

Virtual networks prediction by using the super multiplicative DEA model and fractals

Francisco Daladier Marques Júnior¹

IFPB/Campus Cajazeiras - UNINFO
Rua José Antônio da Silva, 300, Jardim Oásis.
Cajazeiras-Paraíba
e-mail: daladierjr@ifpb.edu.br

Ali Emrouznejad²

Surrey University - Surrey Business School
GU2 7XH
Guildford, Surrey, UK
e-mail: a.emrouznejad@surrey.ac.uk

Jorge Luiz de Castro e Silva³

UECE-LARCES
Av. Dr. Silas Munguba, 1700
Fortaleza-Ceará
e-mail: jlcs@larces.uece.br

Paulo Roberto Freire Cunha⁴

UFPE – Centro de Informática
Av. Jornalista Anibal Fernandes SN, Cidade Universitária
Recife-Pernambuco
E-mail: prfc@cin.ufpe.br

ABSTRACT

Virtual Networks (VN) have been used to support the network traffic in data centres for the delivery of all kinds of services in cloud computing. Here, we developed a super-efficiency multiplicative data envelopment analysis model (SMDEA) for VN service's forecasting based on real measurements. Another contribution of this essay is to show that the self-similarity (SS) with Long-Range Dependence (LRD) has a different performance per network/setting/device that were analysed as decision-making units (DMU) by the multiplicative DEA models. This paper also employs fractal analysis on computer networks to predict traffic trends using a one-time series evaluation per DMU. Then, the multiplicative DEA models give the decision-maker the capacity to pick a setting with smoother traffic and the better TCP transfer rate over time. Finally, the results demonstrate the superiority of SMDEA versus classic super-efficiency DEA models, also providing future research directions.

KEYWORDS. Multiplicative Data Envelopment Analysis models. Fractal Analysis. Roughness/Smoothness. Virtual Networks.

SE - DEA- Análise Envolvória de Dados

1. Introduction

Virtualization is an abstraction software concept, which means to mimic a similar behaviour of any physical resource, such as CPU, memory, storage, network links, etc. According to [Chowdhury and Boutaba 2010], network virtualization is the core component of network infrastructure in data-centres, for cloud computing providers (CSP) to connect virtual machines (VM), deploy traffic engineering policies, and so on. Moreover, network virtualization can share and slice bandwidth in traditional and software-defined networks (SDN) backbones of infrastructure providers [Bellavista et al. 2015].

Traditional network traffic behaviour has been widely analysed – over thirty years – based on the findings of the Fractals and Self-Similarity (SS) in diverse environments such as the seminal works on telecommunications [Mandelbrot 1965], ethernet [Leland et al. 1994], web traffic [Crovello and Bestavros 1997], and many more. All those studies proved that their traffic is statistically SS with a long-range of dependence (LRD). SS with LRD traffic according to these prior works had dangerous implications for the design, control, and analysis of high-speed networks.

Furthermore, most of works that showed the SS with LRD in computer networks only observed these stochastic patterns within a unique device/setting/network. To the best of our knowledge until 2019 all articles on SS only investigated it in a single network, generating a large temporal series where the fractal indices were computed, exploratory data analysis performed, and conclusions drawn about the data acquired. However, the work of [Marques Júnior et al. 2019] has devised a new approach, introducing a fractal evaluation framework capable of predicting/comparing various virtual networks using fractal decision variables, to observe among the set of virtual networks under analysis, which one had the optimal linear combination to provide the best services to the clients over time. This work launched a predictor, through the creation of a multi-criteria decision-making (MCDM) technique so-called windows multiplicative data envelopment analysis (WMDEA) a way of ranking virtual networks in comparison, according to the same decision variables in a dynamic manner. Therefore, in this work, only one time series is evaluated per DMU, i.e., a static DEA analysis using fractal time series per decision-making unit (DMU) to pinpoint which one had the best fractal pattern for providing transmission control protocol (TCP) services over time.

The MCDM method chosen by this research is the data envelopment analysis (DEA) created by [Charnes et al. 1978]. DEA has gained extensive attention over its 46 years of existence, the technique is being applied with success to guarantee the comparison of DMUs using the same criteria or/variables to point out which DMUs are efficient or not in several problems. DEA has been applied on performance evaluation, mainly in managerial, education, supply chain, energy, health, banks, industry, and so on [Emrouznejad and Yang 2017]. Due to its nature of optimisation of non-parametric variables of input and output, DEA seeks to minimise the input or maximise the outputs, or both simultaneously, depending on the model and orientation used to solve each problem. However, by evaluating the DMU, the majority of DEA models generated as a result if each DMU is/not efficient, forgetting to point out which DMU is better than other techniques for ranking.

Multiplicative DEA models were proposed by [Seiford et al. 1982] to treat the scale elasticity using ratio variables, i.e. they were designed to work with problems which exhibit a frontier of efficiency with parts concave and non-concave simultaneously [Banker 2004].

