptographic Module Validation Program (CMVP) submissions from April 1, 2022, references ISO/IEC 19790:2012 and delegates entropy source health testing to SP 800-90B. The NIST SP 800-140 series of documents supplements FIPS 140-3 for CMVP evaluation.

D. Limitations of existing mandatory tests

The RCT and APT tests mandated by SP 800-90B are designed for high detection probability against catastrophic failures at extremely low false-positive rates. However, they are not designed to detect gradual distributional drift, slowly increasing bias, emerging autocorrelation, or changes in the higher moments of the output distribution. The comprehensive offline suites—SP 800-22 [10], Marsaglia's Diehard battery [23], L'Ecuyer and Simard's TestU01 [11], and the tests described by Knuth [24]—possess the statistical power to detect such anomalies but require batch processing of large sample collections. This creates a detection gap between the lightweight mandatory tests and the comprehensive offline suites, motivating the investigation of intermediate-complexity online tests based on distributional statistics.

III. THE MEAN–MEDIAN INEQUALITY

A. Statement and history

Let X be a random variable with finite mean $\mu = \mathrm{E}[X]$ and finite variance $\sigma^2 = \mathrm{Var}(X)$. Let m denote any median of X. Then:

$$|\mu - m| \leq \sigma \tag{1}$$

This result was first established by Hotelling and Solomons [12] in 1932 as a corollary of their bounds on Pearson's skewness coefficient. The inequality is variously referred to as the Hotelling–Solomons inequality or the mean–median–standard deviation bound in the literature.

O'Cinneide [13] provided a particularly clear exposition in 1990, giving an explicit proof via an analysis-of-variance decomposition and establishing the conditions for equality: the bound is attained if and only if the distribution is a two-point mass concentrated at $\mu - \sigma$ and $\mu + \sigma$ with probabilities chosen so that one of these points is the median. O'Cinneide further established a generalisation to arbitrary quantiles: for the p-th percentile x_p, we have $|\mu - x_p| \leq \sigma\sqrt{(p/(1-p))}$, which reduces to (1) when $p = 1/2$.

B. Proof via Jensen's inequality

The most elegant proof, due to Mallows [25], proceeds in three steps exploiting standard inequalities. We reproduce it here for completeness, as it illuminates the computational strategy for health testing.

$$|\mu - m| =$$
$$|\mathrm{E}[X - m]| \leq \mathrm{E}|X - m| \leq \mathrm{E}|X - \mu| \leq \sqrt{(\mathrm{E}[(X-\mu)^2])} = \sigma \tag{2}$$

The first inequality follows from convexity of the absolute value function (Jensen's inequality). The second follows from the characterisation of the median as the minimiser of the expected absolute deviation: $\mathrm{E}|X - a|$ is minimised at $a = m$. The third inequality is the Cauchy–Schwarz inequality (equivalently, the RMS–mean absolute deviation inequality). This chain of reasoning is central to our proposed health test: it shows that computing μ, m, and σ suffices to verify a universal distributional invariant.

C. Sharpened bounds and related results

Maruyama [26] recently established a sample-size-dependent refinement, showing that the ratio $|\mu - m|/\sigma$ is strictly bounded below 1 for distributions that are not two-point masses. Related results include the mode–median–mean ordering for unimodal distributions studied by Groeneveld and Meeden [27], and the analysis of Pearson's rule of thumb (mean − mode ≈ 3(mean − median)) by von Hippel [28], who demonstrated that this approximation fails for many common distributions. The one-sided Chebyshev (Cantelli) inequality provides an alternative route to (1): $\mathrm{P}(X - \mu \geq k\sigma) \leq 1/(1 + k^2)$;

setting $k = 1$ yields $P(X \geq \mu + \sigma) \leq 1/2$, and since the median satisfies $P(X \geq m) \geq 1/2$, the bound $m \leq \mu + \sigma$ follows.

