INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2016; **00**:1–19
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nem

Online Traffic Prediction in the Cloud

Bruno L. Dalmazo*† [1], João P. Vilela[1], Marilia Curado[1]

[1] *CISUC, Department of Informatics Engineering*
University of Coimbra, Coimbra, Portugal

SUMMARY

Network traffic prediction is a fundamental tool to harness several management tasks, such as monitoring and managing network traffic. Online traffic prediction is usually performed based on large sets of historical data used in training algorithms, for example, to determine the size of static windows to bound the amount of traffic under consideration. However, using large sets of historical data may not be suitable to highly volatile environments, such as cloud computing, where the coupling between time series observations decreases rapidly with time. To fill this gap, this work presents a dynamic window size algorithm for traffic prediction, that contains a methodology to optimize a threshold parameter α that affects both the prediction and computational cost of our scheme. The α parameter is the control mechanism that defines which is the minimum data traffic variability needed to the dynamic window size changes. Thus, with the optimization of this parameter, the number of operations of the dynamic window size algorithm decreases significantly. We evaluate the α estimation methodology against several prediction models by assessing the Normalized Mean Square Error and Mean Absolute Percent Error of predicted values over observed values from two real cloud computing data sets, collected by monitoring the utilization of Dropbox, and a data center data set including traffic from several common cloud computing services.

Received ...

KEY WORDS: Cloud computing; network traffic prediction; short-range dependence; sliding window algorithm.

1. INTRODUCTION

The uprise of next generation networking paradigms, such as the internet of things and cloud computing, has entirely changed the way that networks are conceived and managed. By deploying a cloud computing model, organizations have many advantages such as on-demand computing services and reduced maintenance costs [1]. However, along with these benefits, cloud computing brought a multitude of challenges into the focus of worldwide research [2].

More than ever, many services and products rely on cloud-based systems and networks. Neglecting the management of these network assets may cause irreparable economic harm to

*Correspondence to: Bruno L. Dalmazo, Department of Informatics Engineering, Pólo II - Pinhal de Marrocos, Coimbra, University of Coimbra, Portugal
†E-mail: dalmazo@dei.uc.pt

businesses and their customers alike [3]. Network administrators of these cloud-supporting networks have to monitor and analyze their networks in order to collect relevant network traffic information that may be used to support decision-making. In this context, network managers need tools suited to deal with the high network traffic volume common in cloud environments.

Effective monitoring of computer networks must be constantly performed to assist in the detection and identification of network problems as they happen or even beforehand [4]. If, on one hand, network traffic monitors are able to store statistics about the network connectivity and availability of applications in order to build a baseline that describes its proper behavior, on the other hand, cloud providers generate huge amounts of information, such that storing data from the entire network infrastructure may become prohibitively expensive. Management systems thus require efficient techniques to reduce the required service and operating resources [5].

Network traffic prediction can make use of statistics accumulated over time for making inferences about the future behavior of network traffic, therefore enabling the detection of suspicious patterns of network traffic. Some metrics usually considered are the throughput, response time, jitter and lost data. For this information to be useful in planning strategies and responding to problems as they happen [6], online traffic prediction is called upon.

However, predicting network traffic is becoming a more complex task, specially with the surge in traffic that is due to the permanent connectivity of individuals and machines to the Internet [7]. This challenge is even greater in cloud computing because its traffic may suffer sudden changes [8, 9], and the elastic and scalable nature of cloud environments may be easily confused with traffic anomalies, hampering its forecast [10]. In addition, traditional tools for predicting data traffic usually take into account large historical data, therefore being classified as Long-range Dependence (LRD) approaches. However, LRD-based techniques are not the most suitable for online traffic prediction of cloud computing systems because the network baseline does not have the same periodic behavior as in traditional networks [11]. For instance, predicting approaches on the basis of large historical dependency are supported by features such as seasonality, and thus present similar behavior in regular time intervals. The main disadvantage is that this type of prediction approach requires a time window with a large number of values, and this adds an extra workload to compute the estimation.

Solving the problem of processing large amounts of data for traffic prediction represents an important achievement in cloud computing to avoid unnecessary overhead and minimize the operation costs. Moreover, dynamically reducing the amount of information to process is also relevant to other applications, such as traffic shaping for improved Quality of Service (QoS) [12], forecasting network traffic to detect anomalies and spot problems before they occur [11] or to conceive more accurate simulation models [13].

