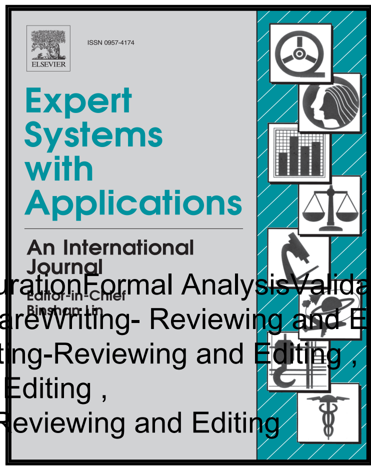


Optimising virtual networks over time by using Windows Multiplicative DEA model

Francisco Daladier Marques Júnior ConceptualizationMethodologyData curationFormal AnalysisValidationSoftwar
Ali Emrouznejad SupervisionMethodologyFormal AnalysisValidationSoftwareWriting- Reviewing and Editing ,
Kelvin Lopes Dias SupervisionValidationData curationFormal AnalysisWriting-Reviewing and Editing ,
Paulo Roberto Freire Cunha SupervisionValidationWriting-Reviewing and Editing ,
Jorge Luiz de Castro e Silva Data curationMethodologyValidationWriting-Reviewing and Editing

PII: S0957-4174(19)30318-5
DOI: https://doi.org/10.1016/j.eswa.2019.05.005
Reference: ESWA 12658

To appear in: *Expert Systems With Applications*

Received date: 25 August 2018
Revised date: 7 May 2019
Accepted date: 7 May 2019

Please cite this article as: Francisco Daladier Marques Júnior ConceptualizationMethodologyData curationFormal A
Ali Emrouznejad SupervisionMethodologyFormal AnalysisValidationSoftwareWriting- Reviewing and Editing ,
Kelvin Lopes Dias SupervisionValidationData curationFormal AnalysisWriting-Reviewing and Editing ,
Paulo Roberto Freire Cunha SupervisionValidationWriting-Reviewing and Editing ,
Jorge Luiz de Castro e Silva Data curationMethodologyValidationWriting-Reviewing and Editing ,
Optimising virtual networks over time by using Windows Multiplicative DEA model, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.05.005

Highlights:

- Develop a fractal framework for virtual network optimisation and assessment;
- Introduce the Windows Multiplicative DEA model in the presence of ratio data;
- Show that devices on virtual networks have distinct fractal behaviour over time;
- Prediction of a virtual setting with higher and stable TCP performance by long time;
- The DEA results' dataset is available at URL: http://dx.doi.org/10.17632/776sjbz7z5.5.

Optimising virtual networks over time by using Windows Multiplicative DEA model

Francisco Daladier Marques Júnior[1a], **Ali Emrouznejad**[b], **Kelvin Lopes Dias**[c], **Paulo Roberto Freire Cunha**[c], **Jorge Luiz de Castro e Silva**[d]

[a] Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Rua José Antônio da Silva, 300, Jardim Oásis, 58900-000, Cajazeiras, Paraíba, Brazil

[b] Aston Business School, Aston University, B4 7ET, Birmingham, United Kingdom

[c] Centro de Informática, Universidade Federal de Pernambuco (UFPE), Avenida Jornalista Anibal Fernandes, S/N, 50740-560, Recife, Pernambuco, Brazil

[d] LARCES, Universidade Estadual do Ceará, Avenida Doutor Silas Munguba, 1700, Itaperi, 60714-903, Fortaleza, Ceará, Brazil

Abstract

Recently, the prediction of the most efficient configuration of a vast set of devices used for mounting an optimised cloud computing services and virtual networks environments have attracted growing attention. This paper proposes a paradigm shift in modelling transmission control protocol (TCP) behaviour over time in virtual networks by using data envelopment analysis (DEA) models. Firstly, it proves that self-similarity with long-range dependency is presented differently in every network device. This study implements a novel fractal dimension concept on virtual networks for prediction, where this key index informs if the transport layer forwards services with smooth or jagged behaviour over time. Another substantial contribution is proving that virtual network devices have a distinct fractal memory, TCP bandwidth performance, and fractal dimension over time, presenting themselves as important factor for forecasting of spatiotemporal data. Thus, a continuous stepwise fractal performance evaluation framework methodology is developed as an expert system for virtual network assessment and performs a fractal analysis as a knowledge representation. In addition, due to the limitations of classical DEA models, the windows multiplicative data envelopment analysis (WMDEA) model is used to dynamically assess the fractal time series from virtual network hypervisors. For knowledge acquisition, 50 different virtual network hypervisors were appraised as decision-making units (DMU). Finally, this expert system also acts as a math hypervisor capable of determining the correct fractal pattern to follow when delivering TCP services in an optimised virtual network.

Keywords: Cloud Computing; Windows Multiplicative Data Envelopment Analysis; Fractal Expert System; Virtual Networks; Network Optimisation; Stepwise Performance Evaluation.

[*] Corresponding Author. Phone: +55 83 3532 4192
E-mail addresses: daladierjr@ifpb.edu.br, fdmj@cin.ufpe.br (Francisco Daladier Marques Júnior), a.emrouznejad@aston.ac.uk (Ali Emrouznejad), kld@cin.ufpe.br (Kelvin Lopes Dias), prfc@cin.ufpe.br (Paulo Roberto Freire Cunha), jlcs@larces.uece.br (Jorge Luiz de Castro e Silva).

Declarations of interest: none.

1. Introduction

In cloud computing, virtualization and networking are crucial since they present throughout the framework of offered services. Virtualization is a software abstraction technique enabling the partitioning of hardware resources in an isolated way between multiple virtual machines (VMs) or containers. The goal of virtualization is to enable the portability of higher-level functions, as well as the sharing and/or aggregation of physical resources (Sahoo et al., 2010).

According to Chowdhury & Boutaba (2010), network virtualization is the main component of datacenter infrastructure as well as a primary component for cloud service providers (CSP). This kind of virtualization is used to connect VMs to form logical networks and to implement traffic engineering policies. According to Wang et al. (2013), the obvious analogy between operating system (OS) virtualization and network virtualization has led the authors to refer to network virtualization as the "hypervisor to the Internet." Despite the advantages of virtualization such as server consolidation and storage, virtualization can affect network performance when the cloud infrastructure is shared among multiple tenants. Hence, good service delivery should rely on the understanding of network traffic in virtual environments and its impact on overall system performance.

Even though traditional network traffic behaviour had been widely analysed based on the findings of fractal characteristics and self-similarity (SS) (B. Mandelbrot, 1965), (Leland et al., 1994), (Crovella & Bestavros, 1997), studies based on such models have not been carried out in virtual networks. All these studies proved that traffic is statistically SS with a long-range dependency (LRD) and evinced this stochastic pattern by analysing just one network setting in a very long time, resulting in an incipient fractal analysis about a huge and unique time series.

Therefore, an understanding of the behaviour of traffic using transmission control protocol (TCP) on virtual network infrastructures plays a crucial role in stochastic prediction of a software framework able to provide excellent transport services for a long period of time. In fact, the TCP protocol provides end-to-end reliable data transmission, flow control, and congestion control in the transport layer of the Internet protocol suite. For clarification, when an application is implemented for offering services on the Internet, it does not have to deal with the complexity of providing a reliable communication channel to ensure that the data arrives correctly and in order at the destination, even when crossing congested and remote networks, because this is the role of TCP. Thus, one of purposes of this article is the introduction of a math hypervisor for forecasting the best TCP end-to-end agreements between the virtual network hypervisors to create an optimised virtual network.

The work of Cronkite-Ratcliff et al. (2016) introduced a mechanism to modulate the behaviour of TCP in virtual networks called virtualized congestion control (vCC). However, vCC does not mention the fractal behaviour of each of the TCP congestion control approaches used in virtual network hypervisors. To the best of our knowledge, the findings or a deep fractal analysis on the traffic of virtual networks have not yet been addressed by the industry or academia.

The variables evaluated are related to fractal theory, and a deep understanding of the fractal behaviour of every system is mandatory to evaluate performance correctly over time to make a correct decision for the vast dataset being appraised. This is a multi-objective problem to be solved, hence mathematical modelling must be applied to solve this complex math question. In the context of computer networks or cloud computing, these techniques are named by network optimisation tools that are a set of multi-objective problems solved by linear programming formulations, where multiple desirable objectives compete with each other and the decision-maker has to elect one of the many solutions (Iqbal et al., 2016).

Data Envelopment Analysis (DEA) was created by Charnes, Cooper, and Rhodes (1978) as a non-parametric technique that is used for evaluation in many areas such as education, supply chain, healthcare, big data, industry, and so on (Emrouznejad & Yang, 2017). Owing to the nature of its optimisation of input and output variables, DEA seeks to minimize input variables or maximize output variables, or both simultaneously, depending on the model and orientation used to solve each problem. Soleimani-damaneh (2009b) proposes a fuzzy fractal time-continuous framework for determining of the maximal flow in a generic network, the same author extended his approach creating a fuzzy DEA model (Soleimani-damaneh, 2009a), where both works are lacking the variables under appraisal. This paper proposes a paradigm shift in modelling of the TCP behaviour over time in virtual networks through DEA models.

Hence, the contributions of this paper are summarized as follows. First, it proves that the self-similarity with long-range dependency is presented differently in every network device for forecasting. This discovery is related to introduction of the fractal dimension, or Hausdorff dimension, as a measure of smoothness/irregularity to evaluate TCP traffic, as well as to predict a time series in virtual networks. In comparison to the above mentioned works on traditional fractal network traffic, this paper goes beyond for proving that every virtual network, evaluated as decision-making units (DMU), have a distinct fractal behaviour over time. It shows that virtual network devices have a distinct fractal memory, TCP bandwidth performance, and fractal dimension over time presenting themselves as important for prediction of time series data. Indeed, the fractal dimension is a robust index of smoothness or irregularity, since it does not change when a time series is scaled, translated, or corrupted by noise, and is not stationary

(Lloyd et al., 2004). Another measure used to attest to SS with LRD is the Hurst parameter or fractal memory. Secondly, for the first time in the literature, this paper proposes a windows multiplicative DEA (WMDEA) model – as math hypervisor – to assess dynamically the fractal time series from virtual network hypervisors. It has been explained in the recent literature that, efficiency values of DMUs could be miscomputed in classical DEA models in the presence of input and/or output variables in the form of ratios (Emrouznejad & Amin, 2009). Because of this issue, when the variables are ratios, the production function shows itself as non-concave in some regions, and the production possibility set (PPS) is also non-convex. Thus, the piecewise frontier of classical DEA models might be transformed into pieces of log-linear surfaces (Banker & Maindiratta, 1986). To rectify conventional DEA models to obey convexity, proportionality, and many other postulates, one must employ the multiplicative DEA models introduced by (Seiford et al., 1982). According to Olesen et al. (2017), the standard DEA formulations are commonly unsuitable if at least one input or output is in the form of a ratio. Commonly, DEA is used to evaluate the efficiency of each DMU in a single period of time, statically. However, when time series data are available, it is possible to use DEA to dynamically to evaluate a set of DMUs over time. The first of these inter-temporal models proposed in the literature was the window analysis model (Charnes et al., 1984). However, the classical window analysis is a radial model, and it also needs to be rectified for a multiplicative approach when the variables under evaluation are in form of ratios. For closing this gap, this work is introducing the windows multiplicative DEA model. Thirdly, this paper also devises a continuous stepwise fractal performance evaluation framework methodology as an expert system for virtual networks assessment and optimisation, which performs a fractal analysis as knowledge representation. The fractal performance evaluation methodology has been designed, and also may be employed as an expert system capable of solving complex problems like a human specialist (Martín de Diego, Siordia, Fernández-Isabel, Conde, & Cabello, 2019). Hence, the proposed framework has acquired knowledge on measurements, where the fractal analysis rules on experimental data per setting determined the knowledge representation. Further, the WMDEA model launched herein acts as an inference engine or math hypervisor to predict all possibilities and to appraise the best set to offer virtual network services for enhancing the TCP network performance with stability, thereby granting the customer satisfaction over a long-time scale. Finally, the use of the proposed analytical model allows the decision-maker to choose the optimal virtual network hypervisor, forecasting the best way to create a virtual network to provide services using TCP. Choosing the most efficient configuration has the stochastic guarantee of carrying a greater volume of traffic, with stability of the behaviour of the transport layer for a longer period of time. So, the empirical results further explained show that this adjusted dynamic DEA formulation devised is able to solve the virtual network problem herein raised. For another side, the WMDEA formulation launched is also suitable to be applied in every problem where all the decision variables are in the form of a ratio.

The rest of this article is organised as follows. Section 2 promotes a background of related works, main concepts, application of DEA models for solving cloud computing problems, and formalisms employed by the research. Section 3 presents and details the continuous stepwise fractal performance evaluation framework methodology with the topology used on measurements. In Section 4, the results are appraised and discussed as a numerical example. Finally, Section 5 presents the conclusions.

2. Background

In this section, a brief review on self-similarity and fractal dimension is presented, along with the formulations that will be further used in this research. After that, the DEA, windows analysis, and multiplicative DEA models are detailed, as well as their applications. Finally, the presentation of the WMDEA model is depicted.

2.1 Self-Similarity & Fractal Dimension

SS was discovered by Mandelbrot (1965) as a way of invariance related to the bursts of traffic in telecommunication channels on time scales (Mandelbrot & Van Ness, 1968). Fractal geometry, another technique introduced by Mandelbrot (1982), was created with the aim to mathematically accommodate objects from nature which exhibit patterns of irregularity and fragmentation, by identifying a family of jagged shapes named **fractals**.