D. Application to HRNG health testing

For an ideal binary HRNG producing independent identically distributed (i.i.d.) uniform bits, the expected output byte values follow the binomial distribution $B(8, 1/2)$ with $\mu = 4$ and $\sigma \approx 1.414$. The median is 4, and $|4 - 4| = 0 \ll \sigma$. A degraded source with probability bias $p \neq 1/2$ shifts the mean to $8p$ and alters the variance to $8p(1 - p)$, which the mean and variance monitors can detect. If the source develops non-trivial dependence structures (e.g., Markov correlations), the empirical distribution of byte values will deviate from the binomial model, potentially causing $|\hat{\mu} - \hat{m}|$ to grow relative to $\hat{\sigma}$. When the sample statistic $|\hat{\mu} - \hat{m}|$ exceeds $\hat{\sigma}$ by a margin that cannot be attributed to sampling fluctuation at the desired confidence level, a health test alarm should be raised.

IV. EFFICIENT STREAMING ALGORITHMS

A. Online mean and variance: Welford's algorithm

Welford [14] introduced a numerically stable single-pass algorithm for computing the running mean and variance. Given a stream $x_1, x_2, \ldots$ of samples, the algorithm maintains two accumulators M_k (running mean) and S_k (running sum of squared deviations) via the recurrences:

$$M_1 = x_1, \qquad M_k = M_{k-1} + (x_k - M_{k-1})/k \tag{3}$$

$$S_1 = 0, \qquad S_k = S_{k-1} + (x_k - M_{k-1})(x_k - M_k) \tag{4}$$

The sample variance is then $s^2 = S_k/(k - 1)$. This algorithm requires exactly three floating-point values (M_k, S_k, and k) and performs one subtraction, one division, one multiplication, and one addition per sample—achieving O(1) time and O(1) space. Knuth [24, §4.2.2] popularised this method and credited Welford. Chan, Golub, and LeVeque [29] provided a rigorous rounding-error analysis comparing Welford's algorithm to the naïve sum-of-squares formula, the two-pass algorithm, and their own pairwise algorithm. They demonstrated that Welford's algorithm achieves relative error proportional to the machine epsilon ε times the condition number $\kappa = (\mu^2 + \sigma^2)/\sigma^2$, which is excellent when the coefficient of variation μ/σ is not excessively large.

For sliding-window variants where one must also *remove* the oldest sample, the circular-buffer approach maintains running sums Σ and Σ^2 and subtracts the departing value. West [30] and the pairwise algorithm of Chan et al. [31] extend the approach to weighted and parallel settings, respectively.

B. Exact sliding-window median

The sliding-window median problem requires maintaining the median of the most recent n samples as new values arrive and old values expire. The standard approach uses a *two-heap structure*: a max-heap stores the lower half of the window, and a min-heap stores the upper half, with the invariant that heap sizes differ by at most one. The median is retrieved from the top of the larger heap (or the average of both tops for even-sized windows) in O(1) time. Insertion requires O(log n) time for a heap push plus potential rebalancing. For sliding windows, deletion of the expired element is handled via lazy deletion with a hash map tracking pending removals; invalid elements are popped when they appear at the heap top. This construction is a natural application of priority queue theory as presented in Cormen et al. [15].

Order-statistics trees—balanced binary search trees augmented with subtree size counts—provide an alternative with cleaner deletion semantics. The rank operation locates the median in O(log n) time, and deletion is also O(log n). These structures are described in Chapter 14 of Cormen et al. [15] (3rd edition) and are available in some standard libraries (e.g., the GNU C++ policy-based data structures).

C. Approximate quantile computation

When exact median computation is too expensive—particularly in hardware implementations or under extreme throughput requirements—approximate quantile algorithms offer a practical alternative with tunable accuracy–memory trade-offs.

The Greenwald–Khanna (GK) algorithm [17] achieves ε-approximate quantile estimation (the returned value has rank

Algorithm	Time/sample	Memory	Exact?	Notes
Welford (mean/var.)	O(1)	O(1)	Yes	All environments
Two-heap median	O(log n)	O(n)	Yes	Software, moderate n
Order-statistics tree	O(log n)	O(n)	Yes	Cleaner deletion
Greenwald–Khanna	O(1) amort.	O(1/ε · log εN)	ε-approx	Memory-constrained
t-digest	O(log n) am.	O(δ)	ε-approx	Tail-sensitive
KLL sketch	O(1) amort.	O(1/ε · loglog 1/δ)	ε-approx	Optimal space