In order to address these issues, this paper presents a dynamic sliding window algorithm that defines the amount of traffic under consideration for traffic prediction according to traffic variability. Furthermore, a new methodology which exploits widely the parameter α from the dynamic sliding window algorithm, is proposed. From the optimization of this parameter, it is possible to maximize the prediction accuracy in comparison with the older version of the algorithm that uses a static value for defining the boundary to sliding window changes the size. In addition, in this work, we have broadened and detailed the state of the art regarding Short-range Dependence (SRD) prediction approaches and online traffic predictors. Moreover, to evaluate the feasibility of our proposal in more

Prepared using nemauth.cls
DOI: 10.1002/nem

general terms we have used real traces from Dropbox as well as from a real data center containing traffic from several common cloud computing services.

The remainder of the paper is organized as follows. Section 2 covers some of the most prominent related work. Section 3 describes the proposed solution and the methodology used for this paper, whilst Section 4 presents the evaluation and discusses the results. Section 5 concludes with some final remarks and prospective directions for future research.

2. RELATED WORK

Network traffic prediction has received a great deal of attention from the scientific community as a means to facilitate monitoring and managing computer networks [14]. In this field, most research efforts are focused on classical methods strongly based on historical data such as time series and neural networks. In this study, we consider previous works that (1) have short dependence on historical data, and (2) may be performed online.

2.1. Short-range dependency traffic prediction

Maria Papadopouli *et al.* [13] evaluate a set of forecast algorithms in order to characterize the traffic load in an IEEE802.11 infrastructure. Their work describes the Simple Moving Average (SMA) as the unweighted mean of the previous data points in the time series. In addition, SMA is less demanding than more complex predictors, such as Autoregressive Integrated Moving Average (ARIMA), that require a large amount of historical data. They emphasize some advantages of SMA, such as its simplicity, low complexity and ease of application.

Short-range dependency is also exploited in the field of the management of power systems. James W. Taylor considers five Weighted Moving Averages for forecasting load up to one day ahead [15]. These models include several exponential smoothing formulations, as well as methods using discount weighted regression, cubic splines, and singular value decomposition (SVD). In this paper, the author improves the forecasting results by changing the granularity of time period in a SVD.

Aiping Li *et al.* [16] study anomaly detection methods for high-speed network traffic. The purpose of this work is to come up with a sensible mechanism for detecting significant changes in massive data streams with a large number of flows. Through a model based on a Weighted Moving Average (WMA), the algorithm estimates the value of the next interval, being able to detect distributed denial-of-service (DDoS) and scan attacks. For that, all traffic that does not match the reference model is considered an anomaly.

In [17], Frank Klinker describes mathematical tools to identify and predict market trends by the relationship between the original and the predicted data. In particular, it shows that the Exponential Moving Average (EMA) can be used for efficient forecast of network traffic with short historical data. EMA is also used in the Piorno *et al.* [18] work. They compare several prediction algorithms to predict Sun's cycles and the changing weather conditions to improve solar panel for exploiting the extra energy available.

In a previous work [19], we have presented a systematic approach to estimate network traffic resorting to a statistical method based on a Poisson process (Poisson Moving Average - PMA) with window of static size to weight past observations. After that, we have proposed a dynamic

Prepared using nemauth.cls
DOI: 10.1002/nem

approach based on maximum variance inside window for improving accuracy among short-range dependency approaches [20].

These prediction methods are considered SRD approaches because they resort to windows of fixed/static but small size. The window size is, however, determined from metrics of the overall data, such as the global average, therefore limiting their applicability to online prediction in scenarios where too much data must be processed. In other words, the dynamic nature of cloud computing turns this kind of operation expensive. For example, once a huge amount of data is generated, it is not feasible to calculate the window size constantly from parameters of the whole data set.

2.2. *Online traffic prediction*

Yuehui Chen *et al.* [21] use genetic programming to build a Flexible Neural Tree (FNT) for online network traffic prediction. This approach was used for a better understanding of the main features of the traffic data. Moreover, the proposed method is able to forecast small-time scale traffic measurements and can reproduce the statistical features of real traffic measurements. However, to achieve proper results, it requires initial input that is dependent on the characteristics of data under evaluation.

