STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **8**, December 2020, pp 810–821.
Published online in International Academic Press (www.IAPress.org)

Generalized Self-Similar First Order Autoregressive Generator (GSFO-ARG) for Internet Traffic

Jumoke Popoola [1,*], Waheed Babatunde Yahya [1], Olusogo Popoola [2], Oyebayo Ridwan Olaniran[1]

[1]*Department of Statistics, Faculty of Physical Sciences, University of Ilorin, Nigeria*
[2]*Department of Computer Engineering, Faculty of Engineering and Technology, University of Ilorin, Nigeria*

Abstract Internet traffic data such as the number of transmitted packets and time spent on the transmission of Internet protocols (IPs) have been shown to exhibit long memory property, often referred to as self-similarity. Simulating this type of dataset is an important aspect of delay avoidance planning, especially when trying to mimic real-life processing of packets on the Internet. Most of the existing procedures often assumed the process follows a Gaussian distribution, and thus long memory processes such as Fractional Brownian Motion (FBM) and Fractional Gaussian Noise (FGN) among others are used. These approaches often result in estimation errors arising from the use of inappropriate distribution. However, it has been established that the distribution of Internet processes are heavy-tailed. Therefore, in this paper, a new method that is capable of generating heavy-tailed self-similar traffic is proposed based on the first-order autoregressive AR (1) process. The proposed method is compared with some of the existing methods at varying values of the self-similar index and sample sizes. The imposed self-similarity indices were estimated using the Range/Standard deviation statistic (R/S). Performance analysis was achieved using the absolute percentage errors. The results showed that the proposed method has a lower average error when compared to other competing methods.

Keywords Self-similar index, Internet traffic packet, Gaussian process, Hurst parameter

AMS 2010 subject classifications 60G18, 60G15

DOI: 10.19139/soic-2310-5070-926

1. Introduction

Scientic experimentation has gained a powerful tool with the advent of computers. Nowadays, their immense processing capabilities are used to perform complex simulations of data in various real-world situations. In a few seconds, the (almost) tireless machines are capable of simulating and testing panoply of conditions that a man would take a lifetime to reproduce. Supported by the law of large numbers, their outputs are often considered as decision aids or as scientic arguments, if the codied simulation and resulting built model is guaranteed to be free of errors [9].

The term self-similarity was first described by Hurst [8] in his reports about the Nile River. It is now known to be embedded in many of the processes relating to natural and articial events [9], with its popularity that may be traced to the ndings that unfold the so-called self-similar nature of Internet traffic. This has brought attention to the fact that the behaviour of trafc in the network aggregation points should be understood by considering the self-similar and non-memoryless property. Quantication of self-similarity is often done by estimating the parameter known as the Hurst parameter described in [22], the simulation of sequences with the aforementioned property constitutes a rather defying task, mostly because of the retrospective nature of its denition. The generator of a sequence with

*Correspondence to: Jumoke Popoola (Email: jmkbalpop@unilorin.edu.ng). Department of Statistics, Faculty of Physical Sciences, University of Ilorin, P.M.B. 1515, Ilorin, Nigeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

the self-similarity structure requires the produced points to be orderly stored in the memory and processed before further points are created.

As a matter of retrospective, we investigated some papers that employed simulated Internet data in the form of voice, text, video, or their combinations to carry out various research on Internet traffic. Among the works reviewed, the majority had the problem of simulating Internet traffic data using wrong models which may not adequately capture the true self-similar and long-range dependent properties of Internet traffic. For example, the use of Fractional Brownian Motion (FBM) process for modelling self-similarity in stock market prices [12]. The same process was suggested in [13] for modelling connectionless communications with parameters based on an equivalent burst model. Also, more recent authors, [23, 7] constructed algorithms for simulating Internet data that are based on simulating Gaussian processes and Markov Decision Processes. The drawback of the approach lies in its reliance on Gaussian process.

[23] proposed a method for detecting network anomalies and intrusions in communication networks based on support vector machine (SVM) and broad learning system (BLS) to detect anomalies and intrusions in datasets obtained from packet data networks. The authors used the SVM algorithm as a supervised learning approach to identify optimal hyperplanes (decision boundaries) for classifying known network anomalies and intrusions. The authors further compared the algorithm using the polynomial (linear, quadratic, and cubic), Gaussian, and sigmoid kernels. The developed models are trained and tested using data from the Internet routing tables, a simulated air force base network, and an experimental testbed. As robust as the approach, the demerit is the reliance on Gaussian process, which is unrealistic in most Internet traffic data. To bridge this gap, we proposed a generalized self-similar Internet traffic generator that is useful for generating self-similar processes from any probability distribution.

[7] worked on a virtual network embedding (VNE) via a Monte Carlo tree search using two VNE algorithms: MaVEn-M and MaVEn-S. MaVEn-M employs the multi-commodity flow algorithm for virtual link mapping, while MaVEn-S uses the shortest-path algorithm. The cog in the wheel of the approach is that it is based on the Markov Decision Process (MDP) framework and devises action policies (node mappings) using the Monte Carlo tree search algorithm. Besides, the VNE approach lacks a thorough justification of the Virtual Node Mapping (VNoM) procedure used. Furthermore, the MDP method used cannot adequately capture the self-similar property of Internet traffic.