Table I: Algorithmic complexity comparison for continuous health test statistics

within $\pm\varepsilon N$ of the true quantile rank, where N is the stream length) using $O((1/\varepsilon)\log(\varepsilon N))$ space with amortised constant time per insertion. The t-digest algorithm of Dunning and Ertl [18] achieves extreme accuracy at the distribution tails while maintaining a compact representation through adaptive centroid clustering; it has been widely deployed in production systems including Elasticsearch and Apache Lucene. For optimal space complexity, the KLL sketch of Karnin, Lang, and Liberty [32] achieves randomised ε-approximate quantiles in $O((1/\varepsilon)\log\log(1/\delta))$ space, matching the lower bound up to the log log factor.

For the sliding-window setting specifically, the foundational framework of Datar, Gionis, Indyk, and Motwani [33] on maintaining stream statistics over sliding windows, extended by Babcock, Datar, Motwani, and O'Callaghan [34] to variance maintenance, provides the theoretical underpinning.

V. NUMERICAL STABILITY FOR LONG-RUNNING HEALTH TESTS

Continuous TRNG health monitoring may process on the order of 10^9–10^{12} samples without reset in high-throughput cryptographic applications. Accumulated floating-point round-off presents a non-trivial engineering concern at these scales.

Welford's algorithm already provides substantial built-in stability because it operates on deviations from a running mean rather than accumulating raw sums of squares—a key advantage over the naïve textbook formula $\sigma^2 = (1/n)\Sigma x^2_i - \bar{x}^2$, which suffers from catastrophic cancellation when $\sigma \ll \mu$ [29]. For additional robustness, the Kahan summation algorithm [35] maintains a compensation variable tracking lost low-order bits, achieving accumulated error bounds effectively independent of the summation length. Neumaier [36] improved the Kahan approach for cases where the addend exceeds the running sum, yielding the Kahan–Babuška–Neumaier (KBN) algorithm. Klein [37] further extended this with higher-order recursive error compensation. The definitive analysis of these techniques appears in Higham [38].

For the specific application of TRNG health testing, we note that HRNG output values typically occupy a small dynamic range (e.g., 0–255 for 8-bit samples), so the condition number $\kappa = (\mu^2 + \sigma^2)/\sigma^2$ remains moderate. The primary numerical risk is therefore not catastrophic cancellation but gradual drift in the running sum. We recommend the following practical strategy: employ Welford's algorithm with Kahan-compensated accumulation of the correction terms in S_k and M_k, or alternatively use the Chan–Golub–LeVeque pairwise algorithm [31] when parallel processing of sub-windows is desired. Integer arithmetic is a further option for small-range TRNG outputs (8–16 bit), entirely eliminating floating-point concerns at the cost of larger accumulator widths (64-bit integers suffice for 10^1 samples of 8-bit values).

VI. IMPLEMENTATION CONSIDERATIONS

A. Embedded and hardware deployments

The deployment environment for TRNG health tests spans a wide spectrum: from software-based monitoring on general-purpose processors to fully integrated hardware tests within FPGA or ASIC implementations of the entropy source itself. The work of Veljković, Rožić, and Verbauwhede [39] on low-cost implementations of on-the-fly tests and Yang et al. [40] on embedded HW/SW platforms for TRNG testing provides direct precedent.

For the mean–variance computation via Welford's algorithm, hardware implementation requires a single multiply-accumulate (MAC) unit and three registers—resource requirements comparable to the existing APT. The median computation is more demanding. For a typical window of $W = 1024$ samples at 8 bits each, an exact order-statistics tree requires approximately 20–30 KB including node overhead—feasible in software but expensive in dedicated logic.

The Hotelling–Solomons inequality provides an elegant resolution to this tension: one can compute the mean and standard deviation in O(1) space, obtain even a rough median estimate (e.g., via a histogram-based approximation or a P^2 algorithm), and verify that the estimate falls within $[\mu - \sigma, \mu + \sigma]$. This converts the median check from an exact computation problem into a consistency verification problem, where even a coarse median estimate suffices to detect a health test violation. For an ideal uniform-random byte source, the nominal mean is 127.5 and the nominal standard deviation is approximately 73.9; the bound $[\mu - \sigma, \mu + \sigma]$ spans nearly the entire range [0, 255], making violation extremely unlikely under normal operation and thus providing a highly specific alarm.