Super-efficiency DEA (SDEA) model was initially proposed by [Andersen and Petersen 1993] which presented it as a procedure for ranking DMUs or identifying outliers. Accordingly, we are proposing a multiplicative DEA model for ranking only one-time series data per DMU. It should be noted that some of our variables are in the form of ratios, hence the standard DEA models do not produce a correct evaluation due to the convexity, proportionality [Emrouznejad and Cabanda 2010], and monotonicity [Banker and Maindiratta 1986] axioms.

Shortly, the contributions of this paper are: **a)** propose super multiplicative DEA (SMDEA) models, where the decision-maker will have the possibility of choosing one or more sets of tools at the same time that can offer smooth traffic with maximised TCP transfer rate for an extended

time with memory, guaranteeing optimised services for Internet subscribers; **b)** to present a public dataset with the results obtained with the SMDEA proposed in comparison with another SDEA classic model, and; **c)** to indicate directions on what can be done to provide optimal services on the Internet, based on fractals.

The rest of this paper is organised as follows. Section 2 presents the background of related works, DEA, and the multiplicative DEA models launched. In section 3, it is detailed the performance evaluation of VNs. In Section 4, the results are presented and discussed. Finally, Section 5 presents the conclusions and the direction for future works.

2. Background

This section presents the SS, the fractal dimension, and the formalisms that will be used later in a detailed manner. Finally, this section brings the multiplicative DEA models devised.

2.1 Self-Similarity & Fractals

Over the last three decades, it has been observed that computer network traffic is strongly positive and auto-correlated with LRD being modelled as a non-stationary self-similar stochastic process [Leland et al. 1994], [Crovella and Bestavros 1997]. SS is related to a property of an object, or stochastic process which seems to have the same behaviour at scaling over time and space. It is important to understand modelling a time series and guiding predictions regarding extreme events or future trends behaviour using fractal analysis, i.e. the fractal analysis serves to represent a system in a dimensionally small representation [Kantelhardt 2008].

The Fractal or SS theory was devised by [Mandelbrot 1965] and observed that memoryless channels are unable to account for the behaviour of telecom circuits of high quality or low rate of errors. Although these errors are occasionally encountered, many of them appear to be always grouped, which keeps groups in bursts, being grouped successively over and over time. The SS has been employed in many areas such as the stock markets, hydrology, medicine, DNA sequences, computer networks and security, meteorology, geology, and many others. Most specifically, in computer networks, an interesting survey related to the state-of-art of the SS with LRD must be found in [Liu et al. 2012].

There are many ways to measure the SS degree of a time series such as rescaled range analysis (R/S), Hill's estimator, detrended fluctuation analysis, and so on [Weron 2002]. Moreover, graphical tools also constitute themselves as alternatives to make a fractal analysis such as periodogram, empirical cumulative distribution function (ECDF) plots, autocorrelation function (ACF) plots, histograms, etc. The graphs of the time series data are characterised by a probability density/distribution function (PDF) with a heavy-tail pattern that decreases more slowly than the exponential ones [Paxson and Floyd 1995].

We chose the R/S method due to its simplicity, it is also called the Hurst parameter (H) [Hurst 1951]. Furthermore, the R/S statistic is calculated to obtain the sample standard deviation for completing the H 's computation.

According to H 's value, there are three kinds of SS stochastic process, which are: a) **Short-Range Dependence (SRD)** – when $0 < H < 0.5$, also called an anti-persistent process or memoryless. Exhibits of Noah's effect due to these events are unlikely to happen; b) **Long-Range Dependence (LRD)** – when $0.5 > H < 1$, also named a persistent process. Exhibits of Joseph's effect because of these events are reverberated or have memory over time, where generally short numbers follow short numbers and long numbers follow long numbers, and; c) **Fractal-Brownian Motion (FbM) or Random walk** – when $H = 0.5$, where the increments are entirely independent of each other, generally caused by the source.

We are only interested in the persistent ones to make an accurate prognosis of the data trend. According to [de Lima and de Almeida Amazonas 2013], when a LRD's time series has H closer (tending) to 1, then its memory (persistence degree) is higher. Thus, H is used as the output variable on VN optimisation using our devised multiplicative DEA formulations. Another important concept related to SS with LRD is Fractal or Hausdorff dimension D which serves as a

measure of roughness (smoothness) of a time series containing values on the interval $[1,2)$ [Gneiting et al. 2012].

Fractal dimension is already being employed in financial markets as a measure of the market's volatility to predict market trends. According to [Mandelbrot and Taleb 2012], H is directly related to fractal dimension D which serves to show whether a time series exhibits a "mild" or "wild" randomness. The time series used in this work are considered to be one-dimensional due to the CRV obtained being measured in a homogenous time interval, for this reason, there is a mono-fractal pattern of our data ($D(0) = 1$), and that they were calculated in some manner too. H value is also directly related to fractal dimension, where $D = 2 - H$, thus when the time series exhibits SRD then the autocorrelation of the CRV is negative, and when it exhibits LRD the autocorrelation is positive. However, it is necessary to have a more accurate analysis of fractal dimension, through several methods to obtain fractal dimension including rodogram, Genton, Hall-Wood, madogram, box-count, variogram, spectral density, wavelet-based estimators, etc. [Gneiting et al. 2012]. We have used the first four of the earlier cited methods for obtaining fractal dimension as input variable on multiplicative DEA assessments. However, due to the values of different kinds of fractal dimension methods computed being highly auto-correlated, the rodogram was then chosen as a unique input variable.