In [10], a similar and extended work of [7] considered network topology and energy efficiency in the VNE process based on Energy Efficient, Concurrent, and Topology-Aware (EE-CTA). The shortcoming of EE-CTA is that the generated substrate network and virtual network data used for the analysis is based on the same predetermined MDP as used in [7].

Besides network-based approaches, fitness-based dynamic virtual network embedding (DYVINE) have also been proposed to solve the problem of poor usage and extremely low acceptance rate in VNE for efficient mapping to support the dynamic nature of incoming virtual networks [4]. The major demerit of [4] is the simulation setup for the arrival and lifetime of the incoming virtual networks, which is assumed to follow the Poisson and exponential distributions, respectively. These distributions are not adequate to explain the self-similar property of Internet traffic with its non-memoryless characteristic. Given the aforementioned flaws in the various existing self-similar Internet traffic generators, we propose a new generalized method for simulating self-similar processes based on Box-Jenkins autoregressive moving average (ARMA) of order 1, AR (1). The modified version of the AR(1) process is used to build an algorithm for a generator that is applicable for simulating self-similar processes that is adaptable to any distribution. The new method is also readily suitable for simulating Internet traffic which may not necessarily follow Gaussian distribution or Poisson-Exponential process.

The main result of this paper is a theorem presenting the derived Internet traffic generator proposed as well as the efficiency of the model using simulated data. In particular, self-similar traffic from a mixture of both Gaussian and non-Gaussian processes were simulated. In addition, we presented the performance comparison of the proposed generator in the simulation of exact self-similar Internet traffic with some competing techniques such as Paxson, Fractional Gaussian Noise (FGN), Hosking method, Fractional Brownian Motion (FBM) processes, Cholesky method, and Fractional Autoregressive Integrated Moving Average (F-ARIMA).

The rest of the paper is organized as follows; Section 2 gives preliminaries of self-similar processes. Section 3 presents the main results of this paper. Section 4 presents the numerical examples from simulations. Section 5 presents the discussion of results while the conclusion and future work are presented in Section 6.

2. Development of self-similar process model

In this section, a brief consideration of some terminologies related to Internet traffic and particularly heavy-tailed distribution is presented.

2.1. Self-similarity

A self-similar object has been described to be exactly or approximately similar to a part of itself [16, 17, 18]. Self-similarity in Internet traffic occurs when packets of the same burst length arrive at the same time or when packets burst at the same inter-arrival period on the server [14, 16, 18]. Simply speaking, an Internet traffic process that exhibits self-similarity implies that the process is indistinguishable from its scaled versions, which are obtained by averaging the original process within different observation time scales. The mathematical description of self-similarity is given by [1, 16].

$$X_j^m = \frac{1}{m}(X_{jm-m+1}, +X_{jm-m+2} + \cdots + X_{jm}) \tag{1}$$

Equation (1) describes an incremental process $X_j (j = 1, 2, \dots)$ whose average in non-overlapping blocks of size m is another process $X_j^m (j = 1, 2,)$.Here, process X_j is said to be self-similar if $X_j^m \equiv m^{H-1} X_j$ where $m(m \geq 1)$ is the scale parameter, and $H, (0.5 < H \leq 1)$ is the Hurst parameter, which is used to measure the burstiness of an Internet traffic process. Obviously, when $H = 1$, processes X_j^m and X_j have the same distribution without any decay since $var(X_j^m) = var(X_j)$.

2.2. Long-Range Dependence (LRD)

A self-similar process can also contain a property of long-range dependence [25]. Long-Range Dependence (LRD) describes the memory effect where a current value strongly depends upon the past values of a stochastic process, and it is characterized by its autocorrelation function at various time points usually called the lag points. Given the Hurst parameter H for $0 < H < 1, H \neq 0.5$ the autocorrelation function $r(k)$ for lag k is;

$$r(k) = H(2H - 1)k^{-2H-2} \tag{2}$$

For values $0.5 < H < 1$, the autocorrelation function $r(k)$ decays hyperbolically to ck^{-2H-2} as k increases, which means that the autocorrelation function is not summable [6]. This is opposite to the property of short-range dependence (SRD), where the autocorrelation function decays exponentially, and the equation (2) has a finite value. Short and long-range dependence have a common relationship with the value of the Hurst parameter (H) of the self-similar process [14, 26, 16, 17, 18, 11, 20, 19, 21]. When the value of H lies in the interval $0 < H < 0.5$, the self-similar process is said to be SRD and if the value of H lies in the interval $0.5 < H < 1$ the process is said to have LRD.

The reviewed classical approaches that captures the behaviour of Internet traffic are detailed in the subsections that follows.

2.3. Fractional Brownian Motion and Fractional Gaussian Noise

The Fractional Brownian Motion (FBM) described by [12] is a Gaussian stochastic process in continuous time defined as:

$$B_H(t) = B_H(0) + \frac{1}{\Gamma(H + 1/2)} \left\{ \int_{-\infty}^{0} [(t - s)^{H-1/2} - (-s)^{H-1/2}] \right\} dB(s) + \int_{-\infty}^{0} (t - s)^{H-1/2} dB(s) \tag{3}$$

for $t > 0$ (and similarly for $t < 0$) where $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} \exp(-x)dx$ and $0 < H < 1$. The FBM $B_H(t)$ has the following covariance function:

$$E[B_H(t)B_H(s)] = 1/2(t|^{2H} + |s|^{2H} - |t-s|^{2H}) \tag{4}$$

The Hurst exponent describes the raggedness of the resultant motion with a higher value leading to smoother motion. The value of H determines what kind of process the FBM is. For clarity, it was reported in [12] that:

i. if $H = 1/2$ then the process is, in fact, a Brownian motion or Wiener process;
ii. if $H > 1/2$ then the increments of the process are positively correlated;
iii. if $H < 1/2$, then the increments of the process are negatively correlated.