B. Tibshirani's search-space reduction

Tibshirani [41] exploited the mean–median inequality directly as an algorithmic tool: given μ and σ, the median must lie in $[\mu - \sigma, \mu + \sigma]$, reducing the search space for median computation via successive binning. This observation is directly applicable to our setting: after computing the running mean and standard deviation via Welford's algorithm, one can restrict a histogram-based median approximation to the interval $[\mu - \sigma, \mu + \sigma]$, substantially reducing the number of bins required.

C. Commercial reference implementations

The Intel Digital Random Number Generator (DRNG) architecture [42] provides a commercial reference for integrated TRNG health testing. The Intel implementation employs an online entropy estimator as part of its conditioner, with hardware-level health monitoring that triggers reseed operations when statistical anomalies are detected. Sunar, Martin, and Stinson [2] established the provable-security framework for ring-oscillator TRNGs, and Baudet, Lubicz, Micolod, and Tassiaux [43] deepened the security analysis with entropy rate evaluation and bias analysis. Cherkaoui, Fischer, Fesquet, and Aubert [44] demonstrated a 200 Mbit/s TRNG design with built-in entropy assessment, and more recently Lubicz and Fischer [45] provided a complete algorithmic framework for entropy rate computation in oscillator-based designs.

VII. RELATED WORK

The broader context of TRNG testing includes several lines of related research. Maurer [46] proposed a universal statistical test capable of detecting any significant deviation from randomness for ergodic stationary sources with finite memory, providing the information-theoretic foundation for entropy-based online testing. Skórżki [47] demonstrated that min-entropy can be estimated with substantially fewer samples than specified in SP 800-90B under the i.i.d. assumption.

Recent work by Johnston [48] proposed a lightweight "polygon test" for online entropy boundary testing of non-i.i.d. data, identifying specific shortcomings in SP 800-90B's non-i.i.d. tests. Zhu, Ma, Chen, Lin, and Jing [49] analysed and improved the entropy estimators in SP 800-90B for non-i.i.d. entropy sources. These contributions reinforce the motivation for additional lightweight online tests that can complement the mandated RCT and APT.

On the implementation side, Bucci and Luzzi [50] addressed the design of testable random bit generators with integrated self-test capabilities, and Balasch et al. [51] bridged the gap between academic testing methodologies and industry certification requirements. Barak, Shaltiel, and Tromer [4] constructed efficient randomness extractors resilient to bounded adversarial manipulation of the entropy source, and Dichtl [3] demonstrated practical attack scenarios against real hardware generators.

The side-channel analysis community has also found the mean–median relationship useful: in power analysis attacks and leakage detection tests [52], the choice between mean-based and median-based statistics for trace characterisation is informed precisely by bounds such as (1), which quantify how far these estimators may diverge. Saarinen [53] studied entropy and bit patterns in ring oscillator jitter, directly relevant to the statistical monitoring of oscillator-based TRNG output.

VIII. CONCLUSION

We have surveyed the theoretical foundations, algorithmic building blocks, and practical implementation considerations for employing the mean, median, and standard deviation as continuous online health tests for hardware random number generators. The Hotelling–Solomons inequality $|\mu - m| \leq \sigma$ provides a distribution-free, mathematically guaranteed consistency check linking these three statistics. When combined with Welford's O(1)-memory online mean/variance algorithm and either an exact two-heap or an approximate quantile sketch for the median, this bound yields a health test that:

(1) operates in constant time and near-constant memory per sample for the mean and variance, with O(log n) time for the exact median or O(1) amortised time for an approximate median;

(2) detects distributional anomalies—bias, correlation, skew, and variance changes—that the mandatory repetition count and adaptive proportion tests of NIST SP 800-90B are not designed to capture;

(3) provides a provable guarantee: any violation of the bound signals a fundamental departure from stationarity that warrants investigation, independent of distributional assumptions;

(4) maintains numerical stability over arbitrarily long operational lifetimes when implemented with Welford's algorithm and, optionally, Kahan-compensated summation.