Zare Moayedi and Masnadi-Shirazi [22] propose a network traffic prediction and anomaly detection model based on Autoregressive Integrated Moving Average (ARIMA). In this paper, they decompose the data flow in order to isolate anomalies from normal traffic variation. The authors then try to predict anomalies independently from normal traffic. Their work was evaluated with synthetic data and depends on large historic data for forecasting.

Wen-Kuang Kuo and Kuo-Wei Wu [23] propose a traffic predictor designed to provide online prediction with the goal of guaranteeing QoS in real-time live video transmission. The predictor, based on variable step size least mean square algorithm, achieves high channel utilization and guarantees the QoS requirements for real-time video. However, it obtains information only from the last simple scene, restricting the ability to forecast abrupt changes, common in cloud environments.

Rajnish Yadav and Manoj Balakrishnan [24] present a comparative performance evaluation between the Autoregressive Integrated Moving Average (ARIMA) and an Adaptive Neuro Fuzzy Inference System (ANFIS). The goal of this work is to model the behavior of wireless network traffic. In the scenario evaluated, ANFIS shows the best results, but with high computational costs.

Although these works allow online traffic prediction, they are unsuitable to the cloud environment due to their dependency on large historical data for training the algorithms, and thus increases the cost of the prediction operation. In this work we analyse a dynamic sliding window mechanism based on SRD for traffic prediction with a new approach to estimate the minimum variance necessary for changing the window size. This solution facilitates online traffic prediction by reducing the amount of data necessary to process when compared to LRD-based schemes. Moreover, windows sizes are determined dynamically, without requiring statistics of the overall data – only local data from the current and previous window is needed.

DOI: 10.1002/nem

3. DYNAMIC WINDOW SIZE MECHANISM

The dynamic window size is a mechanism to limit the amount of information that is used for traffic prediction, therefore making it suitable for online prediction in a cloud computing context.

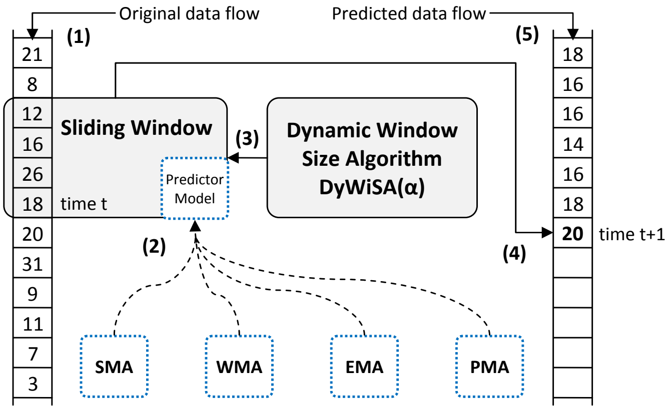


Figure 1. Elements of the proposed solution and iterations

Figure 1 illustrates the main conceptual components and their interactions. Real-time cloud traffic data (step 1) is gathered and analyzed in the *Sliding Window* component in order to estimate network traffic from short historical data. This cloud data traffic is processed according to a particular predictor model, as illustrated in step 2. Possible candidates for the predictor model (described in Section 2.1) include Simple Moving Average, Weighted Moving Average, Exponential Moving Average and Poisson Moving Average. The *Dynamic Window Size Algorithm* component is responsible for the definition of the window size that serves as input to the *Sliding Window* component (step 3). The next value of cloud data traffic is predicted (step 4) according to the chosen predictor model, therefore resulting in a sequence of predicted values for the cloud data traffic (step 5).

By employing a window of dynamic but limited size (SRD characteristic), we minimize the workload by reducing the amount of data that must be processed by the predictor model. We now describe each component in more detail.

3.1. Sliding Window

In order to reduce the complexity of predicting network traffic, we consider time-bounded past information by means of a sliding window of size defined by the *Dynamic Window Size Algorithm* (Algorithm 1 – DyWiSA). A window of the given size is used to weight past observations of data traffic according to the distribution employed by the predictor model.

The example illustrated in Figure 1 shows a sliding window with size four. Each value of the original data flow is weighted with a portion of the statistical distribution of the corresponding predictor model [19]. For instance, the *DyWiSA* is familiar with the statistical behavior of the predictor models. In this specific case, for each time slot, the *Sliding Window* considers the fourth part of the distribution to ponder the number of network packets.