The self-similar behaviour of computer networks was initially proposed by (Leland et al., 1994), and was used to predict generated Ethernet traffic through the analysis of measurements data in the Bellcore Morristown Research and Engineering Center for a period of three and a half years (August/1989 to February/1992) in the same network setting. The cited work showed that the network´s traffic presents a burstiness pattern which remains in an extremely wide range of time scales. Thus, it was proven that the network traffic had a different behaviour than the formal models previously employed on telecommunication traffic (Markov´s or queueing theory, Poisson-related models, etc.). The work of Oliveira et al. (2003) showed this identical fractal pattern on wireless networks, also using just one network setting. In the end, it is necessary to improve methods used in modelling the traffic and the performance of computer networks for obeying their fractal nature.

In the recent literature, Wang et al. (2013) investigated the peak power management of datacenters based on fractals. They presented a spatiotemporal analysis of power demand of datacenters executed by Microsoft over a six-month period, from July to December 2011. These data are from eight (8) representative server clusters executing a myriad of workloads including web-search, email, map-reduce jobs, and other cloud services, serving millions of users globally. The work of Wang et al. (2013) creates abstractions for capturing power demands in the form of peaks and valleys on clouds. The Hurst parameter is computed to identify the presence of SS, and uses some fractal plots to show the SS with LRD inside these cloud services workloads.

In Markovian theory, it is suggested that communication channels have only two states, a good state (with errors of low probability), and a bad state (with a high probability of errors), and a probability of transitioning between the states (Gilbert, 1960). In channels that have SS in each burst, there is a statistically independent grouping between them, so a model would be necessary of not only two states, but of a hierarchy of several master stochastic processes.

One common way to calculate a fractal coefficient of a time series is by using the linear regression method, also named rescaled-range (R/S) analysis (Mandelbrot & Wallis, 1969). Through this R/S analysis, the Hurst parameter is calculated (in honour of the engineer who developed this calculation, to understand the unusual behaviour of floods of a perennial dam on the Nile River in an Egyptian benchmark of 100 years of floods) (Hurst, 1956).

Such analysis begins by dividing a time series (returns) of size L into d sub-series of size n (Weron, 2002). Then, for each sub-series $m = 1,\dots, d$:

1) Find out the mean (E_m) and the standard deviation (S_m).

2) Normalize the data ($Z_{i,m}$) by subtracting the mean of the samples $X_{i,m} = Z_{i,m} - E_m$ for $i = 1, \dots, n$.

3) Create a cumulative time series:

$$Y_{m,1} = \sum_{j=1}^{i} X_{j,m} \quad \textbf{(1)}$$

4) Find the range $R_m = \max\{Y_{1,m}, \dots, Y_{n,m}\} - min\{Y_{1,m}, \dots, Y_{n,m}\}$.

5) Compute the re-scaling of the range R_m / S_m.

6) Calculate the average value of the rescaled-range for all sub-series of length n:

$$\left(\frac{R}{S}\right) = \frac{1}{d}\sum_{m=1}^{d} \frac{R_m}{S_m} \quad \textbf{(2)}$$

7) By the value of the R/S ratio, its statistics asymptotically follow the relation below:

$$\left(\frac{R}{S}\right)_n \sim cn^H \quad \textbf{(3)}$$

8) Finally, the value of *H* is calculated by linear regression over a growing time series sample:

$$\log\left(\frac{R}{S}\right) = \log c + H \log n \quad \textbf{(4)}$$

The parameter *H* is used to capture the intensity of the scale dependence in every time series. So, in the case of:

a) $0.5 > H < 1$, the process will be persistent in retaining higher memory, as big as *H*´s value had;

b) $0 > H < 0.5$, the process is anti-persistent, or without memory; and

c) $H = 0.5$, then the time series are highly random or chaotic, keeping in a fractal-Brownian motion (fBm).

The Hurst parameter has been used to measure the fractal memory of many time series to explain if each time series had a pattern where high averages follow high averages or low means keep following low means, or was chaotic, as well as without memory. The majority of researches which present the SS on distinct types of computer networks used *H* alone as an unique fractal index to demonstrate this pattern in just one big time series. Yet, this essay made a deep fractal analysis of 50 distinct virtual networks' settings to choose the most efficient virtual network hypervisor over others by using an inter-temporal DEA technique proposed here, acting like a math hypervisor from TCP traffic or each one protocol employed under similarly appraisal.

All values of *H* were computed using linear regression of time series acquired on measurements per setting. The R/S method described in equations 1-4 was implemented in R[1] with results equal to those of the pracma[2] package. However, according to Weron (2002), there are several other ways to obtain *H,* such as a maximum likelihood estimator (MLE), detrended fluctuation analysis (DFA), periodogram regression, Hill estimator (Hill, 1975), Whittle´s procedure (Paxson & Floyd, 1995), and several more.

The fractal geometry is based on the "law of large numbers" (Mandelbrot & Taleb, 2012), owing to the fact that the average of a high random set of numbers tends to be close to the mean of the entire population. The term "Brownian domain of attraction" was coined by Mandelbrot & Wallis (1969), and states that stochastic process are characterised by three properties: (i) the law of large numbers; (ii) the central limit theorem; and (iii) the asymptotic independence between past and future. Thus, in regards to *H*´s values, one knows that: a) when *H>0.5,* there is a positive correlation between the past and future traffic data, i.e. these processes have infinite memory over time; b) when *H=0.5,* the property of number (iii) is violated, showing which of

[1]R – The R project for Statistical Computing (see: https://www.r-project.org)
[2]Pracma – The R package for more advanced functions in numerical analysis, with a special view on optimisation and time series routines (see: https://cran.r-projet.org/web/packages/pracma/pracma.pdf)

the fBM´s time series do not belong to the Brownian domain of attraction; and c) when $H<0.5$, the correlation among past and present is negative, i.e. the traffic data is memoryless, proving itself to be without utility for prediction.

The autocorrelation function (ACF) has an important role in identifying an index of dependence present in time series variables. In a self-similarity stochastic process, the ACF indicates a strong relation among values that are repeated with adherence, owing to its positive autocorrelation memory (Shang et al., 2007).

The fractal dimension or Hausdorff dimension *D* was initially defined as a parameter to measure a degree of irregularity from coastlines. *D* increases when the burstiness increases, and decreases when there is a smoothness effect (Mandelbrot, 1975). As stated by Campbell & Abhyankar (1978), a formal definition of a fractal is related to an object, where its dimension (D) is higher than its topological dimension (D_{topo}). This D_{topo} is associated with a topological dimension of the plot of a random function, where D_{topo} has 1 value for a time series (our case), 2 for a map or graphic, and 3 for a volume (Dauphiné, 2013).

D is related to the slope of a log–log graph of its madogram near the origin (Bez & Bertrand, 2011). Thus, even a non-fractal time series has a fractal dimension related to a roughness/smoothness value, i.e. serving as a measure of irregularity or stability of a set. The madogram is considered as a mathematically superior version of a Hall–Wood estimator, and it is concurrently more outlier-resistant and efficient than others of its competitors (Gneiting et al., 2012).

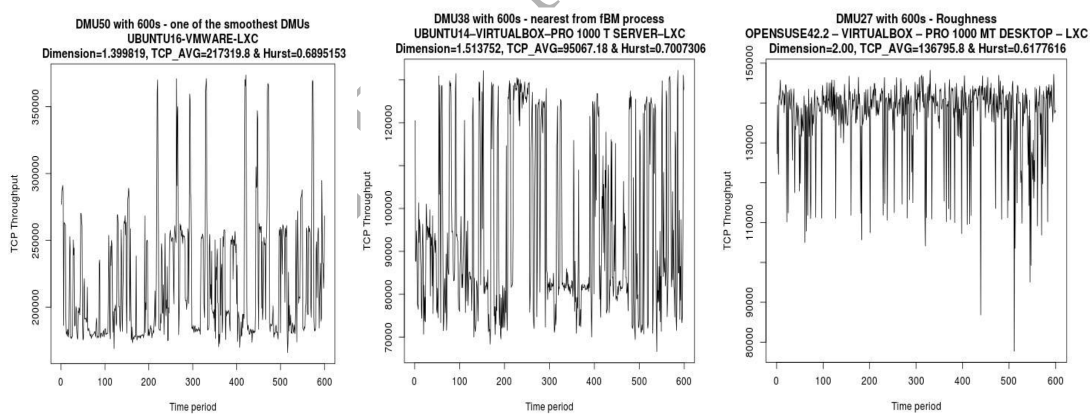
To calculate *D*, more accurate formalisms have been devised such as Box-count, rodogram, Genton, and madogram (Gneiting et al., 2012). These prior methods may help to obtain the right decision-making process to predict a chance phenomenon over spatiotemporal data using the fractal approach. For this reason, the madogram method was chosen to compute *D*, because it works well with extreme values (Bez & Bertrand, 2011).

The madogram is a measure of variability, and describes itself as the relationship between similarity and distance of the points x and $x+h$ (Stein et al., 2008). Thus, let $Z(x)$ and $Z(x+h)$ be two values from variable Z located under points x and $x+h$, where these points are separated by a lag of size h. The madogram is computed as the mean of the sum of all differences between pairs of values which are divisible by 2, as follows:

$$D = \gamma(h) = \frac{1}{2n}\sum_{i=1}^{n}|Z(x_i) - Z(x_i + h)| \quad \textbf{(5)}$$

After this computation for each time series, one can arrive at three interesting conclusions regarding *D*. These are: 1) when *D* tends to 1, then the time series presents a **smoothing effect**, i.e. the traffic has a TCP bandwidth that keeps more stable over time; 2) when *D* tends to 2, then the time series shows a **roughness** or **burstiness effect**, that is, the traffic is irregular with infinite variance and high variability on time (Gneiting et al., 2012); and 3) when *D* tends to 1.5, the time series exhibits a fBM or random walk. The same conclusion was initially stated by Mandelbrot (1967), which affirms that *D´s* value of coastlines from South Africa (*D=1.02*) has the smoothest effect in the atlas. In this same essay, Britain had the roughest effect (*D=1.25*) as compared to all appraised coastline maps.

For a graphical illustration of fractal behaviour over time, one can see the volatility patterns of time series from three selected DMUs that were aggregated in 600 seconds as examples of smooth, fBM, and roughness effects in an uniform time lag. Note that in Figure 1(a), the traffic averages are more stable, i.e. the peaks are more regular. In Figure 1(b), there is a highly random traffic pattern characterising a chaotic stochastic process. Figure 1(c) presents a burstiness effect because the means are jagged over time. Additionally, the fractal dimension's effect is noted in the three images from Figure 1 with a similar interpretation to that in Mandelbrot (1982), i.e. when *D* increases its value, invisible details from virtual network traffic become very apparent rather than separate when *D* decreases, such as in growing the resolution of an image when *D* is bigger, and reducing it when *D* is lesser. Further, Figure 1 is interesting for the demonstration of the importance of forecasting on virtual network services, not only of fractal dimensions but also as related to the three fractal variables used in this research, for network optimisation that seeks to elect a DMU with a higher and more stable TCP performance bandwidth for a long-time span.



(a) One of the smoothest settings. **(b)** A setting near of the fBM pattern. **(c)** A setting exhibiting burstiness effect.

Fig. 1. The TCP bandwidth behaviour over time of some settings according to the fractal dimension perspective.

The fractal dimension has successfully been applied for comparison of one dataset with another for ranking (Hall & Roy, 1994). To close this gap in computer networks, we are proposing to use this index as an input variable to be minimised by an inference engine. All madogram dimensions were computed using an R package named fractaldim[3]. Alternatively, it is a rule of thumb that *H* is an output variable regardless of the method considered to solve this problem, as the aim is always to increase *H*. Thus, seeking to maximise *H* plus a higher TCP transfer rate average is a math surety of more significant memory and bandwidth performance over time, and is also linked to a smoother transport layer.

Hence, it is mandatory to learn how to analyse a time series data for targeting predictions regarding extreme events or future trend behaviour using the fractal analysis. In short, the fractal analysis is a formal tool applied to represent stochastic processes in a dimensionally small representation (Kantelhardt, 2008).

Recently, fractal tools and analysis has become a growing field of research in expert systems with applications being used to solve problems where it is necessary to predict values from continuous random variables data. For example, in López-Ortega & López-Popa (2012), a suite was proposed to assist in the creation of musical pieces by applying fractals, fuzzy logic, and expert systems. The research of Przystalski & Ogorzałek (2017) is to show the usefulness of fractal methods when applied to multilevel images and binarisation methods for skin cancer pattern recognition. The work of Florindo & Bruno (2013) proposes a multi-resolution texture analysis based on application of fractal descriptors, with superior results as compared to classical methods. In Ni et al. (2011), a stock trend prediction is presented based on fractal selection and a support vector machine, to forecast the direction of the daily stock price index.

Indeed, this essay shows that every time series related to continuous random variables per DMU has a distinct dimension, TCP performance, and fractal memory over time in an independent manner, bringing new conclusions on self-similarity theory in a way never explored before by academia. For this reason, we use fractal variables to identify how to maintain higher network stability performance over time using TCP by application of a math hypervisor herein launched.

2.2 DEA, Multiplicative models, Windows Analysis and its applications

DEA is a mathematical modelling method to measure the relative performance of decision making units (DMUs), but is mainly applied for managerial purposes. DEA compares DMUs, to evaluate their efficiency through a linear combination of input variables employed to produce outputs.