The increment process, $X = X_k : k = 0, 1, ...$ of Fractional Brownian Motion is known as Fractional Gaussian Noise (FGN) defined by;

$$X(t) = B_H(t+1) - B_H(t) \tag{5}$$

It is clear that X_k has a standard normal distribution for every k, but that there is (in general) no independence. To be precise, the corresponding auto-covariance function $\gamma()$ is of the form

$$\gamma(k) = \frac{1}{2}[|k-1|^{2H} - 2|k|^{2H} + |k+1|^{2H}] \tag{6}$$

for $k \in Z$. If $H = 1/2$, all the covariances are 0 (except, of course, for $k = 0$). Since Fractional Gaussian Noise is a Gaussian process, this implies independence. This agrees with the properties of ordinary Brownian Motion, which has independent increments.

2.4. Fractional Autoregressive Moving Average Process

Another widely used process with long-range dependence is Fractional Autoregressive Moving Average Process (F-ARMA). The parameters of this model control long-range dependence as well as the short term behaviour. The F-ARIMA model is based on the ARMA model [5]. An ARMA (p, q) process $X = X_k : k = 0, 1, ...$ is a short memory process that is the solution of;

$$\phi(L)X_k = \theta(L)\epsilon_k. \tag{7}$$

where ϕ and θ are polynomials of order p and q respectively, and ϵ is a white noise process, i.e., the ϵ_k are i.i.d. standard normal random variables. The lag operator L is dened as $LX_k = X_{k-1}$. A generalization of this model is the ARIMA (p, d, q) process for $d = 0, 1, ...$, dened by the property that $(1-L)^d X_k$ is an ARMA (p, q) process. As implied by its name, the fractional ARIMA model admits a fractional value for the parameter d. For this, we have to understand how $(1-L)^d X_k$ dened for fractional d. The fractional order is computed using the binomial expansion in (8);

$$(1-L)^d X_k = \sum_{k=0}^{\infty} \binom{d}{k} (-L)^d X_k \tag{8}$$

where the binomial coefficient $\binom{d}{k}$ is defined as;

$$\binom{d}{k} = \frac{\Gamma(d+1)}{\Gamma(d-k+1)\Gamma(k+1)} \tag{9}$$

Since the case $d > 1/2$ can be reduced to the case $-1/2 < d \leq 1/2$ by taking appropriate differences, the latter case is particularly interesting. X is a stationary process for $-1/2 < d < 1/2$. Long-range dependence occurs for $0 < d < 1/2$, implying that the process is also asymptotically second-order self-similar in this case. The corresponding Hurst parameter is $H = 1/2 + d$, [5].

Apart from the standard processes presented above, there are some other interesting hybrid approaches for simulating self-similar processes that are based on either the FBM or FGN. Some of these methods are presented in the subsections that follow.

2.5. Paxson method

Paxson [15] proposed a rather intuitive method for simulating the Fractional Gaussian Noise process. Paxson studied the output of the FGN process by statistically testing if the resulting samples satisfy the desired properties. The approach was found to be justification deficient as it is unclear why the obtained sample should be close to Gaussian. In the Paxson method, the approximate FGN sample is the Fourier transform of

$$b_k = \begin{cases} 0, & k = 0 \\ \sqrt{\frac{R_k f(t_k)}{N}} \exp(i\Phi_k), & k = 1, 2, 3, \ldots, N/2 \\ b^*_{N-k}, & k = \frac{N}{2} + 1, \ldots, N-1 \end{cases} \tag{10}$$

where $R_1, R_2, R_3, \ldots, R_k$ are independent exponentially distributed random variables with mean 1 for $k \geq 1$, and the $(*)$ denotes the complex conjugate. Besides, $\Phi_{N/2}$ is set to zero while $\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_k$ are independent uniformly distributed random variables on $[0, 2\pi]$ for $k \geq 1$, also independent of the R_k. In this case, t_k equals $2\pi k/N$.

2.6. Hosking Method

The Hosking method [9] (also known as the Durbin or Levinson method) is an algorithm to simulate a general stationary Gaussian process. Therefore, the focus here is on the simulation of Fractional Gaussian Noise $X_0, X_1, \ldots$.The method generates $X_(n+1)$ given $X_n, \ldots, X_0$ recursively. It does not use specific properties of Fractional Brownian Motion nor Fractional Gaussian Noise and thus applicable to any stationary Gaussian process. The key advantage of the approach is that the distribution of $X_(n+1)$ given the past can be explicitly computed.

2.7. Cholesky method

The Cholesky method [5] was developed using the Cholesky decomposition of the covariance matrix of Gaussian processes. The approach involves redefining the covariance matrix $\Gamma(n)$ such that it can be written as $L(n)L(n)'$, where $L(n)$ is an $(n+1) \times (n+1)$ lower triangular matrix. Denoting element (i, j) of $L(n)$ by l_{ij} for $i, j = 0, 1, \ldots n$, then $L(n)$ is said to be the lower triangular matrix if $l_{ij} = 0$ for $j > i$. It can be proven that such a decomposition exists when $\Gamma(n)$ is a symmetric positive denite matrix.Unlike the Hosking method, the Cholesky method can be applied to non-stationary Gaussian processes.