Throughout D 's value, one can arrive at two important conclusions: i) When D is close to the value of 1, then the time series present a **smoothing's effect**, i.e. the traffic has an TCP transfer rate that gets more stable over time; ii) When D is close to 2, then the time series present a **roughness effect**, i.e. the traffic is irregular/burstiness with infinite variance and high variability over time

The fractal dimension has huge importance for comparing a dataset with another for ranking [Hall and Roy 1994]. However, the super-multiplicative DEA (SMDEA) model is searching for efficient DMU with lesser values of D . Then, the aim is to seek to maximise H plus a significant TCP transfer rate mean which is the guarantee of a bigger memory and performance over time, with smoother traffic too.

This research also shows that the SS with LRD has distinct performance behaviour when the set of tools to deliver a service is changed. For this reason, if the network administrator had to use the set of tools chosen by SMDEA to deploy a virtual environment and needs to change one or more of them, then the performance as a whole should be different in each time series. In this study, we only create 50 distinct time series to make an SS analysis of each one as a DMU in the DEA model for finding the most efficient settings to deliver VN services using TCP.

2.2 DEA and Super-efficiency DEA models

DEA is a MCDM technique that has been used over the past 46 years to evaluate DMUs according to the concept of relative efficiency of the same units of production's appraisal in almost all areas [Emrouznejad and Yang 2017]. DEA uses linear programming to compute the relative efficiency of each DMU. DEA is based on Pareto-Koopman's efficiency or envelope frontier, i.e. trying to decrease the input without changing the output values (input-oriented), or trying to increase the output without modifying the input values (output-oriented), or simultaneously minimising the input and maximising the output in non-oriented model.

Firstly, for the characterization of every DEA model, it is mandatory to know the concept of the Production Possibility Set (PPS). PPS is obtained through the estimation of the distance of each unit to the border, which will define if a DMU is/not efficient. For this reason, consider each DMU/unit uses m inputs to produce s outputs. Let a set of n DMUs be evaluated as DMU_j (for each $j = 1, \dots, n$) linked to the input vector of $X_j = (x_{1j}, \dots, x_{mj})$ and the output vector of $Y_j = (y_{1j}, \dots, y_{sj})$, then the PPS can be described as:

$$P = \{(X, Y): X \text{ can produce } Y\}$$

The linear formulation of the CCR output-oriented model is presented in equation (1). The slacks on DEA models may show if a DMU is operating with excess of input which should be decreased (s_i^-) to reach the efficiency frontier, or a shortage of production variables, where these variables should be increased (s_r^+) to seek the efficiency frontier. Hence, the slacks in $(m + s)$ constraints of equation (1) are represented by coefficients $s_i^- \geq 0$ and $s_r^+ \geq 0$ in formulation for inputs and outputs respectively, then:

$$\max \phi_{j_0} \quad (1)$$

Subject to

$$\sum_{j=1}^n \lambda_j * x_{i,j} + s_i^- = x_{i,j_0}, i = 1, \dots, m$$

$$\sum_{j=1}^n \lambda_j * y_{r,j} - s_r^+ - \phi_{j_0} * y_{r,j_0} = 0, r = 1, \dots, s$$

$$\lambda_j \geq 0, j = 1, \dots, n$$

$$\phi \text{ free and } s_i^- \geq 0 \text{ \& } s_r^+ \geq 0$$

The SDEA approach removes the under-evaluation unit called DMU_0 from observations set during the assessments of each DMU. Hence, the SDEA formulation of equation (1) is presented in equation (2).

$$\max \phi_{j_0} \quad (2)$$

Subject to

$$\sum_{j=1 \neq j_0}^n \lambda_j * x_{i,j} + s_i^- = x_{i,j_0}, i = 1, \dots, m$$

$$\sum_{j=1 \neq j_0}^n \lambda_j * y_{r,j} - s_r^+ - \phi_{j_0} * y_{r,j_0} = 0, r = 1, \dots, s$$

$$\lambda_j \geq 0, j = 1, \dots, n$$

$$\phi \text{ free and } s_i^- \geq 0 \text{ \& } s_r^+ \geq 0$$

The difference between DEA and SDEA efficiency scores is that when a DMU is efficient on standard DEA models then its score on SDEA models is higher than the unit, even though inefficient ones keep themselves with the same score in both models.

2.3 Multiplicative DEA models and applications

In some real applications, the decision-maker is not always able to control the input and output variables, e.g. in virtual routers, advertisements of concurrency, number of competitors, median income in service area. These variables which cannot be controlled are considered exogenous or non-controllable, for this reason, it is necessary to adjust standard DEA models to treat these problems [Banker and Morey 1986].