We pick up this WMDEA technique owing to its non-parametric characteristic, i.e. the variables being evaluated do not need to be converted before its use, nor is it necessary to

[3]Fractaldim – R package for computation of fractal dimension available at URL: https://cran.r-projet.org/web/packages/fractaldim/fractaldim.pdf

compute any *a priori* statistic. As an optimisation approach, DEA utilises the simplex algorithm to compute the efficiency of each DMU beside objective functions and their respective restrictions. There are some DEA models to choose from, depending on the aim that the decision-maker wants to target. For example, the model chosen may depend on if he/she wants to minimise the number of inputs without changing the output values, or to maximise the output without modifying the input values, or to collectively minimise the input and maximise the output in so-called allocative or non-radial models.

The choice of the proper model is mandatory to realise a positive decision-making process, and thus to understand how the model should be used to point out the variables, interpretation of results, and planning in COOPER framework (Emrouznejad & Witte, 2010). To revisit the main areas where DEA has been applied in its 40 years of usage, as well as to show research areas more interesting for employing DEA and the name of the journals where this technique has been published, see Emrouznejad & Yang (2017).

The introductory DEA model was created by Charnes et al. (1978) to measure the efficiency of a US inclusive education program named Follow Through. For this reason, this model carries the initials of its creator's names (Charnes, Cooper, and Rhodes) – CCR. In the meantime, a huge number of DEA models were created following the premises of constant returns to scale (CRS) and variable return to scale (VRS), to distinguish if a DMU was working efficiently and correctly or not.

The breakthrough DEA model was Banker, Charnes & Cooper (BCC), developed by Banker et al. (1984). BCC introduces a model with the capacity to appraise DMUs which are in high competition, by trying to generate the maximum amount of outputs that the available technology can produce compatible with input utilisation, and following technical, economic, and scale points of view. Hence, BCC is related to a VRS supposition formed by a mixture of technical efficiency and efficiency scale scores, and creates a measure of the most productive scale size (MPSS) per DMU. VRS mainly serves to figure out if each one of the DMUs is being executed under its optimal capacity (increased return to scale), on optimal capacity (constant return to scale), or below capacity (decreased return to scale). In contrast to CRS models, the VRS models produce distinct efficiency scores just by changing the model orientation.

However, the classical DEA models present a well-known significant shortcoming; they are not mathematically suitable to work with floating-point numbers. In DEA literature, these numbers are called **ratios**. The main problems of using ratios are that they violate the convexity, proportionality (Emrouznejad & Cabanda, 2010), minimum extrapolation, ray unbounded, and monotonicity postulates (Banker & Maindiratta, 1986).

To solve DEA´s ratio problems, it is mandatory to transform the variables to be appraised, to reach a geometric convexity granting an accurate interpolation of the observed production of possibilities. Hence, multiplicative models, e.g. as devised by Seiford et al. (1982), are the right way to rectify these issues. It is worth pointing out that the majority of DEA papers which use classical DEA models present a miscalculation of the efficiency frontier by the use of input/output variables in the form of ratios without converting to a multiplicative DEA formulation, thus presenting wrong conclusions.

Hence, the entirety of the DEA models mentioned earlier are only related to a unique time of evaluation of the DMUs, i.e. they are a static type of DEA performance evaluation. For this reason, DEA models should also be developed to appraise the dynamic behaviour of DMUs over time. In that regard, the Windows DEA (WDEA) model was proposed by Charnes et al. (1984) for measuring the efficiency of each DMU in an independent, inter-temporal manner. In this first work, an input-oriented windows DEA analysis was used to evaluate the US army recruiting command over time. In the meantime, many applications using the windows DEA analysis or their modified versions can be found in the literature, with examples including: measuring the impact of economic growth on the environmental efficiency of the electricity sector from the USA (Halkos & Polemis, 2018), evaluating the efficiency of hotels across all of 20 regions in Italy, to learn the relationship between the size and efficiency of the Italian hospitality sector (Pulina et al., 2010), and assessing the energy and environmental efficiency of 29 administrative regions of China during the period of 2000-2008 (Wang et al., 2013), among many others.

As in fractal theory, if each set is independent of one another, then the variables chosen to solve this multi-objective problem are correct. Nevertheless, no previous study tried to fix the classical DEA windows analysis for computing ratio variables, as this study does. Thus, the next subsection will be the presentation and formalisation of the new windows multiplicative DEA formulation, as devised.

It is essential to highlight the multiplicative DEA formulations that are modified variants of classical DEA radial formulations (CCR or BCC) in order to work acceptably with proportions. Thus, if a DEA model is related to a standard DEA radial model, such as the windows DEA analysis, then this model needs to be rectified using a multiplicative manner as well.

2.3 DEA and their models applied to solve cloud-computing problems

The work of Raja & Ramaiah (2016) proposed the consumer and cloud DEA (CCDEA) model. CCDEA is an evaluation framework for the trust assessment of CSPs based on *n*-levels, where the variables are captured from online forms answered by cloud consumers and stored in databases. CCDEA was compared with the LKJ model (a super-efficiency DEA (SDEA) formulation proposed by Li et al. (2007)) and CCR. CCDEA does not treat stochastic variables and additionally only works with subjective variables, thereby decreasing the confidence of its decision process.

The essay of Truong (2014) used DEA to evaluate cloud-based supply chain software through subjective variables. This research used fuzzy logic and analytic network process (ANP) to weigh the variables and manage the uncertainty. The variables were populated by cloud computing experts, i.e. these variables are also not stochastic. Another drawback is that the DEA model used was not specified.

The work of Jatoth et al. (2016) used a modified SDEA model with an ANP to weigh variables and choose the most efficient CSP. Some cloud performance metrics that were not directly correlated to the problem to be solved were evaluated. Another drawback of the cited work is that the authors populated their variables using empirical data of a site called cloudharmony.com, i.e. their data are not real, nor obtained by any synthetic benchmark tool, and neither stochastic as the work describes itself.

All prior works, however, only evaluate the DMUs statically, and do not convert their variables properly to obtain the scale elasticity on DEA models.

2.4 The Windows Multiplicative DEA model

It is factual that math models need to be designed to solve the problem in question, to fix the cause of possible miscomputations from results. As mentioned earlier, it is indispensable to adjust the standard DEA models to compute efficiency frontiers in the presence of ratios, transforming classic models into multiplicative forms (Banker & Morey, 1986).

As a starting point towards understanding every DEA model, it is obligatory to know the meaning of PPS. PPS is the distance of each unit under assessment to its border, which will define if a DMU is/is not in an efficient frontier, i.e. the PPS is used to determine an efficient subset linked to the PPS´s set of data. As the windows DEA analysis evaluates DMUs dynamically over time, consider when a DMU utilizes *m* inputs to produce *s* outputs per time lag *T*. Let a set of *n* DMUs be appraised, with *t* a sub-vector of a *T* time series (where *t*=1-2-3-4-5, 2-3-4-5-6, 3-4-5-6-7, 4-5-6-7-8, 5-6-7-8-9, and 6-7-8-9-10) with the same length of the window, with DMU_j^t $(for\ each\ j = 1, \dots, n)$ linked to an input vector of $X_j^t = (x_{ij}^t, \dots, x_{mj}^t)$ and an output vector of$Y_j^t = (y_{ij}^t, \dots, y_{sj}^t)$.Then, the PPS must be formalised as:

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

Let us consider the fractal dimension, TCP bandwidth performance, and Hurst parameter as one input and two outputs, respectively, obtained by measurements on virtual networks. Considering only two DMUs (see Table 2), according to P^t both setups, for example DMU_1^t and DMU_2^t (or each one combination), should create a virtual DMU_{12}^{t*}. However, this is not mathematically possible owing to the proportionality issue raised when the input or output variables cannot be increased or decreased proportionally, because of the lack of support of scale elasticity in these models. Another issue is convexity, corresponding to a miscomputation of the weighted sum of ratios that does not correspond to the right values, and that is caused by error on efficiency scores (Emrouznejad & Amin, 2009).

The work of Banker et al. (1984) brings postulates of inefficiency, ray unbounded, and minimum extrapolation, used to "fine tune" classical DEA models in the presence of ratios. Multiplicative models are classified as quantitative estimates of returns to scale (RTS) in DEA, and are appropriate to make accurate scale elasticity estimates (Banker et al., 2004). Multiplicative models are also suitable for problems with geometric convexity, i.e. their production function is non-concave in some regions with PPS being non-convex, as exemplified by the three variables already mentioned.

Previous works in DEA tried to relax the convexity axiom in multiplicative models and several models, to rectify this restriction as described in Emrouznejad & Amin (2009). The work of Emrouznejad et al. (2010) presents a multiplicative model that uses the concept of a geometric mean with non-dimensional unit invariance as a property. However, we keep following the convexity, proportionality (Emrouznejad & Cabanda, 2010), minimum extrapolation, monotonicity (Banker & Maindiratta, 1986), and other postulates cited earlier. Hence, the usage of geometric convexity to provide an accurate interpolation of the observed production of possibilities is obligatory in the presence of ratios.

The whole of multiplicative models has been applied in static DEA evaluations. This study is launching a new windows multiplicative DEA formulation suitable for assessments of DMUs over time. So, it is necessary to amplify the PPS by the size of the window to be analysed, as shown by P^t before.

For the sake of understanding, *n* is a symbol related to the number of DMUs, *k* is the number of periods to be evaluated, *p* is the length of the window ($p \leq k$), and *w* is the number of windows. All symbols used are related to the following formulas in Table 1.

Table 1: Formulas used to compute some indexes from windows multiplicative DEA model.

Description	Formula
Number of windows	$w = k - p + 1$
Number of DMUs in each window	$n * p$
Number of distinct DMUs	$n * p * w$

Source: Adapted by the authors from Cooper et al.(2007) and Yang & Chang (2009).

We use output-oriented WMDEA model, since we maximise output variables while keeping the input variables constant. This formulation seeks the lesser values of the fractal dimension and higher values for the average TCP bandwidth, as well as large values of the Hurst parameter. Based on the multiplicative PPS shown over time, the WMDEA model proposed in its initial form is as follows:

$$\max \emptyset_{j_0}^t \quad \textbf{(6)}$$

$$Subject\ to$$

$$\prod_{j=1}^{n} x_{ij}^{\lambda_j^t} \leq x_{i_0}^t, i = 1, \dots, m$$

$$\prod_{j=1}^{n} y_{rj}^{\lambda_j^t} \geq \ {y_{r_0}^t}^{\emptyset_{j_0}^t}, r = 1, \dots, s$$

$$\emptyset_{j_0}^t, \lambda_j^t \geq 0, j = 1, \dots, n$$

To convert the inequalities and identification of slacks in $(m+s)^t$ constraints of the equation (6), it is only necessary to add the multiplicative coefficients ${s_i}^{-t}$ and ${s_r}^{+t}$ in the formulation of inputs and outputs respectively, then:

$$x_{i_0}^t = e^{{s_i}^{-t}} * \prod_{j=1}^{n} x_{ij}^{\lambda_j^t}, i = 1, \dots, m \quad \textbf{(7)}$$

$${y_{r_0}^t}^{\emptyset_{j_0}^t} = e^{-{s_r}^{+t}} * \prod_{j=1}^{n} y_{rj}^{\lambda_j^t}, r = 1, \dots, s$$

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

$$\emptyset_{j_0}^t\ free\ and\ {s_i}^{-t} \geq 0\ \&\ {s_r}^{+t} \geq 0$$

On behalf of the axioms of geometric convexity, monotonicity, and the minimum extrapolation of Banker & Maindiratta (1986), the closed and convex set $\hat{P}^t$ of multiplicative observed data must be formalised as:

$$\hat{P}^t = \{(\hat{X}^t, \hat{Y}^t) | \sum_{j=1}^{n} \hat{x}_{ij}^{\lambda_j^t} \leq \hat{x}_i^t, i = 1, \dots, m, \sum_{j=1}^{n} \hat{y}_{rj}^{\lambda_j^t} \geq \hat{y}_r^t, r = 1, \dots, s, for\ all\ \lambda_j^t \geq 0\}$$

The one-to-one mapping of variables in a logarithm basis over time transforms the original multiplicative DEA formulations as the equations (6) and (7), in a log-linear programming model obeying the strict monotonicity. Applying logarithms in (7), the windows multiplicative DEA model proposed is presented in a log-linear programming formulation, as given further.

$$\max \emptyset_{j_0}^{t} \tag{8}$$

$$Subject\ to$$

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

$$\sum_{j=1}^{n} \lambda_j^t * \hat{y}_{r,j}^t - \emptyset_{j_0}^t * \hat{y}_{r,j_0}^t - {s_r^+}^t = 0, r = 1, \dots, s$$

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

$$\emptyset_{j_0}^t\ free\ and\ {s_i^-}^t \geq 0\ \&\ {s_r^+}^t \geq 0$$

Where:

- ${\emptyset_{j_0}}^t$ − efficiency value of DMUj_0 in the current window of evaluation;
- x_{ij}^t − i^{th} input of DMUj in the current window of evaluation;
- $x_{i_0}^t$ − input of DMU under evaluation (DMUj_0) in the current window of evaluation;
- y_{rj}^t − r^{th} output of DMUj in the current window of evaluation;
- $y_{r_0}^t$ − output of DMU under evaluation (DMUj_0) in the current window of evaluation;
- λ_j^t − array of weights referring to the peers or benchmarks for DMUj in the current window of evaluation.