It is worth pointing out that the Simple Moving Average, Weighted Moving Average and the Poisson Moving Average use a discrete function to weight the data. However, for the Exponential

Prepared using nemauth.cls
DOI: 10.1002/nem

Moving Average, the DyWiSA divides the function into a finite number of discrete elements before using, namely, it discretizes of the exponential function.

Algorithm 1 Dynamic window size

Input: Average of the current sliding window, $newAvg$
Average of the previous sliding window, $oldAvg$
Current sliding window, $sWindow$
The α level, α
Output: Next window size, $wSize$

Start
  procedure DyWiSA($newAvg$, $oldAvg$, α, $sWindow$)
    $var\ wSize = sWindow.size()$
    $var\ ratio = module(newAvg, oldAvg)$
    if $(ratio > (1 + \alpha))$ **then**
      $var\ volume = \frac{\sigma^2_{max}}{\sigma^2}$
      if $(newAvg > oldAvg)$ **then**
        $wSize = wSize + volume$
      else
        $wSize = wSize - volume$
      end if
    end if
    return $wSize$
  end procedure
End

Thus, at time t, the sliding window has a set of four values $\{12, 16, 26, 18\}$. In the next turn, at time $t + 1$, the next value to enter inside the window will be 20, and when this occurs, the oldest value (12) leaves the sliding window. This process will be repeated as long as there is a data flow from the network.

3.2. Dynamic Window Size Algorithm - DyWiSA

Traffic predictors usually operate over all of previous data [21] or resort to windows of finite but fixed size. However, the network traffic in the cloud computing environment may suffer sudden changes due to the large amount of requests and dynamic demands without prior notification [8]. This led us to consider a sliding window approach as a forgetting process that limits the amount of data to be processed. If the sliding window is large, the predictor will be able to smooth traffic anomalies. This situation happens when the time series are increasing (or decreasing) the data flow quickly. If the sliding window is small, the model will be more sensitive to changes, however it will generate lower workload due to the fewer number of data packets to process. This happens when the data flow presents a stable behavior.

To take these traffic behavior changes into account, we consider the variance (σ^2) between the previous and current values inside of the sliding window. Algorithm 1 describes the operation of the Dynamic Window Size Algorithm. It resorts to a sliding window of variable size, with size changes happening only when the difference between the average of current and previous window exceeds a threshold α.

The algorithm receives as input the average of the current sliding window, the average from the previous sliding window and the current sliding window. It compares the average of the old sliding

DOI: 10.1002/nem

window with the average of the current sliding window. In order to avoid unnecessary algorithm overhead, we consider a threshold α for changes to the sliding window. This threshold corresponds to a boundary value for the population parameter for which the difference between the current value and the mean of the last window is not statistically significant at the α level. Its estimation and evaluation are reported in Section 3.3 and Section 4.3, respectively.

Let *ratio* be a value which measures average changes between the current window and last window. If the difference between *newAvg* and *oldAvg* is higher than the threshold $(1 + \alpha)$, *i.e.* statistically significant, the window size is increased (or decreased) by *volume*. In order to quantify the maximum variance of a sliding window and, consequently, know the variation of the window size, a measurement to express the largest variance possible inside of a subset of the entire population is needed. We consider the theoretical maximum variance (σ^2_{max}) to be the variance of the extreme values of a sliding window. For this, we use the ratio between the σ^2_{max} and the σ^2 inside a sliding window. This whole process is represented by the variable *volume* at line 6 of Algorithm 1.