The various self-similar generators discussed so far can be evaluated and assessed for their efficiencies by estimating the value of the Hurst index H initially imposed at the simulation step. The rescale range (R/S) statistics is one of the popular methods for estimating the H value of a self-similar process. The following subsection presents the procedure.

2.8. R/S analysis

The R/S statistics discussed by Rose [24] is particularly attractive because of its relative robustness against changes in the marginal distribution, even for long-tailed or skew distributions. Given an empirical time series of length $(X_k : k = 1, ..., N)$, the whole series is subdivided into K non-overlapping blocks. The next is to compute the rescaled adjusted range $R(t_i, d)/S(t_i, d)$ for several values d, where $t_i = [N/K](i-1) + 1$ are the starting points of the blocks which satisfy $(t_i - 1) + d \leq N$. Here, $R(t_i, d)$ is defined as:

$$R(t_i, d) = \max\{0, W(t_i, 1), \ldots, W(t_i, d)\} - \min\{0, W(t_i, 1), \ldots, W(t_i, d)\} \tag{11}$$

where

$$W(t_i,k)=\sum_{j=1}^{k}X_{t_i+j-1}-k\left(\frac{1}{d}\sum_{j=1}^{d}X_{t_i+j-1}\right),k=1,\ldots,d. \tag{12}$$

Also, $S(t_i,d)$ denotes the sample standard deviation of $X_{t_i},\ldots,X_{t_i+d-1}$. For each value of d one obtains a number of R/S samples. For small values of d, there are K samples. The number decreases for larger values of d because of the limiting condition on the t_i values. These samples are computed for logarithmically spaced values of d, i.e. $d_{l+1}=md_l$ with $m>1$ starting with d_0 fixed at 10. The R/S plot also known as pox diagram [24] is computed using these values. The diagram is obtained by plotting $\log[R(t_i,d)/S(t_i,d)]$ against $\log d$.

The next step involves fitting a least-squares line to the points of the R/S plot, where the R/S samples of the external values of d are not considered. The slope of the regression line for these R/S samples is an estimate for the Hurst parameter H. Both the number of blocks K and the number of values d should not be chosen to be too small. In addition, some care has to be taken when deciding about the end of the transient, i.e., which of the small values of d should not be taken into consideration for the regression line.

3. Proposed Generalized Self-Similar First Order Autoregressive Generator (GSFO-ARG)

3.1. The Development of GSFO-ARG Generator

Theorem 3.1*: For any random $AR(1)$ memoryless process X_k, with an underlying distribution D and autocorrelation function $\rho_k=\gamma_k/\gamma_0$. The transformed process X'_k defined over X_k is a self-similar process with the same distribution D and autocorrelation function $\rho'_k=(ck_0^{2H-2})^{\frac{k}{k_0}}$, where k_0 is an optimal fractional index parameter, H is the Hurst index and $c=\exp(k_0\log H-(2H-2)\log k_0)$.*

Proof:

As earlier defined, an ARMA (p,q) process $X=X_k:k=0,1,\ldots$ is a short memory process that is the solution of the equation (13) [2].

$$\phi(L)X_k=\theta(L)\epsilon_k. \tag{13}$$

with ϕ, θ, and lag operator L are as defined in subsection 2.4. The equation (13) can then be re-written as;

$$X_k=\phi_1X_{k-1}+\phi_2X_{k-2}+\cdots+\phi_pX_{k-p}+\epsilon_k+\theta_1\epsilon_{k-1}+\theta_2\epsilon_{k-2}+\ldots\theta_q\epsilon_{k-q}. \tag{14}$$

Accordingly, following Cryer and Chan [3], an AR (1) process is;

$$X_k=\phi X_{k-1}+\epsilon_k. \tag{15}$$

The corresponding autocorrelation property of the process is;

$$\rho_k=\frac{\gamma_k}{\gamma_0} \tag{16}$$

and $var(X_k)=\gamma_0$, which then implies from (16) that;

$$\gamma_0=\phi^2\gamma_0+\sigma_e^2. \tag{17}$$

$\rightarrow$

$$\gamma_0=\frac{\sigma_e^2}{1-\phi^2}$$

and

$$\gamma_k=\phi^k\frac{\sigma_e^2}{1-\phi^2}.$$

Thus,

$$\rho_k = \frac{\phi^k \frac{\sigma_\epsilon^2}{1-\phi^2}}{\frac{\sigma_\epsilon^2}{1-\phi^2}}$$