Hence, being a maximisation model, the efficiency score is computed by the inverse of its optimum value per DMU in each time series on the window of evaluation. Thus, to obtain the efficiency score of each DMU over time in (8), it is still necessary to make the computation below:

$$efficiency\_score_{DMU_j}^t = 1/\emptyset_{j_0}^t \tag{9}$$

Finally, to rank the networks, we compute the average *efficiency_score* of each ${DMU_j}^t$ within each window, and an overall average *efficiency_score* of each ${DMU_j}^t$ over all windows.

The work of Olesen et al. (2017) creates the notion of potential ratio (PR) efficiency for the models with CRS and VRS, where a DMU is only considered as fully efficient if and only if it has an efficiency score equal to one, and the sum of all slacks is zero. Still, according to Olesen et al. (2017), the PR satisfies the postulates of selective convexity and being freely disposable of all input-output variables. In the selective convexity model of Olesen et al. (2017), some inputs and outputs are ratios, but in our WMDEA model, all variables are ratios. Hence, by logarithm transformation, we keep the main axioms of DEA including convexity, proportionality, ray unbounded, minimum extrapolation, monotonicity, etc. Similarly, the WMDEA was implemented only using linear programming under CRS and VRS, but as the performance evaluation framework is a stepwise methodology and the developed network optimisation approach needs to elect a smaller set of efficient DMUs, then the WMDEA picked up to obey the CRS assumption. Furthermore, we also implemented various static traditional and multiplicative DEA formulations, with all the types of orientations following radial and non-radial assumptions, and we found that the WMDEA model produce most reliable results amongst all formulations evaluated. The WMDEA formulation is a ratio-convex as in Olesen et al. (2017), but exclusively works with ratios, and not with absolute data (numerator and denominator).

According to Banker et al. (2004), a DMU is of full efficiency if and only if its *efficiency*_score is equal to unity, and all slacks are at zero. Furthermore, towards the virtual network optimisation, the slacks are important only to show if a DMU is fully efficient and are not important for fine-tuning DMUs, because it is almost impossible to reach out this modulation in each piece of software used to mount the DMUs on diverse cloud and virtual environments. When DEA is employed for managerial purposes, the use of slacks may serve for transforming the inefficient DMUs into the efficient ones.

The WMDEA model is implemented using LINDO[4] software, and it was made an utter log-transformation of data (see Table 4) for running it on DEA-SOLVER[5] and PIM-DEA[6], where all results were quite similar. It is worth pointing out that the same results will be obtained if one uses the equation (10) as the objective function, rather the equation (8).

$$\max \emptyset_{j_0}^{t} + \varepsilon * \left(\sum_{i=1}^{m} s_i^{-t} + \sum_{i=1}^{s} s_r^{+t}\right) \quad \textbf{(10)}$$

3. Continuous Stepwise Fractal Performance Evaluation Framework Methodology

This section explains the proposed non-parametric math hypervisor approach, based on the WMDEA model that uses fractal variables to offer more efficient virtual network TCP services over time. This expert system framework is in continuous evolution for providing optimised virtual networks services over time, and according to its stepwise nature, many variables were/will be dropped by the way.

[4]**L**inear, **In**teractive, and **D**iscrete **O**ptimizer (LINDO): see http://www.lindo.com
[5]DEA-SOLVER: See http://www.saitech-inc.com/products/prod-dsp.asp
[6]Performance Improvement Management (PIM-DEA): See http://www.deasoftware.co.uk/

All measurements made obeyed the methodologies of evaluation of the so-called request for comments (RFC) 2544[7] and its extension, the RFC 6815[8]. These methodologies describe assessments that might be used to appraise the performance of network devices as virtual network hypervisors using any benchmark application; either one defines how to report the results of measurements with a duration of only 60 seconds.

To increase the accuracy and precision of results, it was made the design of the experiments. Thus, the number of repetitions of the first DMU was computed according to equation (11). The equation computes how many assessments would be necessary to obtain a confidence interval of the desired width. Then, let *z* be linked to the confidence interval (95% in our case) for the TCP throughput average related to a pre-computed *t*-student table for a time series with 60 seconds, δ be the standard deviation of the time series, and *e* be referred to as a percentage of where the average value is within an actual mean value. Hence, this value reflects that the marginal error of results to up or down must be 10%, i.e. 5% under the first value from the confidence interval, and 5% above the second and last value from the confidence interval. For this reason, the allowed error is 0.05. Given that the proposed model is considered as an expert system, it is possible to assume different values. Lastly, $\bar{x}$ is the average of each one of the time series.

$$n = \left(\frac{z_{1-\alpha} * \delta}{e * \bar{x}}\right)^2 \quad \textbf{(11)}$$

Putting the values of the first DMU of Table 2 in equation (11) to obey the RFCs, we have $z = 2$, $\delta = 3070.099$, $e = 0.05$, and $\bar{x} = 40130.15$, then $n = 9.3644$. Taking the rounded value of *n* ($n \approx 10$) for repetition of the experiments for an entire set of DMUs, this value of *n* is the number of time series that must be evaluated by the new WMDEA formulation. Hence, this work implements measurements with a total time of 600 seconds (10 repetitions) by the virtual network hypervisor/DMU to yield more accuracy and precision, acquiring statistical significance for analysis. The chosen variables are not correlated with each other, and then the inter-temporal DEA model proposed brings correct frontier results. Iperf[9] is a popular network benchmark tool that has been used to synthetically produce the maximum amount of TCP traffic packages per device.

The **guest-to-guest-to-container (GGC)** topology has been implemented to assess the performance of two virtual routers that act as virtual network hypervisors, being represented by a virtual machine (VM) with a container-based strategy and its opposite VM in another side, as shown in Figure 2. All experiments were performed starting with a Guest VM named device under a test (DUT)/iperf server/node sink, through the virtual switch from a type-II hypervisor

[7]Request for Comments of number (RFC) 2544: See https://www.ietf.org/rfc/rfc2544.txt
[8]Request for Comments of number (RFC) 6815 See https://www.ietf.org/rfc/rfc6815.txt
[9]iperf – available at URL: https://iperf.fr

(VirtualBox versus VMWare) which sends its packets/flows to the opposite Guest VM. Further, the Guest VM forwards its packets/flows via the **virtual bridge** to the respective container software used (called traffic generator (TG)), and vice versa. For guaranteeing network reachability among the devices, routing schemes have been employed so that the information can flow from the source to the destination (and vice versa). All TCP window size generated by iperf was of 85.3 Kbytes; so, all virtual network hypervisors should send a similar amount of traffic data, rather than this study that will show the existence of a substantial difference of performance and fractal behaviour per DMU.

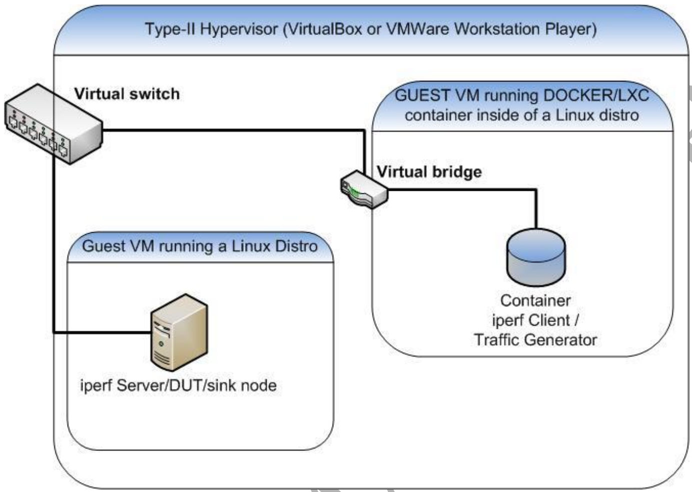


Fig. 2. Guest-to-guest-to-container (GGC) experimentation topology.

At the process of **assembling scenarios**, all VMs used on assessments had one GB of virtual RAM (vRAM) and one virtual CPU (vCPU). The process of **execution of measurements** was done employing five (5) distinct Linux distributions (distro), which are: Arch 12, Fedora 24, OpenSUSE 42.2, Ubuntu 14.04 Server, and Ubuntu 16-04 Server. In each Linux distro, its container tool is followed by the same OS name, e.g. when Fedora24 was used on the Guest VM its container (Docker/LXC) was also Fedora 24.All emulated network devices from VirtualBox (such as AMD PCNet PCI II, AMD PCNet FAST III (standard), Intel PRO / 1000 MT Desktop, and Intel PRO / 1000 T Server (82543GC)) and VMWare were used to assemble the virtual devices. In short, it is recommended that all network-emulated interfaces and all sets of possibilities would be appraised, analysing their performance and fractal behaviour over time. Figure 3 depicts the performance evaluation framework in an expert system context with a flowchart.

The **fractal evaluation of data** is an important layer of the evaluation framework, and must be properly explained. In this process, the overall fractal computation (fractal dimension, TCP bandwidth average, and Hurst parameter) related to the TCP ´s network traffic is performed per DMU, creating the fractal rules used for knowledge representation explained in the self-similarity section.

The last layer is the **network optimisation**. Here, the decision-maker can pick up the best set of virtual network hypervisors to deliver the most efficient virtual network services using TCP over time, as chosen by the inference engine or math hypervisor from the WMDEA model executed on the prior layer. So, the WMDEA results are analysed for electing the best DMU to offer the most effective virtual network services, according to the TCP´s fractal behaviour and performance along the time.

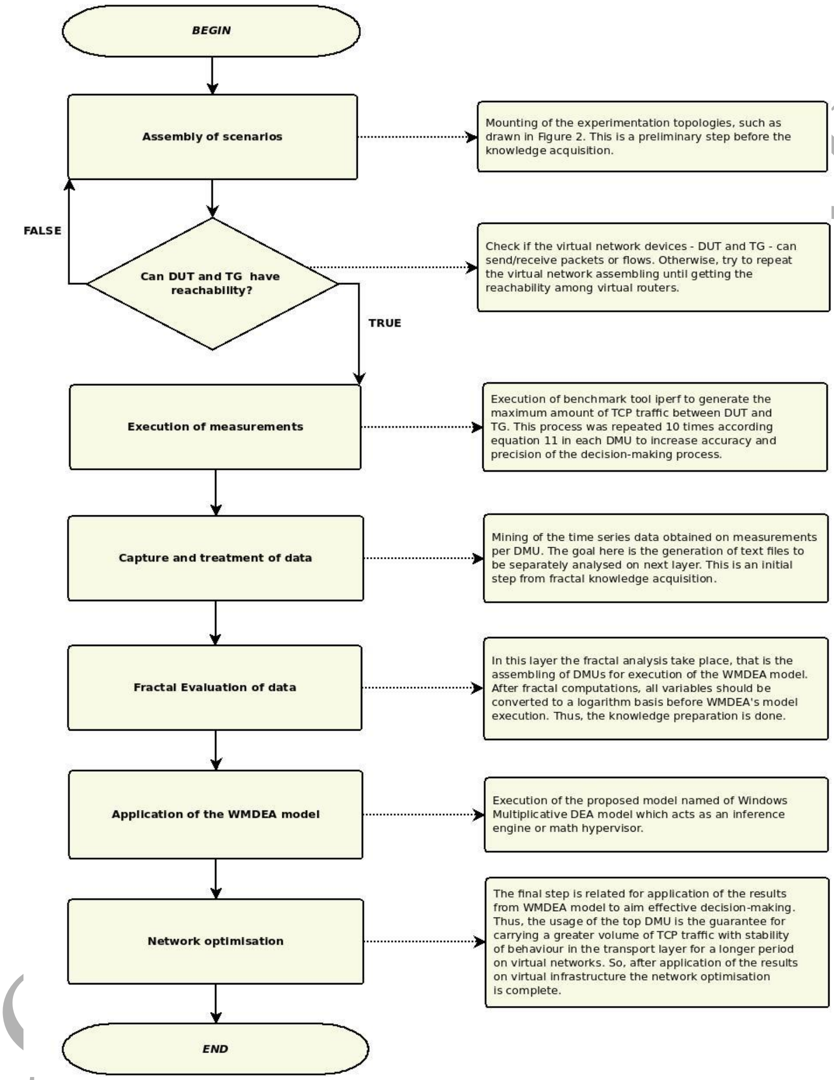


Fig. 3. Flowchart for the proposed evaluation framework.

4. An application with detailed explanation

This section details the results obtained from the last three processes using continuous stepwise fractal performance evaluation framework methodology. Firstly, the fractal evaluation layer is linked to the fractal knowledge representation of the experimental data, i.e. all fractal indexes per device are computed, such as fractal dimension, TCP bandwidth average, and Hurst parameter. However, Tables 2 and 3 show just some results of measurements using TCP on virtual networks, in a GGC topology of 50 DMUs that were evaluated in 10 different and independent time series. Each line of the cited tables is considered as a DMU, i.e. a separated set of tools to mount a virtual network hypervisor for delivering TCP services, and the results are ordered alphabetically. For economy of size, only the first and the last time series are exhibited in Tables 2 and 3. For all of the data and results, the public online dataset[10] can be accessed. Nevertheless, Tables 2 and 3 must be presented as a whole because it is mandatory to make the fractal analysis for an entire set of experiments in every case, as well as for evidencing the TCP bandwidth performance and fractal differences among all DMUs. However, these measurements are the reflection of the unique hardware settings used for experimentations, meaning that if either the hardware or even any version of the tools to mount the virtual network hypervisor are changed, the fractal results and TCP performance per device will be totally different.