However, there are some issue by using standard DEA models if the input and output variables are in the form of ratios, like in our case (see Table 2). [Banker and Maindiratta 1986] have proposed a piecewise log-linear production function connected with the PPS as follows:

$$P = \{(X, Y) | x_i \geq \prod_{j=1}^n x_{ij}^{\lambda_j} > 0, i = 1, \dots, m, 0 < y_r \leq \prod_{j=1}^n y_{rj}^{\lambda_j}, r = 1, \dots, s, \text{ for all } \lambda_j \geq 0\}$$

Instead, other prior works in DEA propose relaxing the convexity axiom in Multiplicative and other models to contour this restriction as is described in [Emrouznejad and Amin 2009].

2.3.1 Multiplicative DEA models proposed

Let's assume an output-oriented assumption, since the objective in our study is to seek to maintain the input variables constant, hence maximizing the outputs, i.e. this approach targets the

lesser values of fractal dimension and at the same time searches higher values for TCP bandwidth, as well as higher Hurst's parameter values.

These Multiplicative DEA models proposed are a dual CCR output-oriented DEA custom model after a log-linearisation. Hence, based on the multiplicative PPS previously shown, a multiplicative DEA model proposed is formulated as follows:

$$\max \phi_{j_0} \quad (3)$$

Subject to

$$x_{i0} = \prod_{j=1}^n x_{ij}^{\lambda_j}, i = 1, \dots, m$$

$$\phi_{j_0} * y_{r0} = \prod_{j=1}^n y_{rj}^{\lambda_j}, r = 1, \dots, s$$

$$\phi_{j_0}, \lambda_j \geq 0, j = 1, \dots, n$$

For identification of slacks in constraints of equation (4), it is necessary to add these Multiplicative coefficients s_i^- and s_r^+ in formulation for inputs and outputs respectively, then:

$$x_{i0} = s_i^- + \prod_{j=1}^n x_{ij}^{\lambda_j}, i = 1, \dots, m \quad (4)$$

$$\phi_{j_0} * y_{r0} - s_r^+ = \prod_{j=1}^n y_{rj}^{\lambda_j}, r = 1, \dots, s$$

The use of a one-to-one mapping of variables in a logarithm basis transforms the original Multiplicative formulations as the equations (3) and (4) in a log-linear programming model obeying the strict monotonicity. Applying logarithms in (4), the Multiplicative DEA models proposed are presented in log-linear programming formulations in Table 1.

Table 1

Equations from Multiplicative and Super-Multiplicative output-oriented models proposed.

Multiplicative output-oriented model	Super Multiplicative output-oriented model
Objective Function: $\max \phi_{j_0} \quad (5)$	Objective Function: $\max \phi_{j_0} \quad (6)$
Subject to: $\sum_{j=1}^n \lambda_j * \hat{x}_{i,j} + s_i^- = \hat{x}_{i,j_0}, i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j * \hat{y}_{r,j} - s_r^+ - \phi_{j_0} * \hat{y}_{r,j_0} = 0, r = 1, \dots, s$	Subject to: $\sum_{j=1 \neq j_0}^n \lambda_j * \hat{x}_{i,j} + s_i^- = \hat{x}_{i,j_0}, i = 1, \dots, m$ $\sum_{j=1 \neq j_0}^n \lambda_j * \hat{y}_{r,j} - s_r^+ - \phi_{j_0} * \hat{y}_{r,j_0} = 0, r = 1, \dots, s$

$\lambda_j \geq 0, j = 1, \dots, n$ $\emptyset$ free	$\lambda_j \geq 0, j = 1, \dots, n$ $\emptyset$ free
---	---

Due to the use of a maximisation model, the efficiency score is computed through the inverse of its optimum value per DMU. Thus, to obtain the efficiency score of each DMU in (5) and (6) it is still necessary to make this computation:

$$efficiency_score = 1/\phi_{j_0} \quad (7)$$

Where the super-efficiency DMUs have $efficiency_score > 1$. We use the proposed super-efficiency multiplicative DEA model (SMDEA) to choose the most efficient units out of numerous DMUs appraised.

3. Performance Evaluation Methodology

This essay also uses the continuous stepwise fractal performance evaluation framework methodology devised by [Marques Junior et al. 2019] with only one difference, because this work compared the DMUs with just one time series, rather than the cited research that assessed by using some time series per DMU in a window manner.

The measurements were conducted following the methodologies expressed in RFC 2544¹ and its extension the RFC 6815². These documents discuss and define some tests that can be used to measure the performance of network devices using benchmark tools, as well as describe formats for reporting the results. These RFCs depict the figure of a device under test (DUT), which receives connections from a tester device called the traffic generator (TG) through ports. In addition, according to these specifications, it is necessary to cite the versions of all tools in comparison, besides presenting the results of the means and variances of the measurements with a duration of only 60 seconds.

This work carried out measurements with a total time of 600 seconds (10 repetitions) by setting/DMU to get more accuracy and statistical significance of analysis. For this reason, our work also may serve as an update and statistical review relating to SS with LRD of previously cited RFCs.