$\rightarrow$

$$\rho_k = \phi^k. \tag{18}$$

If $|\phi| < 1$, the magnitude of the autocorrelation function decreases exponentially as the number of lags k increases. This implies an $AR(1)$ process is a short memory process. Our approach is to make the simple $AR(1)$ process a long memory process by equating the established autocorrelation of any self-similar process to that of the $AR(1)$ process and solving the resultant expression. The derivation follows as;

$$\rho_{k-Selfsimilar} = ck^{2H-2} \tag{19}$$

$$\rho_{k-Selfsimilar} = \rho_{k-AR(1)} \tag{20}$$

$$ck^{2H-2} = \phi^k \tag{21}$$

$\rightarrow$

$$\sqrt[k]{ck^{2H-2}} = \phi.$$

The problem here is to obtain the value of ϕ that will ensure a specific self-similar index H given c and k. To obtain c that will ensure a specific H implies that $\phi = H$;

$$\sqrt[k]{ck^{2H-2}} = H. \tag{22}$$

$\rightarrow$

$$\log c + (2H-2)\log k = k \log H$$

$$\log c = k \log H - (2H-2)\log k.$$

Thus,

$$c = \exp(k \log H - (2H-2)\log k). \tag{23}$$

After the substitution of (23) in (21), we have,

$$\sqrt[k]{\exp(k \log H - (2H-2)\log k)k^{2H-2}} = \phi. \tag{24}$$

And finally, we have,

$$\phi = (\exp(k \log H - (2H-2)\log k)k^{2H-2})^{\frac{1}{k}} \tag{25}$$

Equation (25) shows that if k is chosen appropriately the autocorrelation will decay slowly since the autocorrelation of $AR(1)$ is ϕ^k.

The next step in the simulation process is obtaining the value of k that will also ensure a specific H keeping in mind that $0 < \phi < 1$ is required to achieve a stationary $AR(1)$ as well as positive autocorrelation property of a self-similar sequence. For $k \leq H$ and $\phi > 1$, the $AR(1)$ process will not be stationary. If $k \to \infty$, then $\phi \to 1$ and thus k lies in the interval $H < k < \infty$. Therefore, to obtain an exact self-similar process with a specific H, k should be increased gradually until the target H is achieved. By direct substitution of (25) in (15), our proposed GSFO-ARG generator X'_k can be obtained using:

$$X'_k = (\exp(k_0 \log H - (2H-2)\log k_0)k_0^{2H-2})^{\frac{1}{k_0}} X_{k-1} + \epsilon_k. \tag{26}$$

where k_0 is the optimal k that will ensure a specific H.The distribution of ϵ_k determines the distribution of X'_k, thus GSFO-ARG generator applies to any distribution. The corresponding autocorrelation property is;

$$\rho'_k = (\exp(k_0 \log H - (2H-2)\log k_0)k_0^{2H-2})^{\frac{k}{k_0}} \tag{27}$$

Equation (27) shows that the autocorrelation of the sequence X'_k will decay hyperbolically and eventually yield the expected autocorrelation of any self-similar process when $k_0 \to k$.

4. Numerical Examples and Results

We present the results of some five existing self-similar generators compared with the proposed GSFO-ARG method at different values of the Hurst parameter H. Since the values of H in the interval $0.5 < H < 1$ are desirable for the self-similar process with LRD, which is the focus of this study. Therefore, we choose $H = 0.6, 0.7, 0.8$ and 0.9 values for the implementation of all the self-similar generators discussed in this study. Also, the optimal value of k_0 that is desirable for each of the chosen Hurst parameters H were determined and used for the proposed GSFO-ARG method to generate its own set of stochastic sequences.

Furthermore, the relative performance of each generator is assessed using the percentage error formula given by;

$$E = \frac{H - \hat{H}}{H} \times 100 \tag{28}$$

In some instances, especially in cases for which the value of the Hurst parameter H is underestimated by a generator, the absolute percentage error can be used and it is given by;

$$|E| = \frac{|H - \hat{H}|}{H} \times 100 \tag{29}$$

In equations (28) and (29), $\hat{H}$ represents the estimate of the Hurst parameter H provided by each generator.

Results in Table 1 are the average estimates of the Hurst parameters based on the sequences generated by each of the existing self-similar generators compared with our proposed generator at sample size $n = 2^8$ averaged over 1000 iterations. Table 1 also reports the percentage error (PE) and absolute percentage errors (APE) of estimates as computed by all the six self-similar generators. The APE is used to assess the efficiency of the generators of the self-similar process. Based on this assessment, a method with the least APE is the most efficient among all the self-similar generators considered. Similarly, the results of all the six self-similar generators at sample size $n = 2^10$ over 1000 iterations are also provided in Table 2. A similar pattern of results were equally observed in both the Table 1 and Table 2 for all the six generators. Tables 1 and 2 also show that the proposed GSFO-ARG method has the least $\%$ absolute errors and thus adjudged as the most efficient of all the six methods considered. To have a clear overview and assessment of the level of closeness of the estimated Hurst parameters $\hat{H}$ as provided by the six self-similar generators including the proposed GSFO-ARG method to their true values H, we plotted H against $\hat{H}$ and it is shown in Figures 1a and 1b using the results in Tables 1 and 2 for $n = 2^8$ and $n = 2^{10}$ respectively. The line plot of the true value H against the expected value of $\hat{H}$ is equally provided in solid black lines, as shown in Figures 1a and 1b. It should be noted that $E(\hat{H}) = H$, thus for $H = (0.6, 0.7, 0.8, 0.9)$, $E(\hat{H}) = (0.6, 0.7, 0.8, 0.9)$. Therefore, the plot of H against $E(\hat{H}) = H$ should produce a line that makes an angle $\angle 45^0$ with the x-axis (solid black lines in Figures 1a and 1b) which is the reference line against which the line provided by each of the generators is compared. By this, a generator whose line is the closest to the reference line would be adjudged the best of the six generators.