As can be observed in Tables 2 and 3, the DMU names follow the same pattern. Initially, each one brings the Linux distro used, followed by its hypervisor tool with its network-emulated device interface (if it exists), and finally, the container-based tool employed on measurements. As the proposed framework is a continuous stepwise methodology, some variables were deleted along the way, for example, the variances and different fractal dimension indexes, even though every fractal dimension index brings some stochastic information as the variance in a straightforward manner.

As one of the main contributions of this work, it is noted that the DMUs in Table 2 and 3 have a different fractal dimension, TCP bandwidth performance, and fractal memory (*H*) over time, in an independent manner. This proves that every transport layer forwards the traffic services with smooth or jagged behaviour over time, launching a new stochastic understanding of each flow control´s end-to-end agreement done via TCP traffic between source and destination. Hence, if the network administrator changed only one set of the tools, its fractal dimension, TCP bandwidth performance, and memory may increase/decrease over time. This breakthrough conclusion launches a new approach to assess the virtual network traffic with regard to the fractal analysis. For the sake of understanding, consider that in Table 2, the setting with the biggest TCP transfer rate average (DMU_{19}) has 518.470% higher TCP bandwidth performance than the smaller TCP throughput average (DMU_{42}) among all settings. To make this calculation, we obtain the TCP bandwidth mean of each DMU, and compute its percentage using $\frac{(DMU_{42}-DMU_{19})}{DMU_{19}} * 100\%$ as an example. Note that this performance´s variation pattern remains in all DMUs, and consequently in all of the time series under appraisal. Thus, the

[10]Dataset of results - All results from Windows Multiplicative DEA model versus traditional Windows DEA model with sensitivity analysis are available at: http://dx.doi.org/10.17632/776sjbz7z5.5

decision-maker needs to elect the best set of tools related to the fractal variables chosen, as well as the TCP performance, generating a multicriteria problem to solve on virtual networks.

Tables 2 and 3 detail a fractal evaluation of data never seen in computer networks, developing a turning point in knowledge representation on network/computer system performance assessment, wherein the fractal behaviour of a set of tools must always be analysed independently. Indeed, the entire set of possibilities of configurations might still be appraised, with each one considered as a time series for an accurate predictive and optimised decision-making process. On another hand, these results reflect only the hardware performance that was used for these experiments.

Table 2: Time series 1 of 10 of the complete fractal performance evaluation.

TIME SERIES 1				
#	**VIRTUAL SETTING WITH ENTIRE DESCRIPTION/DMU NAME**	**(I) Fractal Dimension**	**(O) TCP Bandwidth Average in kbps**	**(O) HURST**
1	ARCH12 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.7852090	40130.15	0.6183077
2	ARCH12 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.4331920	38563.87	0.5827894
3	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.6611160	48895.95	0.7726983
4	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.4217680	61124.30	0.5478090
5	ARCH12 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.5399000	43250.30	0.6675938
6	ARCH12 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.7937460	50667.15	0.6319927
7	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.6319600	117504.80	0.6514350
8	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.7304820	125510.00	0.5893940
9	ARCH12 – VMWARE – DOCKER – GGC	1.2531070	38543.98	0.6082866
10	ARCH12 – VMWARE – LXC – GGC	1.2511560	45294.92	0.7683398
11	FEDORA24 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6287100	54551.93	0.5367957
12	FEDORA24 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	2.0000000	56601.18	0.4353884
13	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.9029140	140952.80	0.5034535
14	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.6859220	125431.20	0.5479436
15	FEDORA24 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.8320190	44044.10	0.7534088
16	FEDORA24 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.5709730	55193.28	0.5544900
17	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.9475290	138655.80	0.4322091
18	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.6866830	104569.00	0.5835325
19	FEDORA24 – VMWARE – DOCKER – GGC	1.5064380	231164.10	0.7329962
20	FEDORA24 – VMWARE – LXC – GGC	1.4125050	206155.70	0.5759402
21	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6267230	47673.52	0.6713221
22	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.8371570	40177.47	0.6209980
23	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.7357380	143066.20	0.5398554
24	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.7628200	110523.10	0.5563244
25	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.7132080	44282.12	0.6401471
26	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.4001780	47227.53	0.7249415
27	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.9162090	136452.10	0.5981887
28	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.5161770	119193.40	0.5418862
29	OPENSUSE42.2 – VMWARE – DOCKER – GGC	1.4227910	78763.75	0.5508929
30	OPENSUSE42.2 – VMWARE – LXC – GGC	1.7329260	209598.30	0.5778727
31	UBUNTU14 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6147890	40339.02	0.6057512
32	UBUNTU14 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.8080330	42718.55	0.7461761
33	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.6526870	139935.00	0.5463445
34	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	2.0000000	94764.83	0.6809666
35	UBUNTU14 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.3720660	43070.47	0.6899069
36	UBUNTU14 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.8425400	44271.33	0.6012703
37	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	2.0000000	135731.80	0.7624681
38	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.9059010	91079.55	0.6279829
39	UBUNTU14 – VMWARE – DOCKER – GGC	1.4639170	75191.63	0.6818153
40	UBUNTU14 – VMWARE – LXC – GGC	1.6138730	108825.80	0.6693658
41	UBUNTU16 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.7330460	45606.92	0.6231167
42	UBUNTU16 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.7889940	37376.77	0.4993106
43	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	2.0000000	140416.30	0.4966934
44	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.7457590	124744.50	0.6112910
45	UBUNTU16 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.5677790	63122.28	0.6886994
46	UBUNTU16 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.4224960	61896.33	0.6145179
47	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	2.0000000	133828.10	0.6155002
48	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.7410760	125823.10	0.5863453
49	UBUNTU16 – VMWARE – DOCKER – GGC	2.0000000	153872.80	0.7219129
50	UBUNTU16 – VMWARE – LXC – GGC	1.2596530	213151.10	0.6522807

ACCEPTED MANUSCRIPT

Note: All data in this table has been obtained by measurements on virtual networks. Output and input variables are computed, respectively, by equations (4) and equations (5), mean of TCP bandwidth in kbps in the first time series of ten. Results in details are available at WMDEA Mendeley´s public dataset, including: time series and windows analysis with diverse width of windows, DEA models, comparison and plots.

It is observed that in Tables 2 and 3, there are some values in red when $H<0.5$, presenting Noah´s effect, i.e. they are an isolated fact that hardly happens. Hence, these DMUs are presenting a short-range dependency (SRD) over time. Therefore, these DMUs should be dropped from the evaluation; however, because they do not cause an imbalance in the input-output matrices used for computation on the WMDEA model, the SRD variables must be calculated as obligatory. Furthermore, as the DEA model proposed is non-radial and automatically targets the DMUs with higher H values, DMUs with SRD must be maintained.

Table 3: Time series 10 of 10 of the complete fractal performance evaluation.

TIME SERIES 10				
#	**VIRTUAL SETTING WITH ENTIRE DESCRIPTION/DMU NAME**	**(I) Fractal Dimension**	**(O) TCP Bandwidth Average in kbps**	**(O) HURST**
1	ARCH12 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.8741320	39820.07	0.5147301
2	ARCH12 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.5613560	39022.27	0.6822004
3	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.3487800	49447.10	0.6837165
4	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.5518510	60248.58	0.6283762
5	ARCH12 – VIRTUALBOX – PCNet PCI – LXC – GGC	2.0000000	45230.37	0.7551821
6	ARCH12 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.6241240	40399.67	0.6716724
7	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.7659410	126680.00	0.4934648
8	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.4125940	65236.30	0.7393426
9	ARCH12 – VMWARE – DOCKER – GGC	1.3953410	43181.63	0.7616619
10	ARCH12 – VMWARE – LXC – GGC	1.4392520	49597.13	0.6967720
11	FEDORA24 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6966430	56905.22	0.6030927
12	FEDORA24 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.7343300	53544.65	0.5458348
13	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.8200410	140923.80	0.5592290
14	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.6511500	123265.80	0.5702495
15	FEDORA24 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.7414020	52742.78	0.5736909
16	FEDORA24 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.4686970	56279.52	0.5509028
17	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.7554000	138221.40	0.5633642
18	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.4737900	129838.00	0.5306936
19	FEDORA24 – VMWARE – DOCKER – GGC	1.3996640	208904.50	0.6114788
20	FEDORA24 – VMWARE – LXC – GGC	1.2643750	295963.00	0.7102069
21	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6395040	48031.82	0.6591689
22	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.7139660	39535.63	0.6484466
23	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.8643840	147263.80	0.5351587
24	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.8217410	136428.80	0.6189044
25	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – LXC – GGC	2.0000000	38364.76	0.6526942
26	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.7244650	50379.95	0.6763286
27	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	2.0000000	133626.40	0.6396128
28	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.3687650	105891.20	0.5738729
29	OPENSUSE42.2 – VMWARE – DOCKER – GGC	1.7767970	78939.85	0.5726417
30	OPENSUSE42.2 – VMWARE – LXC – GGC	1.3381750	222197.20	0.7378387
31	UBUNTU14 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.7823050	40194.51	0.6970738
32	UBUNTU14 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.7911080	40927.50	0.6635941
33	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.8211900	138834.80	0.6510289
34	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.9731210	98211.78	0.6155717
35	UBUNTU14 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.8528940	40747.51	0.6635777
36	UBUNTU14 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.8015880	40284.15	0.6948283
37	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.8405040	133292.00	0.6551226
38	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.4807780	94701.20	0.5423257
39	UBUNTU14 – VMWARE – DOCKER – GGC	1.4933670	76047.93	0.6803698
40	UBUNTU14 – VMWARE – LXC – GGC	1.3399310	110447.10	0.6646793
41	UBUNTU16 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	1.6408410	43935.75	0.5564401
42	UBUNTU16 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	1.7482260	44108.12	0.6057765
43	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	1.7284000	140463.80	0.5984046
44	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	1.7735000	119848.00	0.6313183
45	UBUNTU16 – VIRTUALBOX – PCNet PCI – LXC – GGC	1.6733870	58509.80	0.7924933
46	UBUNTU16 – VIRTUALBOX – PCNet FAST – LXC – GGC	1.5982990	47367.53	0.7010329
47	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	1.6008330	133921.90	0.4694506
48	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	1.6349440	112703.40	0.6007014
49	UBUNTU16 – VMWARE – DOCKER – GGC	1.3193650	210437.20	0.6053503
50	UBUNTU16 – VMWARE – LXC – GGC	1.1774390	233141.90	0.6147959

Note: Similar to Table 2, all data of this table was obtained by measurements on virtual networks, and represent data in the 10th time series of ten and all detailed results are available at WMDEA Mendeley´s public dataset.

As observed in Tables 2 and 3, the entire values are still not in the logarithm basis, but it is obligatory, as in every multiplicative DEA model, to perform a log-normalization of these numbers before the model execution. However, because the input fractal dimension and the output Hurst Parameter, are both small numbers, their logarithms will be negative. Hence, the logarithm computation of small numbers must be done by dividing them by a small infinitesimal number ($\varepsilon = 10^{-5}$). After that, the logarithm of this division is taken ($e.g. \log_{10}(\frac{FractalDimension\ or\ HurstParameter}{10^{-5}})$). For another side, the TCP throughput average retains big values, and then only a simple calculus of the logarithms of these numbers is necessary ($e.g. \log_{10}(TCP\_Average)$). For this reason, Table 4 exhibits the variables from Table 3 as converted for the logarithm basis explained at this paragraph, as a basic requirement from multiplicative DEA models. It should be noted to make the same log-transformation of the variables in all periods under appraisal.

Table 4: Time series 10 of 10 of the utter fractal performance evaluation after log-transformation of input and output variables.