The benchmark tool used to generate the maximum amount of TCP traffic per settings/DMU was iperf³.

To evaluate the performance of the two container-based strategies the topology **Guest to Guest to Container (GGC)** was created. At GGC, measurements were conducted from the Guest VM OS (DUT/iperf server) through the virtual switch from type II hypervisors (VirtualBox/VMWare) which passes the packets/flows to another Guest VM. Finally, the Guest VM forwards the packets/flows via virtual a bridge to the container tool (traffic generator (TG)), and vice versa. Note that both Guest VMs run within the type II hypervisors (VMWare Player Workstation versus VirtualBox). Such a topology is the same as that used for both Docker and LXC (Figure 1). For each Linux distro, its container is accompanied by the same primary OS, e.g. when Arch was used on Guest VM its container Docker/LXC was also Arch, and so on.

¹ Request for Comments of number (RFC) 2544 – available at URL: <https://www.ietf.org/rfc/rfc2544.txt>

² Request for Comments of number (RFC) 6815 – available at URL: <https://www.ietf.org/rfc/rfc6815.txt>

³ iperf – available at URL: <https://iperf.fr>

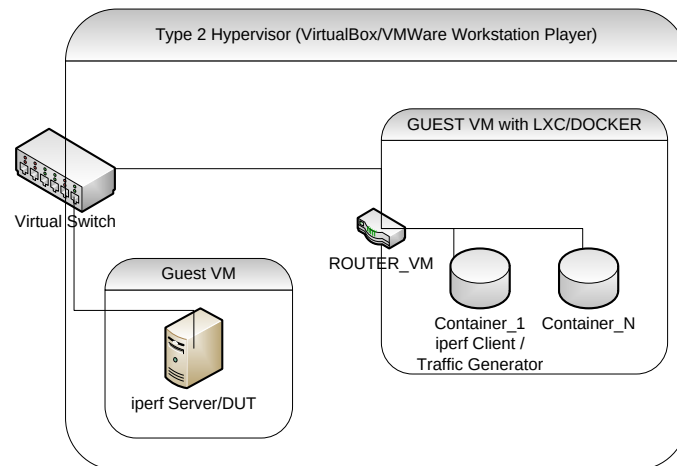


Figure 1. Guest to Guest to Container (GGC) experimentation's topology.

In short, to solve this problem we proposed the fractal dimension as an input and the TCP average of bandwidth and Hurst parameter as output variables in the DEA models. Hence, the optimisation models must target one DMU with a better linear combination that seeks a smooth fractal dimension, huge TCP bandwidth mean, and great fractal memory (H) over time.

4. Results and discussions

This section discusses the results referred to the experiments and DEA models proposed. Table 2 contains the results of measurements using TCP of VN in GGC topology. Note that whichever one we consider as a set of tools to deliver VN services as DMU, the results are ordered alphabetically. Nevertheless, Table 2 must be presented as a whole because it is mandatory for a fractal analysis of an entire set of experiments in every case.

As depicted in Table 2, the DMU names always follow the same format as the Linux distro used, followed by its hypervisor tool plus its network emulated device interface (if it exists), and the container application.

Look in Table 2 that each DMU has an entirely distinct performance over time with different TCP throughput, fractal dimension, and memory (H) during the time. Note that by changing only one set of the tools, both performance and memory of the selected DMU may change over time. For instance, as observed in Table 2 the setting with the biggest TCP transfer rate, (i.e. DMU19), has 463.313% higher TCP throughput than the smaller TCP bandwidth setting (i.e. DMU21). Also, observe that this performance pattern has a huge difference in the same OS.

In summary, as each DMU has a distinct memory (Hurst) and fractal dimension over time, then the decision-maker needs to choose the better set of tools according to a lot of decision variables, creating a complex problem to solve. Thus, Table 2 details the fractal analysis in computer networks, where the fractal nature of a set of tools must always be considered separately.

Table 2

DMU mounting for multiplicative DEA models and the results of measurements without SRD DMUs.

DMU #	DMU settings	(INPUT) Fractal Dimension	(OUTPUT) TCP Average	(OUTPUT) Hurst Parameter
1	ARCH12 – VIRTUALBOX – PCNet PCI – DOCKER	1.753393	40208.64	0.6149959
2	ARCH12 – VIRTUALBOX – PCNet FAST – DOCKER	1.653052	39310.5	0.6380813
3	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER	1.514416	51299.79	0.7139144
4	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER	1.429046	57252.49	0.6303365
5	ARCH12 – VIRTUALBOX – PCNet PCI – LXC	1.819048	45993.36	0.6542005
6	ARCH12 – VIRTUALBOX – PCNet FAST – LXC	1.758913	45400.68	0.7290385
7	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC	1.732166	121542.6	0.5670684
8	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – LXC	1.583254	96962.65	0.8320479
9	ARCH12 – VMWARE – DOCKER	1.147171	45114.94	0.7176149