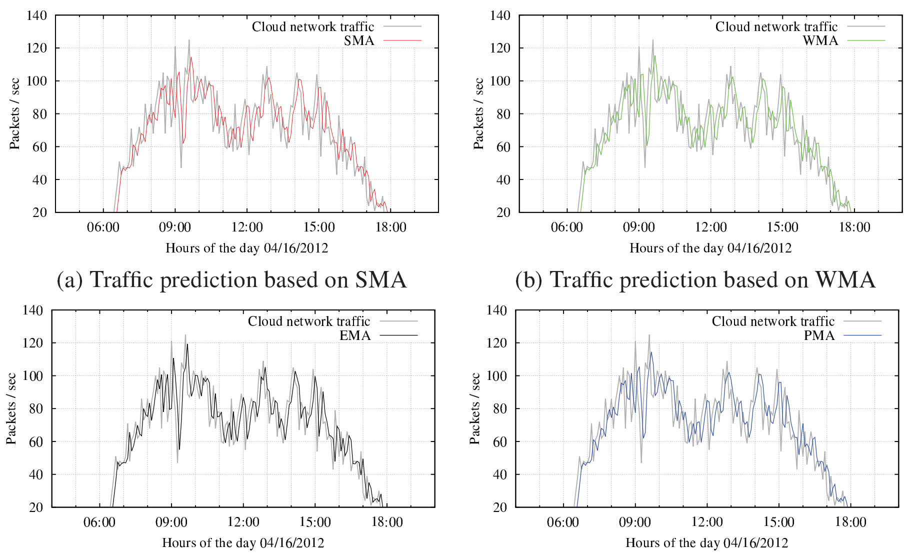
Proposition 1. The theoretical maximum variance of a given set of data can be estimated from the product of the difference of its extreme values, y_a (lowest value), y_b (highest value), and the average, as follows:

$$\sigma^2_{max} = (m - y_a)(y_b - m) \tag{1}$$

Proof

See Appendix A. ■

Figure 2 illustrates the performance of different approaches with the traffic data set containing information from Dropbox monitoring. In order to provide a better viewing of the results, we only show forecasts for a limited period. However, the observable match between real values and predictions held for remaining time periods.



(a) Traffic prediction based on SMA (b) Traffic prediction based on WMA

(c) Traffic prediction based on EMA (d) Traffic prediction based on PMA

Figure 2. Sample of cloud network traffic prediction

DOI: 10.1002/nem

Finally, the algorithm returns the window size to be used by the *Sliding Window* component. This Dynamic Window Size Algorithm is at the core of online traffic prediction by dynamically adapting the window size resorting only to local data from current and previous sliding windows, instead of global traffic data.

3.3. α Parameter Estimation

For this work, we propose a new approach to estimate the best α for each forecast. The goal of this methodology is twofold: to select the α parameter that provides the highest possible accuracy for predicting network traffic; and to minimize the computational costs as much as possible. We consider the α parameter a control mechanism which represents the minimum variation of data traffic needed for the window size to change. A small α will lead to frequent window size changes and higher computationally complexity, while a higher α leads to less frequent changes with corresponding lower computational costs. For instance, when the α is equal to 0.1 it means that the average of the current sliding window must be, at least, 10% greater or smaller than the last sliding window for the algorithm to require a new window size calculation.

Seeking a methodology that determines a value of α that provides good accuracy results without compromising the need for online traffic prediction (*i.e.* little dependence on historical data), we consider only an initial set of windows to determine an optimal α value. As observed in other works [25, 26], the resemblance between the first two sliding windows and the entire dataset suggests that the network traffic data exhibits the property of self-similarity. Taking advantage of this concept, the algorithm will set the best α found in the first two sliding windows to predict the entire dataset. Section 4.3 presents the evaluation of this process.

4. EVALUATION AND DISCUSSION

In this section, we present the data setup and the metrics used to assess the Dynamic Window Size Algorithm and the best parameter α. Furthermore, we evaluate and compare the performance of the static sliding window [19] and the DyWiSA with the α estimation methodology applied to all SRD traffic prediction mechanisms presented in Section 2.

4.1. Metrics

The effectiveness of the prediction is measured through the Normalized Mean Square Error (NMSE) [27] and Mean Absolute Percent Error (MAPE) [28]. NMSE is defined as:

$$NMSE = \frac{1}{\sigma^2}\frac{1}{N}\sum_{t=1}^{N}\left(X_t - \hat{X}_t\right)^2 \tag{2}$$

where σ^2 is the variance of the time series over the prediction duration, X_t is the observed value of the time series at time t, $\hat{X}_t$ is the predicted value expected from X_t, and N is the total number of predicted values. This metric is widely utilized to assess prediction accuracy. Its results are compared with a trivial predictor, which statistically predicts the mean of the actual time series,

DOI: 10.1002/nem

in which case NMSE = 1. If NMSE = 0, this means that it is a perfect predictor, whereas NMSE $>$ 1 means that the predictor performance is worse than that of a trivial predictor [27].