TIME SERIES 10 WITH LOG-TRANSFORMATION OF VARIABLES				
#	**VIRTUAL SETTING ENTIRE DESCRIPTION/DMU NAME**	**(Î) Fractal Dimension**	**(Ô) TCP AVG**	**(Ô) HURST**
1	ARCH12 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	5.2728002	4.6001020	4.7115796
2	ARCH12 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	5.1935019	4.5913125	4.8339120
3	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	5.1299411	4.6941408	4.8348761
4	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	5.1908500	4.7799468	4.7982197
5	ARCH12 – VIRTUALBOX – PCNet PCI – LXC – GGC	5.3010300	4.6554301	4.8780517
6	ARCH12 – VIRTUALBOX – PCNet FAST – LXC – GGC	5.2106192	4.6063778	4.8271575
7	ARCH12 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	5.2469762	5.1027081	4.6932562
8	ARCH12 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	5.1500174	4.8144893	4.8688457
9	ARCH12 – VMWARE – DOCKER – GGC	5.1446804	4.6352990	4.8817622
10	ARCH12 – VMWARE – LXC – GGC	5.1581368	4.6954565	4.8430907
11	FEDORA24 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	5.2295905	4.7551521	4.7803841
12	FEDORA24 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	5.2391317	4.7287161	4.7370612
13	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	5.2600812	5.1489843	4.7475897
14	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	5.2177865	5.0908426	4.7560649
15	FEDORA24 – VIRTUALBOX – PCNet PCI – LXC – GGC	5.2408990	4.7221630	4.7586780
16	FEDORA24 – VIRTUALBOX – PCNet FAST – LXC – GGC	5.1669322	4.7503504	4.7410750
17	FEDORA24 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	5.2443761	5.1405753	4.7507892
18	FEDORA24 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	5.1684356	5.1134018	4.7248439
19	FEDORA24 – VMWARE – DOCKER – GGC	5.1460238	5.3199478	4.7863814
20	FEDORA24 – VMWARE – LXC – GGC	5.1018759	5.4712374	4.8513849
21	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	5.2147125	4.6815290	4.8189967
22	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	5.2340022	4.5969887	4.8118742
23	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	5.2705354	5.1680960	4.7284826
24	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	5.2604866	5.1349061	4.7916236
25	OPENSUSE42.2 – VIRTUALBOX – PCNet PCI – LXC – GGC	5.3010300	4.5839325	4.8147098
26	OPENSUSE42.2 – VIRTUALBOX – PCNet FAST – LXC – GGC	5.2366544	4.7022577	4.8301578
27	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	5.3010300	5.1258923	4.8059171
28	OPENSUSE42.2 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	5.1363289	5.0248599	4.7588157
29	OPENSUSE42.2 – VMWARE – DOCKER – GGC	5.2496378	4.8972963	4.7578830
30	OPENSUSE42.2 – VMWARE – LXC – GGC	5.1265129	5.3467386	4.8679614
31	UBUNTU14 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	5.2509820	4.6041667	4.8432788
32	UBUNTU14 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	5.2531218	4.6120152	4.8219025
33	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	5.2603553	5.1424983	4.8136003
34	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	5.2951537	4.9921636	4.7892786
35	UBUNTU14 – VIRTUALBOX – PCNet PCI – LXC – GGC	5.2678506	4.6101011	4.8218918
36	UBUNTU14 – VIRTUALBOX – PCNet FAST – LXC – GGC	5.2556555	4.6051342	4.8418775
37	UBUNTU14 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	5.2649368	5.1248041	4.8163226
38	UBUNTU14 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	5.1704900	4.9763555	4.7342602
39	UBUNTU14 – VMWARE – DOCKER – GGC	5.1741666	4.8810874	4.8327450
40	UBUNTU14 – VMWARE – LXC – GGC	5.1270824	5.0431543	4.8226122
41	UBUNTU16 – VIRTUALBOX – PCNet PCI – DOCKER – GGC	5.2150665	4.6428180	4.7454184
42	UBUNTU16 – VIRTUALBOX – PCNet FAST – DOCKER – GGC	5.2425976	4.6445185	4.7823124
43	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – DOCKER – GGC	5.2376443	5.1475644	4.7769949
44	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – DOCKER – GGC	5.2488312	5.0786308	4.8002484
45	UBUNTU16 – VIRTUALBOX – PCNet PCI – LXC – GGC	5.2235964	4.7672286	4.8989956
46	UBUNTU16 – VIRTUALBOX – PCNet FAST – LXC – GGC	5.2036580	4.6754807	4.8457384
47	UBUNTU16 – VIRTUALBOX – PRO 1000 MT DESKTOP – LXC – GGC	5.2043460	5.1268516	4.6715899
48	UBUNTU16 – VIRTUALBOX – PRO 1000 T SERVER – LXC – GGC	5.2135029	5.0519370	4.7786586
49	UBUNTU16 – VMWARE – DOCKER – GGC	5.1203650	5.3231225	4.7820068
50	UBUNTU16 – VMWARE – LXC – GGC	5.0709384	5.3676203	4.7887310

Note: The data in this table is the 10th time series of ten after their decision variables have been passed by a logarithmic transformation. This log-normalization is a mandatory process of multiplicative DEA models. Moreover, complete results are available on the WMDEA Mendeley´s public dataset.

Table 5 presents the average efficiency scores of the WMDEA model by window per DMU, with one input variable (fractal dimension) and 2 outputs (TCP bandwidth average and Hurst Parameter), plus a complete ranking by using the proposed inference engine. The window size is 5, and we have 10 time series with 6 windows (1-2-3-4-5, 2-3-4-5-6, 3-4-5-6-7, 4-5-6-7-8, 5-6-7-8-9, and 6-7-8-9-10) for evaluation of the fractal behaviour on virtual networks per DMU over time.

Only the top two most efficient DMUs (DMU_{20} and DMU_{50}) are highlighted in Table 5. The top two DMUs have a quite regular performance over time, keeping the fractal dimension near a smooth value (*D* tends to be 1), a high TCP bandwidth average, and big memory over time. Alternatively, the worst settings ranked are linked to DMUs that present a roughness effect of the fractal dimension (*D* tends to be 2). This means that the DMUs with higher TCP bandwidth, with *D* nearest to 1 and greater H value have a better end-to-end TCP´s flow control agreement over time, influencing for maintaining of quality of services (QoS) in regards of TCP traffic between client and server along the time. It is important to highlight that, the TCP´s flow control is related to the end-to-end QoS delivery, while the TCP´s congestion control is linked to the networks that the packets are crossing.

Table 5: Average of the efficiency scores by window per DMU calculated using the WMDEA model.

	Window 1-2-3-4-5	Window 2-3-4-5-6	Window 3-4-5-6-7	Window 4-5-6-7-8	Window 5-6-7-8-9	Window 6-7-8-9-10	Average all Windows	Ranking
DMU1	0.9544960	0.9468240	0.9429840	0.8981770	0.8940230	0.8902994	0.9211340	47
DMU2	0.9586800	0.9517300	0.9585730	0.9052600	0.9047840	0.9037211	0.9304580	32
DMU3	0.9809680	0.9765330	0.9734890	0.9257300	0.9224970	0.9223650	0.9502640	13
DMU4	0.9780030	0.9729380	0.9757640	0.9293150	0.9260840	0.9240278	0.9510220	12
DMU5	0.9595320	0.9516750	0.9587910	0.9075720	0.9066990	0.9048267	0.9315160	29
DMU6	0.9593810	0.9550070	0.9467100	0.9012280	0.8998350	0.9022974	0.9274100	37
DMU7	0.9605220	0.9476960	0.9456550	0.9392100	0.9362570	0.9319253	0.9435440	21
DMU8	0.9635550	0.9554330	0.9623350	0.9371800	0.9342940	0.9358558	0.9481090	14
DMU9	0.9896650	0.9845240	0.9827070	0.9327840	0.9317470	0.9309630	0.9587320	8
DMU10	0.9799660	0.9750820	0.9806750	0.9411670	0.9408100	0.9399382	0.9596060	6
DMU11	0.9527950	0.9465360	0.9476750	0.9070590	0.9050010	0.9048826	0.9273250	38
DMU12	0.9382600	0.9387210	0.9427890	0.9082180	0.9067560	0.9065120	0.9235430	45
DMU13	0.9446950	0.9352190	0.9334760	0.9292860	0.9220510	0.9227154	0.9312400	31
DMU14	0.9525820	0.9445760	0.9452370	0.9328340	0.9301090	0.9338343	0.9398620	22
DMU15	0.9588990	0.9480220	0.9453280	0.9013440	0.8999150	0.9007728	0.9257140	42
DMU16	0.9514530	0.9361280	0.9341230	0.8968080	0.8949950	0.8993484	0.9188090	49
DMU17	0.9393610	0.9329280	0.9312890	0.9261090	0.9146110	0.9166846	0.9268310	40
DMU18	0.9579300	0.9475350	0.9527280	0.9369150	0.9357920	0.9380123	0.9448190	18
DMU19	0.9887370	0.9750890	0.9745920	0.9731600	0.9658370	0.9649462	0.9737270	3
DMU20	**0.9851240**	**0.9770050**	**0.9744750**	**0.9757780**	**0.9751150**	**0.9821107**	**0.9782680**	**2**
DMU21	0.9576900	0.9466030	0.9467650	0.8950780	0.8942170	0.8996362	0.9233310	46
DMU22	0.9589950	0.9586170	0.9592280	0.9054310	0.9086370	0.9075044	0.9330690	28
DMU23	0.9414430	0.9289880	0.9289810	0.9307540	0.9233710	0.9257259	0.9298770	33
DMU24	0.9563450	0.9500670	0.9497630	0.9302880	0.9234990	0.9275634	0.9395880	23
DMU25	0.9473100	0.9364340	0.9359790	0.8932950	0.8936700	0.8918247	0.9164190	50
DMU26	0.9652950	0.9513780	0.9499940	0.9031930	0.9017480	0.9042729	0.9293130	34
DMU27	0.9461050	0.9333380	0.9268170	0.9239770	0.9168410	0.9184384	0.9275860	36
DMU28	0.9577310	0.9461150	0.9419090	0.9305210	0.9257450	0.9310937	0.9388520	24
DMU29	0.9672910	0.9589510	0.9622100	0.9345960	0.9295540	0.9282797	0.9468130	16
DMU30	0.9794690	0.9763550	0.9759880	0.9742170	0.9657490	0.9689723	0.9734580	4
DMU31	0.9544650	0.9478310	0.9523040	0.9016500	0.9046340	0.9011700	0.9270090	39
DMU32	0.9531810	0.9437550	0.9424880	0.8953570	0.8950990	0.8956930	0.9209290	48
DMU33	0.9627090	0.9502170	0.9524930	0.9431970	0.9350080	0.9361267	0.9466250	17
DMU34	0.9606220	0.9462490	0.9394960	0.9184380	0.9174840	0.9170233	0.9332190	27
DMU35	0.9651460	0.9529500	0.9466940	0.9168230	0.9183390	0.9159457	0.9359830	25
DMU36	0.9507010	0.9451070	0.9495840	0.8977480	0.9048980	0.9049523	0.9254980	43
DMU37	0.9636170	0.9503490	0.9485860	0.9377080	0.9318750	0.9314938	0.9439380	20
DMU38	0.9773790	0.9743820	0.9737740	0.9464520	0.9433260	0.9377278	0.9588400	7
DMU39	0.9819400	0.9757190	0.9763630	0.9367990	0.9353060	0.9335654	0.9566150	9
DMU40	0.9706720	0.9575840	0.9549790	0.9346890	0.9278460	0.9368952	0.9471110	15
DMU41	0.9506540	0.9455430	0.9536780	0.9068860	0.9043440	0.9058429	0.9278250	35
DMU42	0.9545060	0.9512340	0.9477460	0.9014930	0.8985240	0.9007893	0.9257150	41
DMU43	0.9474240	0.9365710	0.9326210	0.9274750	0.9213490	0.9233247	0.9314610	30
DMU44	0.9587650	0.9486180	0.9496760	0.9378740	0.9352090	0.9353974	0.9442570	19
DMU45	0.9779220	0.9711440	0.9703360	0.9321860	0.9297380	0.9296238	0.9518250	11
DMU46	0.9831470	0.9786720	0.9782070	0.9329330	0.9282800	0.9244682	0.9542850	10
DMU47	0.9372310	0.9203880	0.9270900	0.9245810	0.9166610	0.9194039	0.9242260	44
DMU48	0.9537950	0.9444610	0.9436260	0.9258700	0.9225060	0.9238298	0.9356810	26
DMU49	0.9711400	0.9618100	0.9667620	0.9703810	0.9666160	0.9670905	0.9673000	5
DMU50	**0.9957430**	**0.9828830**	**0.9808960**	**0.9808890**	**0.9690530**	**0.9711770**	**0.9801070**	**1**

Note: Efficiency results of each DMU tabulated by the window size equal to five. DMUs in bold are considered as the most efficient for providing TCP services over time according to WMDEA formulation. Similar to previous tables, the complete results are available on the WMDEA Mendeley´s public dataset.

It is also important to show the substantial difference between the efficiency score of both output-oriented models, respectively, i.e. the new WMDEA model and the original windows DEA formulation. For comprehension and size, we shall present only the top three (*DMU_{50}, DMU_{20}, and DMU_{19}*) and two worst (*DMU_{16} and DMU_{25}*) DMU efficiency scores from Table 5, related to a whole windows average of efficiency scores onto Table 6. Hence, Table 6 reflects the efficiency scores of the original windows DEA model of these 5 DMUs selected from Table

5, even though these ranking of settings are quite different in both windows DEA efficiency frontier results.

Table 6: Five DMUs selected to show the original windows DEA results**.**

	Efficiency Window 1-2-3-4-5	Efficiency Window 2-3-4-5-6	Efficiency Window 3-4-5-6-7	Efficiency Window 4-5-6-7-8	Efficiency Window 5-6-7-8-9	Efficiency Window 6-7-8-9-10	Average all Windows
DMU50	0.9484340	0.8159210	0.7974990	0.7970640	0.6698380	0.6901776	0.9484340
DMU20	0.8311280	0.7603730	0.7379560	0.7375860	0.7280850	0.7991422	0.8311280
DMU19	0.8776580	0.7471200	0.7413330	0.7211820	0.6426810	0.6370103	0.8776580
DMU16	0.5596880	0.4737190	0.4597950	0.2698690	0.2632330	0.2798766	0.5596880
DMU25	0.5328880	0.4674980	0.4652570	0.2650390	0.2669250	0.2615000	0.5328880

Note: Selected DMUs from WDEA results for comparison with WMDEA´s efficiency values.

Tables 5 and 6 exhibit the overhaul of results related to the new WMDEA model, besides the original windows DEA formulation. In Table 7, the percentage difference only between the five DMUs selected from the WMDEA model (Table 5) and the original one (Table 6) can be considered. It is quite common in multiplicative DEA models that the efficiency scores have values near 90% to 100% for all DMUs, because of the small differences between log-converted variables. In summary, the radial assumption is not always suitable to produce correct efficiency results, where it is mandatory to employ a non-radial model such as multiplicative models with the capacity to allow multiple input and output variables in the form of ratios, reflecting a fair proportionality in results. It is important to highlight the work of Färe & Lovell (1978) that launched the Russel measure as another way for computing efficiency by introducing a non-radial approach, as well as the research from Banker et al. (2004) that presents non-radial models as ideals to treat the changes in a mixed set of variables, where input and output orientations can be analysed simultaneously.