10	ARCH12 – VMWARE – LXC	1.310753	60895.55	0.8115613
11	FEDORA24 – VIRTUALBOX – PCNet PCI – DOCKER	1.638401	55586.69	0.5340552
12	FEDORA24 – VIRTUALBOX – PCNet FAST – DOCKER	1.722559	55881.09	0.5435163
13	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER	1.972882	140166.7	0.522739
14	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER	1.696524	120597.2	0.5856474
15	FEDORA24 – VIRTUALBOX – PCNet PCI – LXC	1.779845	49936.99	0.7378332
16	FEDORA24 – VIRTUALBOX – PCNet FAST – LXC	1.675085	54199.33	0.6086283
17	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – LXC	1.551302	115458.8	0.607753
18	FEDORA24 – VMWARE – DOCKER	1.396388	213201.2	0.6737299
19	FEDORA24 – VMWARE – LXC	1.355828	220979	0.8155753
20	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – DOCKER	1.759395	44457.31	0.7755444
21	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – DOCKER	1.69321	39228.46	0.6776561
22	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER	1.954307	144008.7	0.6322082
23	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER	1.706839	113361.1	0.5502623
24	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – LXC	1.89594	43281.01	0.6908904
25	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – LXC	1.637878	48302.53	0.6469991
26	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC	1.981541	136795.8	0.6177616
27	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – LXC	1.68567	113857.6	0.5861757
28	OPENSUSE42.2 – VMWARE – DOCKER	1.522748	80704.14	0.7692495
29	OPENSUSE42.2 – VMWARE – LXC	1.446501	213607.4	0.6591165
30	UBUNTU14 – VIRTUALBOX – PCNet PCI – DOCKER	1.708281	46006.23	0.6374863
31	UBUNTU14 – VIRTUALBOX – PCNet FAST – DOCKER	1.779336	41057.64	0.6240196
32	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER	1.729518	146780.8	0.8451504
33	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER	1.861722	95542.23	0.6515093
34	UBUNTU14 – VIRTUALBOX – PCNet PCI – LXC	1.698271	41681.55	0.7819212
35	UBUNTU14 – VIRTUALBOX – PCNet FAST – LXC	1.824841	41195.93	0.7150646
36	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC	1.743032	140712.6	0.7451546
37	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – LXC	1.518187	95067.18	0.7007306
38	UBUNTU14 – VMWARE – DOCKER	1.412104	73706.68	0.6205952
39	UBUNTU14 – VMWARE – LXC	1.621151	105727.1	0.7187318
40	UBUNTU16 – VIRTUALBOX – PCNet PCI – DOCKER	1.610912	44675.4	0.5748441
41	UBUNTU16 – VIRTUALBOX – PCNet FAST – DOCKER	1.689229	43207.5	0.7135971
42	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER	1.661918	123330.8	0.5855856
43	UBUNTU16 – VIRTUALBOX – PCNet PCI – LXC	1.394481	61309.08	0.6616379
44	UBUNTU16 – VIRTUALBOX – PCNet FAST – LXC	1.434656	57188.68	0.8082247
45	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC	2	131292.4	0.5152529
46	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – LXC	1.802845	111738.8	0.5759278
47	UBUNTU16 – VMWARE – DOCKER	1.488392	183325.3	0.8549348
48	UBUNTU16 – VMWARE – LXC	1.319123	217319.8	0.6895153

For multiplicative DEA models, we need to do a log-transform of the values from Table 2. However, as the input – Fractal dimension and the output – Hurst Parameter are both small ratio numbers then their logarithms will be negative. For this reason, a log-normalization of these small numbers must be done by dividing them by a small infinitesimal number ($\epsilon = 10^{-5}$) and further taking the logarithm of these division (*e. g.*, $\log_{10}(\frac{FractalDimension or Hurst}{10^{-5}})$). By contrast, as output – TCP bandwidth average has huge values, then it is necessary only to take the logarithms of these numbers (*e. g.*, $\log_{10}(TCP_Average)$).

Table 3 presents all multiplicative optimum values ($\emptyset_{j_0}^*$) of the objective function, efficiency scores, sensitivity analysis, and ranking from TOP3 DMUs using the SMDEA model proposed in (equation (6)). Only the TOP3 super-efficient scores are presented in the ranking column from Table 3. It should be noted that the most efficient (DMU19) that has no slack presenting a CRS, i.e. DMU19 is the optimal solution for this VN problem (see Theorem 6 of [Banker et al. 2004]. DMU19 has the biggest TCP bandwidth mean, 5th highest memory, and 4th lesser fractal dimension, reflecting a smoothing effect as in other super-efficient DMUs (DMU48 and DMU9). DMU48 has the second biggest TCP bandwidth, the second lesser fractal dimension representing a smoother effect of VN traffic, and also has a considerable memory over time. It is also important to note

which of our SMDEA formulations has no infeasible solution for zero DMU, demonstrating the math strength of this model for the prediction of VNs over time.

Table 3

$\emptyset_{j_0}^*$, efficiency, sensitivity analysis (slacks), and ranking from TOP3 DMU ranked by SMDEA model.