MAPE measures expressed errors as a percentage of the actual data over the prediction data. It is calculated as the average of the unsigned percentage error, and it is defined by the formula:

$$MAPE = \begin{cases} \left(\frac{1}{N}\sum_{t=1}^{N}\frac{|X_t-\hat{X}_t|}{|X_t|}\right)\times 100 & if(X_t>0) \\ \left(\frac{1}{N}\sum_{t=1}^{N}\frac{|X_t-\hat{X}_t|}{|\overline{X}|}\right)\times 100 & otherwise \end{cases} \tag{3}$$

where, X_t is the observed value, $\hat{X}_t$ is the predicted value and N represents the total number of values in the time series. If the denominator is zero then the actual value X_t is replaced by the average of time series, $\overline{X}$. When having a perfect fit, MAPE is zero.

4.2. *Datasets Setup*

In this article, we used two scenarios to evaluate our proposal. First, we used two datasets from Dropbox monitoring [29]. After that, we used data from a data center including traffic from several common cloud computing services [30].

4.2.1. Dropbox Datasets: In this case study, we used two datasets from Dropbox monitoring as described in the [29], they are: Home 1 and Campus 2. Home 1 dataset consists of ADSL and Fiber to the Home customers of a nation-wide ISP, but they might use WiFi routers at home to share the connection. Campus 2 was instead collected in academic environments such as wired workstations in research and administrative offices as well as campus-wide wireless access points.

All the measurements and data presented here were collected from March 24, 2012 to May 5, 2012. The evaluated time series data is focused on Dropbox utilization, which is one of the most widely-used cloud storage system nowadays [29]. The original Dropbox dataset encompasses more than 100 metrics about the network traffic. However, for this study, we consider the total number of packets observed from the client (server) to the server (client). The time series were divided in intervals of five minutes each, and the analysis mechanism was performed by applying a sliding window weighted with the four SRD traffic prediction mechanisms.

4.2.2. Data Center Dataset: For a better characterization of cloud computing environment, we use other dataset that provides data from monitoring a variety of services common in cloud computing [30]. In this work, the authors describe several services present in the dataset such as webmail servers, web portals, instant messaging, web services and multicast video streams. In addition, the dataset presents data from a two-layer topology and introduced server virtualization techniques in order to reduce heating and power consumption.

The data was collected in an academic environment and the dataset consists of more than five years of monitoring. However, the authors provide just a snippet of the total data used in the original paper [30], of around 10 days. The granularity of the time series generated is sixty seconds each time slot and called by *Data Center* throughout this study. The Data Center keeps data from around 1000

DOI: 10.1002/nem

servers located in western and midwestern U.S. The analysis mechanism was performed over the data in the same way as described in the Dropbox Case Study.

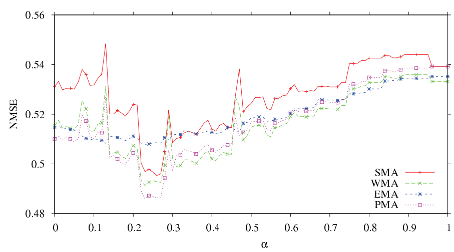
4.3. α Parameter Evaluation

In this section we analyze the impact of α on the prediction accuracy and present a methodology for selecting its value for two scenarios: *Dropbox* and *Data Center* datasets. Figure 3(a) depicts accuracy results (NMSE) of the two initial windows for a range of α values between 0.01 and 1.0. This shows that the optimal α for the two initial windows is 0.23 (smaller NMSE), which incidentally is also the optimal α value when considering the whole dataset, as shown in Figure 3(b). The same happens when considering the MAPE accuracy metric, as shown in Figures 3(c) and 3(d), respectively for the two initial windows and the overall dataset.

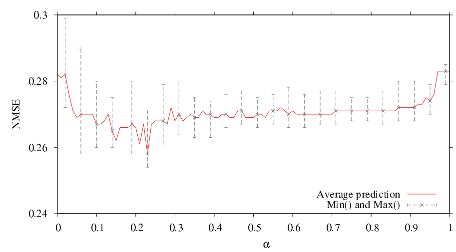
Figure 4 presents similar results for the Data Center dataset. For this case, our methodology leads to an optimal α value of 0.15 for both metrics as well. While this may not provide an overall optimal value, we figured out that this methodology provides, in general, a good approximation.