Table 7: Comparison between five DMUs selected for the windows multiplicative DEA model (Table 5) versus the original windows DEA model (Table 6).

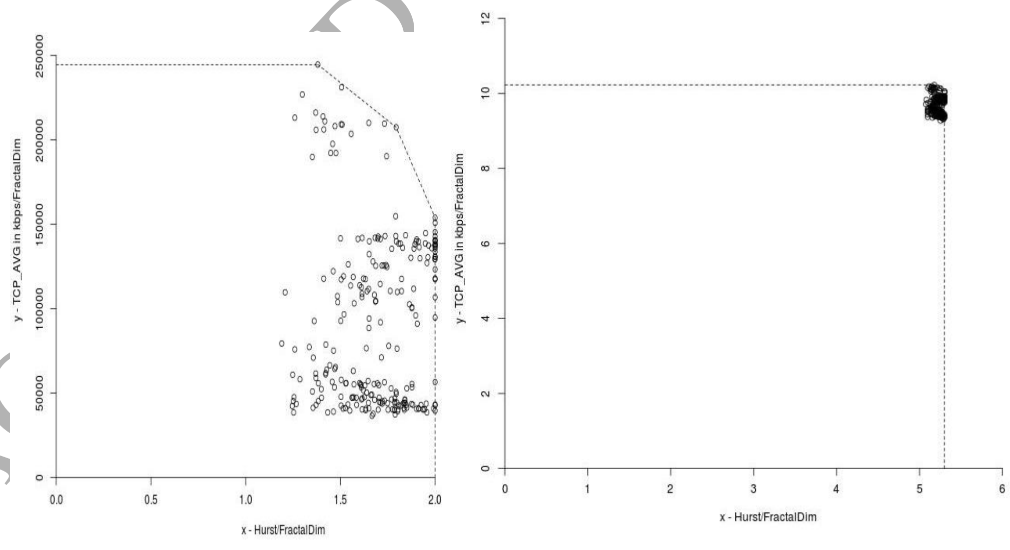
	Difference in % Window 1	Difference in % Window 2	Difference in % Window 3	Difference in % Window 4	Difference in % Window 5	Difference in % Window 6
DMU50	-4.7511429	-16.9869644	-18.6968634	-18.7406855	-30.8770205	-28.9338990
DMU20	-15.6321854	-22.1730465	-24.2714374	-24.4104663	-25.3334198	-18.6301279
DMU19	-11.2344428	-23.3792640	-23.9340213	-25.8927882	-33.4586255	-33.9848925
DMU16	-41.1754596	-49.3959111	-50.7778773	-69.9078415	-70.5883351	-68.8800691
DMU25	-43.7472449	-50.0767748	-50.2919268	-70.3301911	-70.1315999	-70.6780931

Note: The percentage difference of the selected DMUs´ efficiency scores from Table 6 between the WMDEA versus the WDEA model.

Another important tool to prove that the WMDEA model makes a correct evaluation of data in the presence of ratio variables is by showing a comparison among PPS and efficiency frontiers of the model proposed herein, versus the original WDEA formulation. Figure 4 illustrates the PPS comparison in the first window of analysis for both models under appraisal. In Figure 4(a), some DMUs are close or spread out along efficiency frontier, rather than Figure 4(b) where DMUs are near each other in the PPS. That is, the original Windows DEA formulation does not properly treat some DMUs which are operating higher or lower than their capacities (see also Emrouznejad & Amin (2009)). Note that in Figure 4(a), which some DMUs considered as efficient have a small TCP average (TCP_AVG), meaning that some of these efficiency results bring a wrong efficiency score that needs to be fixed by the windows multiplicative DEA formulation. So, it is necessary to change the standard convexity from Figure 4(a) by the geometric convexity from Figure 4(b) to reach out the fair convexity of the observed production of possibilities.

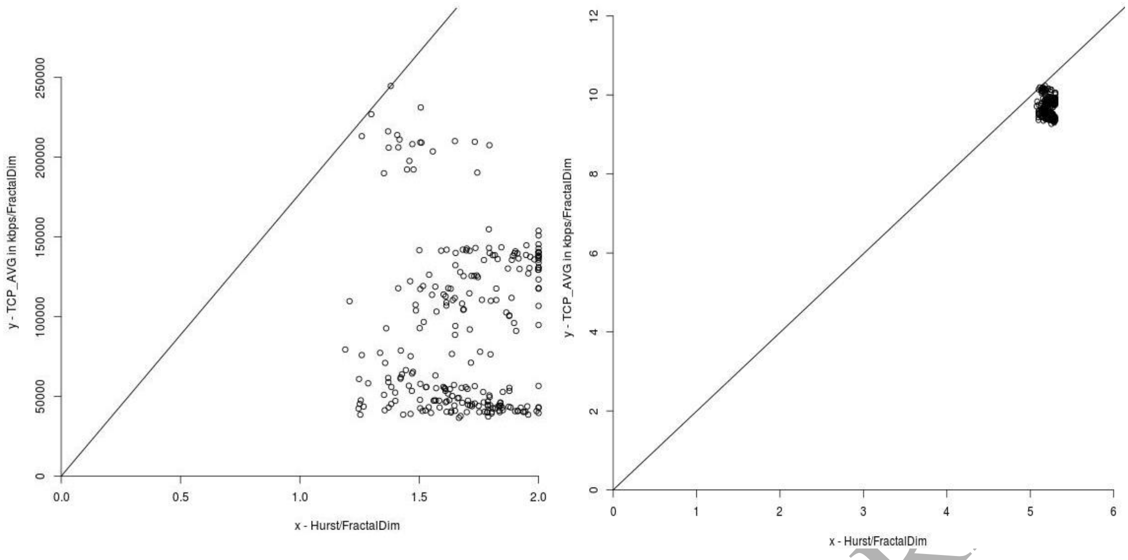
Figure 5 presents a ray unbounded arising in the efficiency frontier of both dynamic DEA models compared. Also, owing to the multiplicative axioms some DMUs in this window of evaluation (1-5) are far away of the ray, such as in Figure 5(a), instead of the Figure 5(b) where all DMUs are near to the ray due to the log-normalization. This feature is a reflection of geometric concavity about the multiplicative PPS.

Since the number of DMUs under appraisal in each window of analysis is 250, then the Figures (4) and (5) do not present the labels of DMUs of these graphics. In short, other plots from the rest of the comparison between these dynamic DEA models present these same patterns and could be found in the WMDEA's public dataset already mentioned.



(a) WDEA **(b)** WMDEA

Fig. 4 Comparison between the PPS from both dynamic DEA models in the first of the five windows under evaluation.



(a) WDEA **(b)** WMDEA

Fig. 5 Comparison among the efficiency frontiers with ray unbounded.

Considering the top five ranking in all windows of analysis from WMDEA in Table 8, the DMU50 – highlighted in bold – is elected as the best virtual network hypervisor over time. As seen in Table 5, the DMU50 is ranked in 1st place four times, and in the 2nd position twice. The fractal pattern from DMU50 evidences its superiority on delivery of virtual networks services using TCP strategies in all windows under appraisal, i.e. the DMU50 has the best end-to-end TCP´s flow control agreement among the virtual network hypervisors over time. The DMU20 is the top two in all windows, with three 2nd positions and two 4th places. Finally, the top three – DMU19 – has two 3rd positions, one 4th, and one 5th place. So, the WMDEA is always targeting a DMU with a small value of fractal dimension, that at the same time has a big TCP bandwidth performance and a huge Hurst parameter. One concludes that the WMDEA formulation makes the right choice in selecting the DMU50 as the solution for this multi-objective network optimisation problem.

Table 8: WMDEA TOP five ranking by window

Ranking	Window1	Window2	Window3	Window4	Window5	Window6	All Windows
1°	**DMU50**	DMU9	DMU10	**DMU50**	**DMU50**	**DMU50**	**DMU50**
2°	DMU9	**DMU50**	**DMU50**	DMU20	DMU20	DMU20	DMU20
3°	DMU19	DMU46	DMU9	DMU30	DMU19	DMU30	DMU19
4°	DMU20	DMU20	DMU46	DMU19	DMU49	DMU49	DMU30
5°	DMU46	DMU3	DMU39	DMU49	DMU30	DMU19	DMU49

Briefly, the highlights of the results from the WMDEA formulation brought by this research are the type-II hypervisor **VMWare** and the container-based tool **LXC,** respectively. These applications are present in all of the top three DMUs pointed out as optimal solutions to solve this fractal problem on virtual network traffic. The highlighted operating systems are **Fedora 24** and Ubuntu 16. Likewise, it is mandatory to only select the set of tools from the top DMU, and not from an isolated application (or OS, or hypervisor, or container, or others). Hence, if a datacenter administrator chooses the top DMU to execute their applications/services, then the WMDEA model will grant the most efficient virtual network services for network clients, with a smoother and higher TCP flow behaviour in a long time lag. Thus, the use of this math hypervisor allows for forecasting of the optimal virtual network hypervisor and electing the best manner to create a virtual network forwarding services using TCP. Hence, choosing the most efficient hypervisor has the stochastic guarantee of carrying a greater volume of traffic, with stability of the behaviour of the transport layer for a longer period of time, like an optimised TCP contract between the parts on virtual traffic service delivery.

Lastly, the vCC strategy from the work of Cronkite-Ratcliff et al. (2016) forgot to treat the fractal behaviour of each one of the TCP´s congestion control approaches used to activate virtual network hypervisors. So, for closing this gap, this study introduces an analytical way of stochastically adjusting the TCP´s behaviour in network hypervisors by predicting the best flow control strategy of traffic in virtual networks, which then must follow this best fractal pattern in the forwarding of more efficient TCP services from tenants over time. Another critic to vCC is that this strategy only evaluates few TCP´s congestion control just inside of an unique network, without cross the packets between virtual networks, so, decreasing the confidence of its results. Nevertheless, our study evaluates the TCP´s flow control agreements between the virtual network hypervisors which are linked to quality of TCP delivery service among client and server.

5. Conclusion and direction for future research

Network optimisation is a multi-objective linear programming algorithm used to predict efficient services towards network performance, ensuring the optimal use of system resources. So, this work proposes a continuous stepwise fractal performance evaluation framework methodology as an expert system, with the goal to reach out the virtual network optimisation from TCP behaviour over time.

Initially, the virtual networks were mounted, and after this, the fractal variables were acquired by measurements for obtaining knowledge on the time series of each DMU obeying the RFCs 2544 and 6815. It is worth to mention that prior works which proved the SS on computer networks only evaluate one network/setting in a long time span, i.e. generating just a big time series for the fractal analysis. In order to cover this gap, this work appraises the TCP traffic of many distinct virtual networks, showing that every virtual network hypervisor has a different performance and fractal behaviour from TCP over time. In addition, this research is the first to use the fractal dimension concept on virtual networks for forecasting. Thus, a set of time series is captured and analysed per DMU, according to the fractal rules.

An adjusted DEA formulation was developed, as the inference engine or math hypervisor, to effectively work with ratio variables in an evaluation over time using the new WMDEA model. The proposition of this formulation is due to the usage of variables in the form of ratios, whereas the standard DEA models need to be rectified in order to work acceptably with ratio variables. Hence, the WMDEA formulation rectifies the traditional WDEA to work correctly with variables in form of ratios in a dynamic way.

The WMDEA formulation is suitable to predict on CSPs or enterprise systems that can choose a set of best settings to deliver optimised TCP traffic on virtual networks over time. Then, a log-linear programming approach was developed to work acceptably with a geometric convexity, obeying the convexity, proportionality, ray unbounded, minimum extrapolation, and monotonicity axioms. For this reason, the use of the math hypervisor's elected by WMDEA guarantees of carrying a greater volume of traffic from TCP, with stability on the transport layer, for a longer period of time on virtual networks. In short, the usage of the optimal solution selected from WMDEA is the best end-to-end TCP contract between the client and server over time.

As highlights of the assessed software tools, the type-II hypervisor VMWare Workstation and the LXC as container tool are selected, because these technologies are in all of the top two DMUs chosen by the proposed model. Individually, the most efficient operating system was the Fedora 24, followed by Ubuntu 16. Instead, these highlighting tools must be joined as a DMU or a virtual network hypervisor, not individually. Hence, our approach can be extended to dynamically evaluate virtual networks/systems in every hardware and software environment acting as a math hypervisor. However, one limitation is that we considered only the few variables that are currently available for this analysis. Future works could first identify other variables that may affect the TCP traffic (or others' transport protocols, TCP versions, etc.) on virtual networks, and then could include them in the analysis.

This is the first publication in the area of optimising virtual networks using an expert system with modelling of WMDEA. Hence, interested readers could follow this research in several directions. For example, research could pursue a) scaling-up the scope of the fractal knowledge related to computation, cloud workload, other network protocols, and so on, aiming for the development of a dynamic-network multiplicative DEA model for delivering of more stable services according to an interrelated analysis of a set of cloud infrastructure layers; b) making a comparison between the WMDEA formulation using these same fractal variables against the main dynamic MCDM techniques mixed with artificial intelligence approaches, such as artificial neural networks, evolutionary algorithms, fuzzy logics, and others; c) applying this expert system to assess software-defined network orchestrators, cloud operating systems, or

TCP´s congestion control strategies (or even distinct versions of the same protocol); and d) conceiving an optimised cloud network service orchestration as a service and extending this work to evaluate real traffic, both in the same cloud and in clouds that exchange traffic between continents.