Future work should address the empirical calibration of detection thresholds for the sample version of the mean–median inequality under specific TRNG output models, the formal derivation of false-positive and false-negative rates as functions of window size and entropy level, and the hardware synthesis cost of the proposed test for FPGA and ASIC implementations.

REFERENCES

[1] V. Fischer, "A closer look at security in random number generators design," in Proc. COSADE 2012, LNCS, vol. 7275, pp. 167–182, Springer, 2012.

[2] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," IEEE Trans. Comput., vol. 56, no. 1, pp. 109–119, Jan. 2007.

[3] M. Dichtl, "How to predict the output of a hardware random number generator," in Proc. CHES 2003, LNCS, vol. 2779, pp. 181–188, Springer, 2003.

[4] B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," in Proc. CHES 2003, LNCS, vol. 2779, pp. 166–180, Springer, 2003.

[5] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," NIST SP 800-90B, Jan. 2018.

[6] W. Killmann and W. Schindler, "A proposal for: Functionality classes and evaluation methodology for true (physical) random number generators," BSI AIS 31, ver. 3, May 2020.

[7] W. Schindler and W. Killmann, "Evaluation criteria for true (physical) random number generators used in cryptographic applications," in Proc. CHES 2002, LNCS, vol. 2523, pp. 431–449, Springer, 2003.

[8] National Institute of Standards and Technology, "Security requirements for cryptographic modules," FIPS PUB 140-2, May 2001.

[9] National Institute of Standards and Technology, "Security requirements for cryptographic modules," FIPS PUB 140-3, Mar. 2019.

[10] A. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST SP 800-22 Rev. 1a, Apr. 2010.

[11] P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," ACM Trans. Math. Softw., vol. 33, no. 4, art. 22, 2007.

[12] H. Hotelling and L. M. Solomons, "The limits of a measure of skewness," Ann. Math. Statist., vol. 3, no. 2, pp. 141–142, 1932.

[13] C. O'Cinneide, "The mean is within one standard deviation of any median," Amer. Statist., vol. 44, no. 4, pp. 292–293, 1990.

[14] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," Technometrics, vol. 4, no. 3, pp. 419–420, 1962.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 4th ed. Cambridge, MA, USA: MIT Press, 2022.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. Cambridge, MA, USA: MIT Press, 2009, ch. 14.

[17] M. Greenwald and S. Khanna, "Space-efficient online computation of quantile summaries," in Proc. ACM SIGMOD, pp. 58–66, 2001.

[18] T. Dunning and O. Ertl, "Computing extremely accurate quantiles using t-digests," arXiv:1902.04023, 2019.

[19] J. Kelsey, K. A. McKay, and M. S. Turan, "Predictive models for min-entropy estimation," in Proc. CHES 2015, LNCS, vol. 9293, pp. 373–392, Springer, 2015.

[20] W. Schindler, "Evaluation criteria for physical random number generators," in Security in Pervasive Computing, LNCS, vol. 3450, pp. 393–407, Springer, 2005.

[21] W. Schindler, "Efficient online tests for true random number generators," in Proc. CHES 2001, LNCS, vol. 2162, pp. 103–117, Springer, 2001.

[22] W. Killmann and W. Schindler, "A design for a physical RNG with robust entropy estimators," in Proc. CHES 2008, LNCS, vol. 5154, pp. 146–163, Springer, 2008.

[23] G. Marsaglia, "The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness," Florida State Univ., 1995.

[24] D. E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd ed. Reading, MA, USA: Addison-Wesley, 1997.

[25] C. L. Mallows, "Another comment on O'Cinneide," Amer. Statist., vol. 45, no. 3, p. 257, 1991.

[26] Y. Maruyama, "A sharper bound of the Hotelling–Solomons inequality," Stat, vol. 13, e710, 2024.

[27] R. A. Groeneveld and G. Meeden, "The mode, median, and mean inequality," Amer. Statist., vol. 31, no. 3, pp. 120–121, 1977.

[28] P. T. von Hippel, "Mean, median, and skew: Correcting a textbook rule," J. Statist. Educ., vol. 13, no. 2, 2005.

[29] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Algorithms for computing the sample variance: Analysis and recommendations," Amer. Statist., vol. 37, no. 3, pp. 242–247, 1983.