Finally, we provide in Figure 2, the line graphs of the percentage absolute errors of all the six self-similar generators at the chosen Hurst parameter values as reported in Table 2 at sample size $n = 2^{10}$. The method that provided a graph with points closest to the horizontal x-axis of the graph is adjudged to be the most efficient among the competing ones.

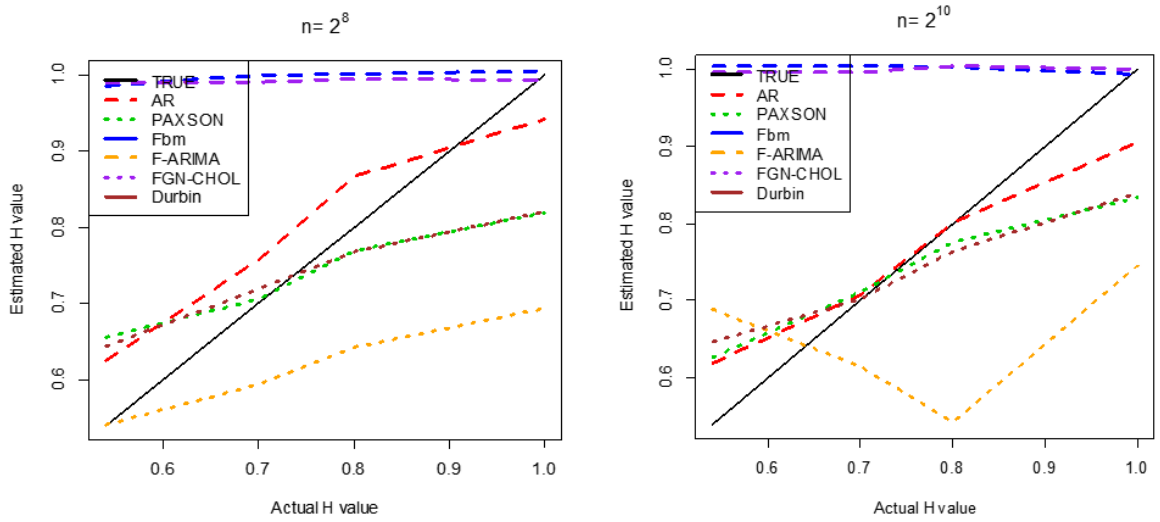

Figure 1. Figure 1a. shows the plots of the Hurst parameters (H) obtained by the five chosen self-similar generators and the proposed GSFO-ARG (AR) generator against the true values H at sample size $n = 2^8$.The plot of H against $E(\hat{H}) = H$ for values of H in the interval $0.5 < H \leq 1$ is also provided in a solid black line. Figure 1b. shows the plots of the Hurst parameters (H) provided by the five chosen self-similar generators and the proposed GSFO-ARG (AR) generator against the true values H at sample size $n = 2^{10}$. The plot of H against $E(\hat{H}) = H$ for values of H in the interval $0.5 < H \leq 1$ is also provided in solid black line.

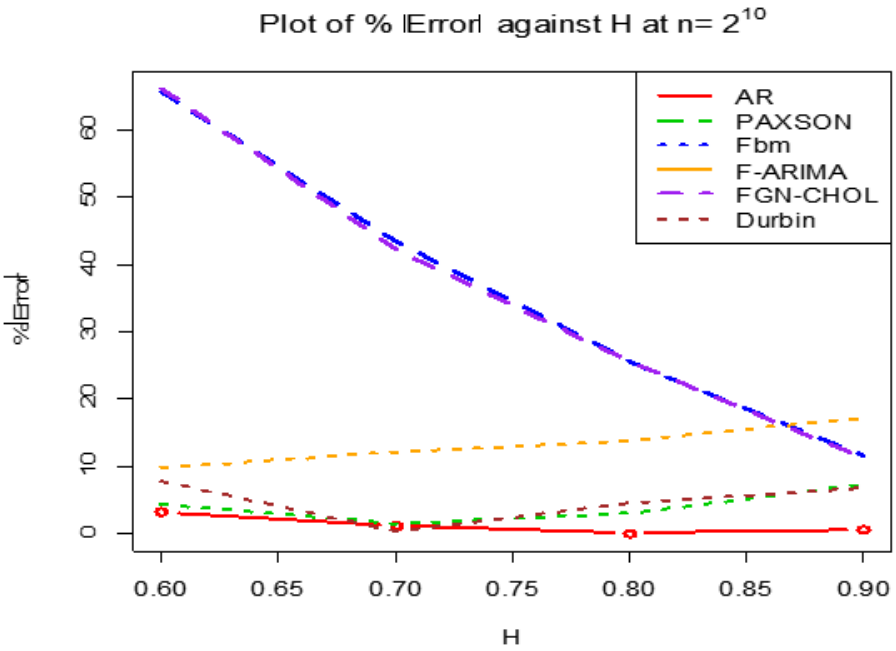

Figure 2. The plots of the percentage absolute error of estimates of the Hurst parameters (H) of all the six self-similar generators against the chosen values of H at sample size $n = 2^{10}$

Table 1. Results of generators performances at $n = 2^8$ over 1000 iterations.