Finally, the empirical results showed that the WMDEA formulation can be used to evaluate the virtual network problem presented. Also, this dynamic multiplicative DEA model may be applied in every problem where all the decision variables are in the form of a ratio.

Acknowledgements

The authors would like to thank the editor of Expert Systems with Applications – *Professor Binshan Lin* – and two anonymous reviewers for their insightful comments. As a result, this paper has been improved substantially. We also would like to thank FACEPE, IFPB, UFPE, and Aston University for providing the opportunity to work on this research.

Author contributions:

Francisco Daladier Marques Júnior: Conceptualization, Methodology, Design and Implementation, Formal Analysis, Validation, Software, Writing-Reviewing and Editing.

Ali Emrouznejad: Supervision, Methodology, Formal Analysis, Validation, Software, Writing-Reviewing and Editing.

Kelvin Lopes Dias: Supervision, Validation, Design , Formal Analysis, Writing-Reviewing and Editing.

Paulo Roberto Freire Cunha: Supervision, Validation, Writing-Reviewing and Editing.

Jorge Luiz de Castro e Silva: Design, Methodology, Validation, Writing-Reviewing and Editing.

References

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. https://doi.org/10.1287/mnsc.30.9.1078

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. https://doi.org/https://doi.org/10.1016/S0377-2217(03)00174-7

Banker, R. D., & Maindiratta, A. (1986). Piecewise Loglinear Estimation of Efficient Production Surfaces. *Manage. Sci.*, *32*(1), 126–135. https://doi.org/10.1287/mnsc.32.1.126

Banker, R. D., & Morey, R. C. (1986). Efficiency Analysis for Exogenously Fixed Inputs and Outputs. *Oper. Res.*, *34*(4), 513–521. https://doi.org/10.1287/opre.34.4.513

Bez, N., & Bertrand, S. (2011). The duality of fractals: Roughness and self-similarity. *Theoretical Ecology*, *4*(3), 371–383. https://doi.org/10.1007/s12080-010-0084-y

Campbell, P., & Abhyankar, S. (1978). Fractals, form, chance and dimension. *The Mathematical Intelligencer*, *1*(1), 35–37. https://doi.org/10.1007/BF03023043

Charnes, A., Clark, C. T., Cooper, W. W., & Golany, B. (1984). A developmental study of data envelopment analysis in measuring the efficiency of maintenance units in the U.S. air forces. *Annals of Operations Research*, *2*(1), 95–112. https://doi.org/10.1007/BF01874734

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, *2*(6), 429–444. https://doi.org/10.1016/0377-2217(78)90138-8

Chowdhury, N. M. M. K., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, *54*(5), 862–876. https://doi.org/10.1016/j.comnet.2009.10.017

Cooper, W. W., Seiford, L. M., & Tone, K. (2007). *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software: Second edition*. *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software: Second Edition*. https://doi.org/10.1007/978-0-387-45283-8

Cronkite-Ratcliff, B., Bergman, A., Vargaftik, S., Ravi, M., McKeown, N., Abraham, I., & Keslassy, I. (2016). Virtualized Congestion Control. In *Proceedings of the 2016 ACM SIGCOMM Conference* (pp. 230–243). New York, NY, USA: ACM.

https://doi.org/10.1145/2934872.2934889

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. https://doi.org/10.1109/90.650143

Dauphiné, A. (2013). A Fractal World. In *Fractal Geography* (pp. 1–19). John Wiley & Sons, Inc. https://doi.org/10.1002/9781118603178.ch1

Emrouznejad, A., & Amin, G. R. (2009). DEA models for ratio data: Convexity consideration. *Applied Mathematical Modelling*, *33*(1), 486–498. https://doi.org/https://doi.org/10.1016/j.apm.2007.11.018

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. Retrieved from https://econpapers.repec.org/RePEc:ids:ijfsmg:v:4:y:2010:i:2:p:114-126

Emrouznejad, A., Cabanda, E., & Gholami, R. (2010). An alternative measure of the ICT-Opportunity Index. *Information & Management*, *47*(4), 246–254. https://doi.org/https://doi.org/10.1016/j.im.2010.04.002

Emrouznejad, A., & Witte, K. De. (2010). COOPER-framework: A unified process for non-parametric projects. *European Journal of Operational Research*, *207*(3), 1573–1586. https://doi.org/https://doi.org/10.1016/j.ejor.2010.07.025

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*. https://doi.org/10.1016/j.seps.2017.01.008

Färe, R., & Knox Lovell, C. A. (1978). Measuring the technical efficiency of production. *Journal of Economic Theory*, *19*(1), 150–162. https://doi.org/https://doi.org/10.1016/0022-0531(78)90060-1

Florindo, J. B., & Bruno, O. M. (2013). Texture analysis by multi-resolution fractal descriptors. *Expert Systems with Applications*, *40*(10), 4022–4028. https://doi.org/https://doi.org/10.1016/j.eswa.2013.01.007

Gilbert, E. N. (1960). Capacity of a burst-noise channel. *The Bell System Technical Journal*, *39*(5), 1253–1265. https://doi.org/10.1002/j.1538-7305.1960.tb03959.x

Gneiting, T., Sevčíková, 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. https://doi.org/10.1214/11-STS370

H. E. Hurst C.M.G. M.A., D. S. F. I. P. (1956). THE PROBLEM OF LONG-TERM STORAGE IN RESERVOIRS. *International Association of Scientific Hydrology. Bulletin*, *1*(3), 13–27. https://doi.org/10.1080/02626665609493644

Halkos, G. E., & Polemis, M. L. (2018). The impact of economic growth on environmental efficiency of the electricity sector: A hybrid window DEA methodology for the USA. *Journal of Environmental Management*, *211*, 334–346. https://doi.org/https://doi.org/10.1016/j.jenvman.2018.01.067

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. Retrieved from http://www.jstor.org/stable/2245054

Hill, B. M. (1975). A Simple General Approach to Inference About the Tail of a Distribution. *Ann. Statist.*, *3*(5), 1163–1174. https://doi.org/10.1214/aos/1176343247

Iqbal, M., Naeem, M., Anpalagan, A., Qadri, N. N., & Imran, M. (2016). Multi-objective optimization in sensor networks: Optimization classification, applications and solution approaches. *Computer Networks*, *99*, 134–161. https://doi.org/https://doi.org/10.1016/j.comnet.2016.01.015

Jatoth, C., Gangadharan, G. R., & Fiore, U. (2016). Evaluating the efficiency of cloud services using modified data envelopment analysis and modified super-efficiency data envelopment analysis. *Soft Computing*. https://doi.org/10.1007/s00500-016-2267-y

Kantelhardt, J. W. (2008). Fractal and Multifractal Time Series. *Encyclopedia of Complexity and Systems Science*, 59. https://doi.org/10.1007/SpringerReference_60393

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. https://doi.org/10.1109/90.282603

Li, S., Jahanshahloo, G. R., & Khodabakhshi, M. (2007). A super-efficiency model for ranking efficient units in data envelopment analysis. *Applied Mathematics and Computation*, *184*(2), 638–648. https://doi.org/10.1016/j.amc.2006.06.063

Lloyd, C. D., Berberoglu, S., Curran, P. J., & Atkinson, P. M. (2004). A comparison of texture measures for the per-field classification of Mediterranean land cover. *International Journal of Remote Sensing*, *25*(19), 3943–3965. https://doi.org/10.1080/0143116042000192321

López-Ortega, O., & López-Popa, S. I. (2012). Fractals, fuzzy logic and expert systems to assist in the construction of musical pieces. *Expert Systems with Applications*, *39*(15), 11911–11923. https://doi.org/https://doi.org/10.1016/j.eswa.2012.02.089

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. https://doi.org/10.1109/TCOM.1965.1089090

Mandelbrot, B. (1967). How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science*, *156*(3775), 636–638. https://doi.org/10.1126/science.156.3775.636

Mandelbrot, B. B. (1975). Stochastic Models for the Earth's Relief, the Shape and the Fractal Dimension of the Coastlines, and the Number-Area Rule for Islands. *Proceedings of the National Academy of Sciences of the United States of America*, *72*(10), 3825–3828. Retrieved from http://www.jstor.org/stable/65184

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature*. 1997. Retrieved from https://books.google.com.br/books?id=SWcPAQAAMAAJ

Mandelbrot, B. B., & Taleb, N. N. (2012). mild vs wild randomness focusing on those risks that matter. In *the Known, the Unknown and the Unknowable in Financial Risk Management : Measurement and Theory Advancing Practice*. https://doi.org/10.1017/CBO9781107415324.004

Mandelbrot, B. B., & Van Ness, J. W. (1968). Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, *10*(4), 422–437. Retrieved from http://www.jstor.org/stable/2027184

Mandelbrot, B. B., & Wallis, J. R. (1969). Computer Experiments with Fractional Gaussian Noises: Part 2, Rescaled Ranges and Spectra. *Water Resources Research*, *5*(1), 242–259. https://doi.org/10.1029/WR005i001p00242

Martín de Diego, I., Siordia, O. S., Fernández-Isabel, A., Conde, C., & Cabello, E. (2019). Subjective data arrangement using clustering techniques for training expert systems. *Expert Systems with Applications*, *115*, 1–15. https://doi.org/https://doi.org/10.1016/j.eswa.2018.07.058

Ni, L.-P., Ni, Z.-W., & Gao, Y.-Z. (2011). Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications*, *38*(5), 5569–5576. https://doi.org/https://doi.org/10.1016/j.eswa.2010.10.079

Olesen, O. B., Petersen, N. C., & Podinovski, V. V. (2017). Efficiency measures and computational approaches for data envelopment analysis models with ratio inputs and outputs. *European Journal of Operational Research*, *261*(2), 640–655. https://doi.org/https://doi.org/10.1016/j.ejor.2017.02.021

Oliveira, C., Kim, J. B., & Suda, T. (2003). Long-range dependence in IEEE 802.11b wireless LAN traffic: an empirical study. In *2002 14th International Conference on Ion Implantation Technology Proceedings (IEEE Cat. No.02EX505)* (pp. 17–23). https://doi.org/10.1109/CCW.2003.1240784

Paxson, V., & Floyd, S. (1995). Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, *3*(3), 226–244. https://doi.org/10.1109/90.392383

Przystalski, K., & Ogorzałek, M. J. (2017). Multispectral skin patterns analysis using fractal methods. *Expert Systems with Applications*, *88*, 318–326. https://doi.org/https://doi.org/10.1016/j.eswa.2017.07.011

Pulina, M., Detotto, C., & Paba, A. (2010). An investigation into the relationship between size and efficiency of the Italian hospitality sector: A window DEA approach. *European Journal of Operational Research*, *204*(3), 613–620. https://doi.org/https://doi.org/10.1016/j.ejor.2009.11.006

Raja, S., & Ramaiah, S. (2016). CCDEA: Consumer and cloud - DEA based trust assessment model for the adoption of cloud services. *Cybernetics and Information Technologies*, *16*(3), 52–69. https://doi.org/10.1515/cait-2016-0034

Sahoo, J., Mohapatra, S., & Lath, R. (2010). Virtualization: A survey on concepts, taxonomy and associated security issues. In *2nd International Conference on Computer and Network Technology, ICCNT 2010* (pp. 222–226). https://doi.org/10.1109/ICCNT.2010.49

Seiford, L., Charnes, A., W. Cooper, W., & A. Stutz, J. (1982). A Multiplicative Model for Efficiency Analysis. *Socio-Economic Planning Sciences*, *16*, 223–224.

Shang, P., Wan, M., & Kama, S. (2007). Fractal nature of highway traffic data. *Computers & Mathematics with Applications*, *54*(1), 107–116. https://doi.org/https://doi.org/10.1016/j.camwa.2006.07.017

Soleimani-damaneh, M. (2009a). Establishing the existence of a distance-based upper bound for a fuzzy DEA model using duality. *Chaos, Solitons & Fractals*, *41*(1), 485–490. https://doi.org/https://doi.org/10.1016/j.chaos.2008.02.012

Soleimani-damaneh, M. (2009b). Maximal flow in possibilistic networks. *Chaos, Solitons & Fractals*, *40*(1), 370–375. https://doi.org/https://doi.org/10.1016/j.chaos.2007.07.071

Stein, A., Shi, W., & Bijker, W. (2008). *Quality Aspects in Spatial Data Mining* (1st ed.). Boca Raton, FL, USA: CRC Press, Inc.

Truong, D. (2014). Cloud-Based Solutions for Supply Chain Management : a Post-Adoption Study. *SBBS Annual Conference*, *21*(1), 697–708.

Wang, D., Ren, C., Govindan, S., Sivasubramaniam, A., Urgaonkar, B., Kansal, A., & Vaid, K. (2013). ACE: Abstracting, characterizing and exploiting datacenter power demands. In *2013 IEEE International Symposium on Workload Characterization (IISWC)* (pp. 44–55). https://doi.org/10.1109/IISWC.2013.6704669

Wang, K., Yu, S., & Zhang, W. (2013). China's regional energy and environmental efficiency: A DEA window analysis based dynamic evaluation. *Mathematical and Computer Modelling*, *58*(5), 1117–1127. https://doi.org/https://doi.org/10.1016/j.mcm.2011.11.067

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. https://doi.org/10.1016/S0378-4371(02)00961-5

Yang, H.-H., & Chang, C.-Y. (2009). Using DEA window analysis to measure efficiencies of Taiwan's integrated telecommunication firms. *Telecommunications Policy*, *33*(1), 98–108. https://doi.org/https://doi.org/10.1016/j.telpol.2008.11.001