[30] D. H. D. West, "Updating mean and variance estimates: An improved method," Commun. ACM, vol. 22, no. 9, pp. 532–535, 1979.

[31] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Updating formulae and a pairwise algorithm for computing sample variances," Stanford Univ., Tech. Rep. STAN-CS-79-773, 1979.

[32] Z. Karnin, K. Lang, and E. Liberty, "Optimal quantile approximation in streams," in Proc. IEEE FOCS, pp. 71–78, 2016.

[33] M. Datar, A. Gionis, P. Indyk, and R. Motwani, "Maintaining stream statistics over sliding windows," SIAM J. Comput., vol. 31, no. 6, pp. 1794–1813, 2002.

[34] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan, "Maintaining variance and k-medians over data stream windows," in Proc. ACM PODS, pp. 234–243, 2003.

[35] W. Kahan, "Pracniques: Further remarks on reducing truncation errors," Commun. ACM, vol. 8, no. 1, p. 40, 1965.

[36] A. Neumaier, "Rundungsfehleranalyse einiger Verfahren zur Summation endlicher Summen," ZAMM, vol. 54, no. 1, pp. 39–51, 1974.

[37] A. Klein, "A generalized Kahan–Babuška–summation–algorithm," Computing, vol. 76, pp. 279–293, 2006.

[38] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed. Philadelphia, PA, USA: SIAM, 2002.

[39] F. Veljković, B. Rožić, and I. Verbauwhede, "Low-cost implementations of on-the-fly tests for random number generators," in Proc. DATE, pp. 959–964, IEEE, 2012.

[40] B. Yang, B. Rožić, N. Mentens, W. Dehaene, and I. Verbauwhede, "Embedded HW/SW platform for on-the-fly testing of true random number generators," in Proc. DATE, pp. 345–350, IEEE, 2015.

[41] R. J. Tibshirani, "Fast computation of the median by successive binning," arXiv:0806.3301, 2008.

[42] Intel Corp., "Intel digital random number generator (DRNG) software implementation guide," 2012. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/guide/intel-digital-random-number-generator-drng-software-implementation-guide.html

[43] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," J. Cryptol., vol. 24, pp. 398–425, 2011.

[44] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, "A very high speed true random number generator with entropy assessment," in Proc. CHES 2013, LNCS, vol. 8086, Springer, 2013.

[45] D. Lubicz and V. Fischer, "Entropy computation for oscillator-based physical random number generators," J. Cryptol., vol. 37, no. 2, p. 13, 2024.

[46] U. M. Maurer, "A universal statistical test for random bit generators," J. Cryptol., vol. 5, no. 2, pp. 89–105, 1992.

[47] M. Skórżki, "Evaluating entropy for true random number generators: Efficient, robust and provably secure," in Proc. CT-RSA 2017, LNCS, vol. 10159, Springer, 2017.

[48] D. Johnston, "Online testing entropy and entropy tests with a two state Markov model," in Proc. SPACE 2024, LNCS, vol. 15351, Springer, 2025.

[49] S. Zhu, Y. Ma, T. Chen, J. Lin, and J. Jing, "Analysis and improvement of entropy estimators in NIST SP 800-90B for non-IID entropy sources," IACR Trans. Symmetric Cryptol., vol. 2017, no. 3, pp. 151–168, 2017.

[50] M. Bucci and R. Luzzi, "Design of testable random bit generators," in Proc. CHES 2005, LNCS, vol. 3659, pp. 131–146, Springer, 2005.

[51] J. Balasch et al., "Design and testing methodologies for true random number generators towards industry certification," in Proc. European Test Symp. (ETS), pp. 1–10, IEEE, 2018.

[52] F. Bernard, D. Mollinedo Garay, M. Haddad, N. Bochard, and V. Fischer, "Low cost and precise jitter measurement method for TRNG entropy assessment," IACR Trans. Cryptogr. Hardw. Embed. Syst. (TCHES), vol. 2024, no. 1, pp. 207–228, 2024.

[53] M.-J. O. Saarinen, "On entropy and bit patterns of ring oscillator jitter," arXiv:2102.02196, 2021.