Hurst (H)	Generator	Mean $(\hat{H})$	% Error (E)	% Absolute error $(\lvert E\rvert)$
0.6	GSFO-ARG; $k_0 = 1.99$	0.6254	4.2271	4.2271
	PAXSON	0.6551	9.1771	9.1771
	FBM	0.9852	64.1951	64.1951
	F-ARIMA	0.5404	-9.9256	9.9256
	FGN-CHOLESKY	0.9894	64.9039	64.9039
	DURBIN	0.6443	7.3895	7.3895
0.7	GSFO-ARG; $k_0 = 1.11$	0.7577	8.2394	8.2394
	PAXSON	0.7059	0.8442	0.8442
	FBM	1.0003	42.8984	42.8984
	F-ARIMA	0.5934	-15.2248	-15.2248
	FGN-CHOLESKY	0.9910	41.5773	41.5773
	DURBIN	0.7197	2.8090	2.8090
0.8	GSFO-ARG; $k_0 = 0.979$	0.8674	8.4273	8.4273
	PAXSON	0.7687	-3.9064	3.9064
	FBM	1.0016	25.2036	25.2036
	F-ARIMA	0.6421	-19.7410	19.7410
	FGN-CHOLESKY	0.9943	24.2854	24.2854
	DURBIN	0.7683	-3.9625	3.9625
0.9	GSFO-ARG; $k_0 = 0.967$	0.9415	4.6139	4.6139
	PAXSON	0.8198	-8.9118	8.9118
	FBM	1.0052	11.6858	11.6858
	F-ARIMA	0.6945	-22.8322	22.8322
	FGN-CHOLESKY	0.9937	10.4148	10.4148
	DURBIN	0.8208	-8.8014	8.8014

5. Discussion of results

Tables 1 and 2 present the estimated average values of the Hurst parameter index H as provided by each of the six self-similar generators including the proposed one at sample sizes $n = 2^8(256)$ and $n = 2^{10}(1,024)$ respectively. Also, the percentage errors and absolute percentage errors of these estimates were reported for each method using the R/S statistics at these two sample sizes. It was observed that at various Hurst index H levels, the proposed GSFO-ARG(1) generator produces a fairly close estimate compared to the other generators. Among the five existing generators considered, Paxson and Durbin's method produced better estimates compared to the other three existing generators (FBM, FGN-Chol, and F-ARIMA) considered.

It can be observed from the various results in Tables 1 and 2 that the F-ARIMA method under-estimated while FBM and FGN-Chol overestimated the imposed H values at all the chosen levels. The results of R/S statistics on all the six methods showed that the proposed GSFO-ARG method yielded the least percentage absolute error (PAE), at all the chosen levels of the Hurst index H.

In summary, the efficiency of five existing self-similar Internet packets generators was examined. Furthermore, an alternative method, GSFO-ARG(1) for generating such a self-similar sequence, was proposed for efficiency gain. The efficiency of the five existing methods and the proposed method was assessed using PAE. The various results presented in Tables 1 and 2 showed that the method that provided the best estimates of the Hurst index

Table 2. Results of generators performances at $n = 2^{10}$ over 1000 iterations.

Hurst (H)	Generator	Mean $(\hat{H})$	% Error (E)	% Absolute error $(\|E\|)$
0.6	GSFO-ARG; $k_0 = 1.99$	0.6190	3.1694	3.1694
	PAXSON	0.6266	4.4307	4.4307
	FBM	0.9940	65.6740	65.6740
	F-ARIMA	0.5416	-9.7321	9.7321
	FGN-CHOLESKY	0.9959	65.9875	65.9875
	DURBIN	0.6460	7.6651	7.6651
0.7	GSFO-ARG; $k_0 = 1.11$	0.7078	1.1143	1.1143
	PAXSON	0.7108	1.5387	1.5387
	FBM	1.0033	43.3237	43.3237
	F-ARIMA	0.6148	-12.1760	12.1760
	FGN-CHOLESKY	0.9958	42.2589	42.2589
	DURBIN	0.7021	0.2987	0.2987
0.8	GSFO-ARG; $k_0 = 0.979$	0.8004	0.0552	0.0552
	PAXSON	0.7759	-3.0085	3.0085
	FBM	1.0046	25.5778	25.5778
	F-ARIMA	0.6892	-13.8563	13.8563
	FGN-CHOLESKY	1.0038	25.4808	25.4808
	DURBIN	0.7644	-4.4543	4.4543
0.9	GSFO-ARG; $k_0 = 0.967$	0.9054	0.5971	0.5971
	PAXSON	0.8343	-7.2995	7.2995
	FBM	1.0039	11.5428	11.5428
	F-ARIMA	0.7448	-17.2419	17.2419
	FGN-CHOLESKY	0.9996	11.0714	11.0714
	DURBIN	0.8391	-6.7662	6.7662

H is the proposed GSFO-ARG method, which was simply shortened as AR(1) generator because its development originated from the first-order autoregressive process.

6. Conclusion

It can generally be concluded that whenever the interest centres on examining the self-similar processes of Internet traffic, the proposed GSFO-ARG(1) method should be employed. This is irrespective of whether the self-similar process is Gaussian or non-Gaussian.

In the future study, it might be imperative to consider computer systems for the impacts of different operating systems and other properties often employed by users systems for generating the sequences of Internet traffic data. This is desirable to determine the influence of different systems configurations on the properties of the self-similar processes as established here and elsewhere [16, 17, 18].