Table I shows the output results when we run the algorithm with different α values in a sample time series. With an α of 0.23, the Dynamic Window Size Algorithm changes the window size 886 times (just 7.13% of the total) with no significant changes in the overall average and standard deviation of the data predicted. As α becomes smaller, the overhead becomes higher and vice-versa.

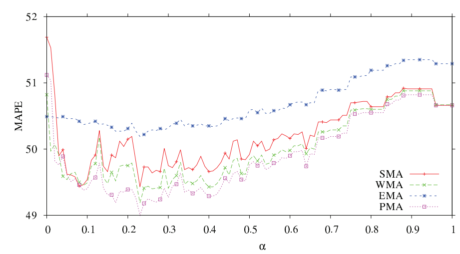
From a computational point of view, the α estimation yields negligible complexity because this process requires only an initial set of two sliding windows. In comparison with the first version of the Dynamic Window Size Algorithm (with fixed α), the prediction time has been decreased by almost half after the α optimization approach. For instance, in the previous version of the DyWiSA, the Poisson Moving Average (the predictor model with the best prediction accuracy) spends 0.1570 seconds to compute the forecast for a data set with more than 12000 values, namely, Home 1 data set



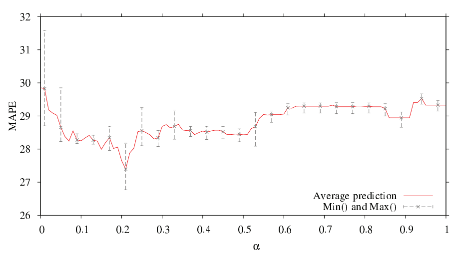
(a) First two sliding windows assessed by NMSE



(b) Full dataset assessed by NMSE



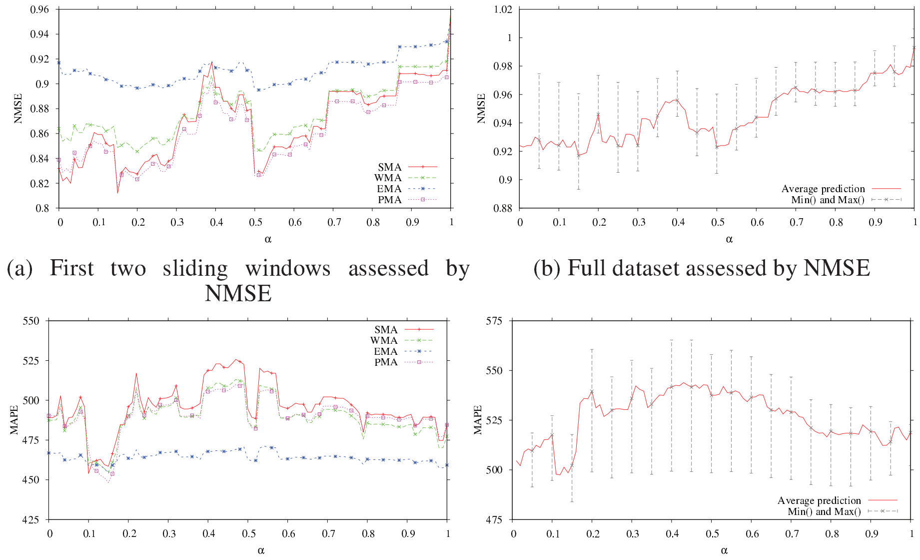
(c) First two sliding windows assessed by MAPE



(d) Full dataset assessed by MAPE

Figure 3. Evaluation of α parameter for the Dropbox dataset

DOI: 10.1002/nem



(a) First two sliding windows assessed by NMSE

(b) Full dataset assessed by NMSE

(c) First two sliding windows assessed by MAPE

(d) Full dataset assessed by MAPE

Figure 4. Evaluation of α parameter for the Data Center dataset

from Dropbox (see Subsection 4.4). After the α optimization, the DyWiSA computes the prediction in 0.0830 seconds. Details about the time consumption improvement, NMSE and MAPE for the other models are presented in Table II.