#	$\emptyset_{j_0}^*$	Efficiency	S1-	S2+	S3+	Ranking
DMU19	0.9911065	100.90%	0	0	0	1
DMU48	0.9990332	100.10%	0	0	0.066192	2
DMU9	0.9995659	100.04%	0	0.078160	0	3

For the sake of paper width, all formulations, results, ranking, plots, and sensitivity analyses of results may be found in our Elsevier's online dataset⁴.

For a demonstration of as the SMDEA model overcomes the traditional SDEA formulations with ratio variables observe Figure 2. Note, that in Figure 2(a) all DMUs are near one another inside of PPS due to the log-transformation of data and the Figure 2(b) just the DMU48 is out of PPS because the original SDEA formulation does not evaluate accurately DMUs that are operating higher (Ex: DMU48 is considered as efficient by SMDEA model) or lesser than their capacities. In fact, it is demonstrated it is necessary to use a frontier with geometric convexity rather than a standard convexity to reach a correct scale elasticity estimate. Yet in Figure 2(b) the DMU9 considered as efficient has a small TCP_AVG, showing that this efficiency frontier presents an incorrect efficiency frontier that needs to be rectified by the SMDEA model. These and other graphics may be found on our public dataset already mentioned.

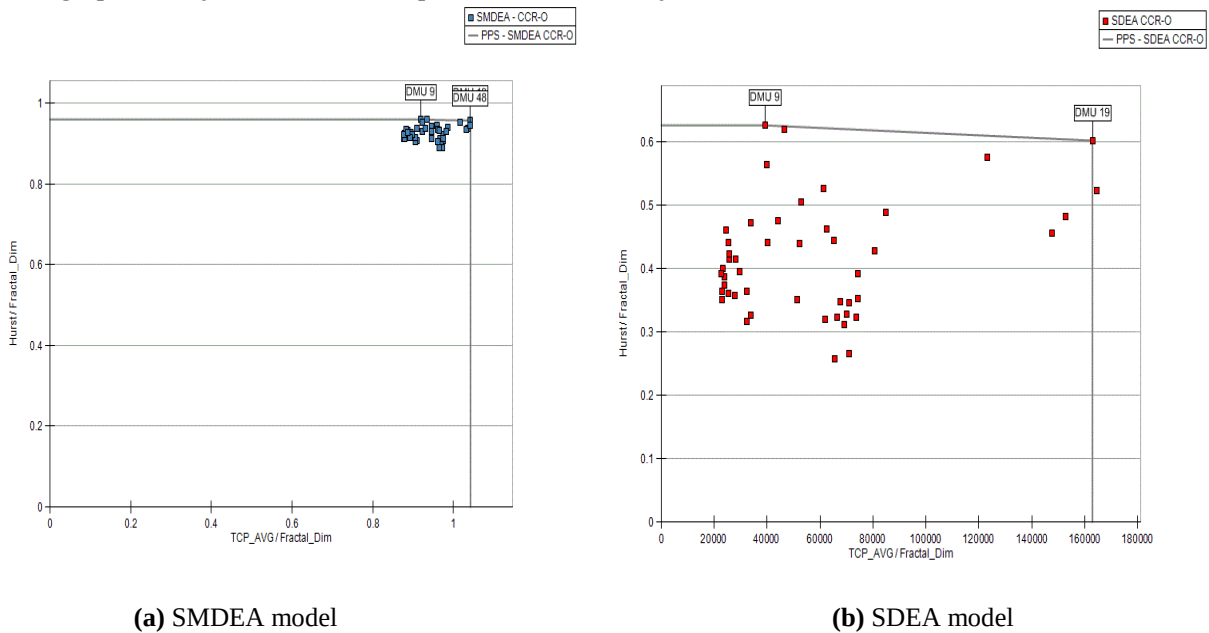


Figure 2. Graphic comparison among SMDEA versus SDEA.

In short, the highlights of the Multiplicative DEA evaluation conducted by this essay are the type II hypervisor VMWare and container-based tool LXC respectively. These applications were present in 2/3 of TOP3 SE DMU and pointed out as optimal solutions to solve the SS with LRD problem of TCP VN traffic. The OS highlights were **Fedora 24**, Ubuntu 16, and Arch 12. However, it is important to only select the set of tools of TOP1 SE DMU, not an isolated application (or OS, hypervisor, container, and so on). However, if the VN administrator picks up the TOP1 SE DMU

⁴SMDEA output-oriented results may be found at following URL: <http://dx.doi.org/10.17632/yj24sgch7m.1>

to run their applications/services, then the decision-maker will guarantee the most efficient VN services for the client on a long-time scale.

5. Conclusion and future works

This paper devises an innovative application for DEA assessment of virtual network services. For solving the optimization problem, a performance evaluation framework methodology based on Multiplicative DEA models has been proposed. Measurements were carried out to obtain the CRV of each DMU obeying the RFCs 2544 and 6815.

One can conclude that the correct use of the fractal dimension of a time series is so important to guarantee for consumers a service with better, more stable and more efficient performance over time. Only having $H > 0.5$ gives a mathematical background that there is a positive autocorrelation among the CRV, but applying roughness/smoothness concept with the rodogram method to measure fractal dimension allows the decision-maker to ensure choosing the super-efficient DMU which it will deliver smoother/efficient VN services over time.