REFERENCES

1. X. Bai and A. Shami, *Modelling self-similar traffic for networks simulation*, arXiv preprint arXiv:1308.3842, 2013.
2. G. Box, G. Jenkins, G. C. Reinsel and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, San Francisco: John Wiley & Sons, 2015.
3. D. Cryer and K. Chan, *Time Series Analysis with Application in R.*, New York USA.: Springer Science + Business Media LLC, 2008.
4. C. K. Dehury and P. K. Sahoo, *DYVINE: fitness-based dynamic virtual network embedding in cloud computing*, IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, pp. 1029–1045, 2019.
5. T. Dieker, *Simulation of fractional Brownian motion*, Enschede: University of Twente, 2004.
6. M. Fras, J. Mohorko and Ž. Čučej, *Limitations of a Mapping Algorithm with Fragmentation Mimics (MAFM) when modeling statistical data sources based on measured packet network traffic*, Computer Networks, Elsevier, vol. 57, no. 17, pp. 3686–3700, 2013.
7. S. Haer and L. I. Trajkovic, *Virtual network embedding via Monte-Carlo tree search*, IEEE Transactions on Cybernetics, vol. 48, no. 2, pp. 510–521, 2018.
8. H. E. Hurst, *Long-term storage capacity of reservoirs*, Transactions of the American Society of Civil Engineers, vol. 116, pp. 770799., 1951.
9. P. Inácio, B. Lakic, M. Freire, M. Pereira, and P. Monteiro, *The design and evaluation of the simple self-similar sequences generator*, Information Sciences, Elsevier, vol. 179, no. 23, pp. 4029–4045, 2009.
10. A. Jahani, L. M. Khanli, M. T. Hagh and M. A. Badamchizadeh, *EE-CTA: Energy efficient, concurrent and topology-aware virtual network embedding as a multi-objective optimization problem*, Computer Standards & Interfaces, 66, 2019.
11. S. A. M. Jamil, M. A. A. Abdullah, S. L. Kek, O. R. Olaniran and S. E. Amran, *Simulation of parametric model towards the fixed covariate of right censored lung cancer data*, Journal of Physics: Conference Series, IOP Publishing, vol. 890, no. 1, pp. 012172, 2017.
12. B. B. Mandelbrot, *A multifractal walk down Wall Street. Scientific American*, Scientific American, vol. 280, no. 2, pp. 70–73, 1999.
13. I. Norros, *On the use of fractional Brownian motion in the theory of connectionless networks*, IEEE Journal on selected areas in communications, vol. 13, no. 6, pp. 953–962, 2006.
14. K. Park and W. Willinger, *Sef-similar Traffic and Performace Evaluation*, NY: John Wiley & Sons., 2000.
15. V. Paxson, *Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic*, ACM SIGCOMM Computer Communication Review, vol. 27, no. 5, pp. 5–18, 1997.
16. J. Popoola and R. A. Ipinyomi, *Empirical Performance of Weibull Self-Similar Tele-traffic Model*, International Journal of Engineering and Applied Sciences (IJEAS), vol. 4, no. 8, pp. 77–79, 2017.
17. J. Popoola, W. B. Yahya, R. A. Ipinyomi and O. R. Olaniran, *On The Distribution Of Transmission And Arrival Times Of Packet-Switched Network Under Self-Similarity And Long Range Dependency*, Asian Journal of Mathematics and Computer Research, vol. 16, no. 4, pp. 185–196, 2017.
18. J. Popoola, O. Popoola and O. R. Olaniran, *An Approximate Performance of Self-Similar Lognormal M/1/K Internet Traffic Model*, Journal of Science and Technology, vol. 11, no. 2, pp. 36–42, 2019.
19. O. R. Olaniran and M. A. A. Abdullah, *Bayesian Variable Selection for Multiclass Classification using Bootstrap Prior Technique*, Austrian Journal of Statistics, vol. 48, no. 2, pp. 63–72, 2019.
20. O. R. Olaniran and W. B. Yahya, *Bayesian hypothesis testing of two normal samples using bootstrap prior technique*, Journal of Modern Applied Statistical Methods, vol. 16, no. 2, pp. 618–638, 2017.
21. O. R. Olaniran and M. A. A. Abdullah, *Subset Selection in High-Dimensional Genomic Data using Hybrid Variational Bayes and Bootstrap priors*, Journal of Physics: Conference Series, IOP Publishing, vol. 1489, pp. 012030, 2020.
22. S. Rezakhah, A. Phillipe and N. Modarresi, *Estimation of Scale and Hurst Parameters of Semi-Selfsimilar Processes*, arXiv preprint, arXiv:1207.2450.
23. A. Rios, Z. Li, G. Xu, A. Alonso and L. Trajkovic, *Detecting network anomalies and intrusions in communication networks*, Proc. 23rd IEEE International Conference on Intelligent, pp. 29–34, 2019.
    Rose, O. (1996). Estimation of the hurst parameter of long-range dependent time series. Wurzburg: Institute of Computer Science Research Report Series 137, University of Wurzburg.
24. O. Rose, *Estimation of the hurst parameter of long-range dependent time series*, Wurzburg: Institute of Computer Science Research Report Series 137, University of Wurzburg, 1996.
25. O. Sheluhin, S. Smolskiy and A. Osin, *Self-similar processes in telecommunications*, Hoboken, New Jersey: John Wiley & Sons, 2007.
26. H. Yılmaz, *IP over DVB: management of self-similarity*, Boazii University: Doctoral dissertation, 2002.