Researchers interested may extend this to conduct a scale-up of all virtual devices using other virtualization techniques to also make a deep SS with LRD analysis using this same performance evaluation framework methodology. Another interesting application could be the use mixed DEA models with artificial intelligence (AI), deep learning, or pure AI methods for comparison of virtual networks or cloud computing service orchestration.

References

- Andersen, P., & Petersen, N. C. (1993). A Procedure for Ranking Efficient Units in Data Envelopment Analysis. *Management Science*, 39(10), 1261–1264.
- Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, 30(9), 1078–1092.
- Banker, R. D., Cooper, W. W., Seiford, L. M., Thrall, R. M., & Zhu, J. (2004). Returns to scale in different DEA models. *European Journal of Operational Research*, 154(2), 345–362.
- Banker, R. D., & Maindiratta, A. (1986). Piecewise Loglinear Estimation of Efficient Production Surfaces. *Manage. Sci.*, 32(1), 126–135.
- Banker, R. D., & Morey, R. C. (1986). Efficiency Analysis for Exogenously Fixed Inputs and Outputs. *Oper. Res.*, 34(4), 513–521.
- Bellavista, P., Callegati, F., Cerroni, W., Contoli, C., Corradi, A., Foschini, L., Santandrea, G. (2015). Virtual network function embedding in real cloud environments. *Computer Networks*, 93, 506–517.
- Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6), 429–444.
- Chowdhury, N. M. M. K., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5), 862–876.
- Crovella, M. E., & Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6), 835–846.
- de Lima, A. B., & de Almeida Amazonas, J. R. (2013). *Internet Teletraffic Modeling and Estimation*. Wharton, TX, USA: River Publishers.
- Emrouznejad, A., & Amin, G. R. (2009). DEA models for ratio data: Convexity consideration. *Applied Mathematical Modelling*, 33(1), 486–498.
- Emrouznejad, A., & Cabanda, E. (2010). An aggregate measure of financial ratios using a multiplicative DEA model. *International Journal of Financial Services Management*, 4(2), 114–126.
- Emrouznejad, A., & Yang, G. Liang. (2017). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978-2016. *Socio-Economic Planning Sciences*.
- Gneiting, T., Sevcíková, H., & Percival, D. B. (2012). Estimators of Fractal Dimension: Assessing the Roughness of Time Series and Spatial Data. *Statistical Science Donald B. Percival Is Principal Mathematician Applied Physics Laboratory*, 27(2), 247–277.

- Hall, P., & Roy, R. (1994). On the Relationship Between Fractal Dimension and Fractal Index for Stationary Stochastic Processes. *The Annals of Applied Probability*, 4(1), 241–253.
- Hurst, H. E. (1951). LONG-TERM STORAGE CAPACITY OF RESERVOIRS. *TRANSACTIONS OF THE AMERICAN SOCIETY OF CIVIL ENGINEERS*, 116, 770–799.
- Kantelhardt, J. W. (2008). Fractal and Multifractal Time Series. *Encyclopedia of Complexity and Systems Science*, 59.
- Leland, W. E., Taqqu, M. S., & Wilson, D. V. (1994). On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1), 1–15.
- Liu, W., Yan, Y., Tang, D., & Tang, R. (2012). Self-Similarity and Heavy-Tail of {ICMP} Traffic. *JCP*, 7(12), 2948–2954.
- Mandelbrot, B. (1965). Self-Similar Error Clusters in Communication Systems and the Concept of Conditional Stationarity. *Communication Technology, IEEE Transactions on*, 13(1), 71–90.
- Mandelbrot, B. B., & Taleb, N. N. (2012). mild vs wild randomness focusing on those risks that matter. Intergovernmental Panel on Climate Change (IPCC), “Summary for Policymakers,” in *Climate Change 2013 – The Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge: Cambridge University Press, 2014, pp. 1–30.
- Marques Júnior, F. D., Emrouznejad, A., Dias, K. L., Cunha, P. R. F., & de Castro e Silva, J. L. (2019). Optimising virtual networks over time by using Windows Multiplicative DEA model. *Expert Systems with Applications*.
- Park, K., Kim, G., & Crovella, M. (1997). On the Effect of Traffic Self-similarity on Network Performance. *SPIE International Conference on Performance and Control of Network Systems*, 3231, 296–310.
- Paxson, V., & Floyd, S. (1995). Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3), 226–244.
- Rizo, L., Torres, D., Dehesa, J., & Munoz, D. (2008). *Cauchy Distribution for Jitter in IP Networks*.
- Seiford, L., Charnes, A., W. Cooper, W., & A. Stutz, J. (1982). A Multiplicative Model for Efficiency Analysis. *Socio-Economic Planning Sciences*, 16, 223–224.
- Weron, R. (2002). Estimating long-range dependence: Finite sample properties and confidence intervals. *Physica A: Statistical Mechanics and Its Applications*, 312(1–2), 285–299.