Table I. Sample of α evaluation

α value	***Model***	***Average***	***Std. Deviation***	***Changes***
0.30	*SMA*	29.79	16.47	759
	WMA	29.80	16.49	759
	EMA	29.81	17.11	759
	PMA	29.80	16.64	759
0.20	*SMA*	29.79	16.25	916
	WMA	29.80	16.28	916
	EMA	29.81	17.09	916
	PMA	29.81	16.51	916
0.10	*SMA*	29.70	16.24	2093
	WMA	29.74	16.27	2093
	EMA	29.81	17.09	2093
	PMA	29.79	16.51	2093
0.02	*SMA*	29.18	15.92	8070
	WMA	29.42	16.05	8070
	EMA	29.79	17.12	8070
	PMA	29.70	16.48	8070

Naturally, the minimum real-time measurement that can be achieved depends on the hardware configuration (*i.e.*, processor clock rate, memory available, etc.). For comparative purposes, all the tests performed in this work were based on a standard personal computer with a DualCore Intel Core 2 Duo CPU 6300 1.86GHz and 3Gb DDR2-SDRAM.

DOI: 10.1002/nem

Table II. Time consumption (seconds) and improvement

Model	α-*Fixed (s)*			α-*Optimized (s)*			*Time Improvement (%)*
	Time	NMSE	MAPE	Time	NMSE	MAPE	
SMA	0.0287	0.4045	41.58	0.0169	0.2709	28.18	41.11
WMA	0.1024	0.3422	36.50	0.0707	0.2590	27.47	30.96
EMA	0.1174	0.2807	29.36	0.0768	0.2600	27.20	34.58
PMA	0.1570	0.2720	28.90	0.0830	0.2543	26.77	47.13

It is evident that α has an important impact on the prediction quality, while also affecting the computational requirements of the Dynamic Window Size Algorithm. To warrant a fair comparison between all datasets evaluated in this work, we consider the same methodology to determine the best α from the initial sliding windows, as described above.

4.4. *Dropbox Case Study*

Firstly, we report the results of the static approach in which the sliding window size remains constant during the prediction data. We then present performance results for the dynamic approach, which calculates the sliding window size according to Algorithm 1.

4.4.1. Static Approach: As shown in [20], the evaluation of the network traffic prediction based on sliding window with static size is determined by metrics of the overall historical data, such as the global average. Then, the *Static Approach* demands an *a priori* analysis of the dataset. After this process, the analysis of the data generates several statistical descriptors that are employed as the size parameter of the static sliding windows. Table III provides results for three different values of the sliding window: arithmetic mean, standard deviation and variance. The first line shows statistics of the Dropbox dataset that was used as input for the predictors, while the second line exhibits results achieved for a trivial predictor that always predicts the next value as the arithmetic mean of data. The following lines show results for Poisson prediction model as well as three others (EMA, WMA and SMA) presented in Section 2.

While for most metrics the results of the remaining predictors are not far from those obtained by the Poisson approach, it is worth noting that the results achieved by this approach have the best performance concerning NMSE and MAPE, where PMA excels when compared to the others. This means that the difference between the estimated values and the real values is the lowest in the evaluation's result.

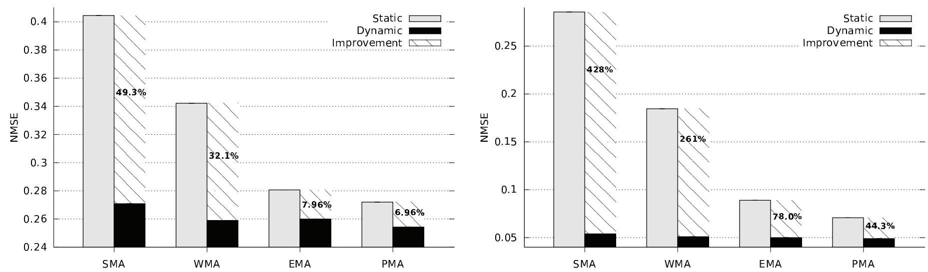
As illustrated in Table III, for the sliding window with size arithmetic mean, we note that the Mean Square Error ranges between 143.08 to 581.04 (the lowest error range), while for sliding window size with variance size, goes from 268.79 to 1908.38 (the highest error range). Thus, sliding window size arithmetic mean provides results more stable than other approaches. As we can observe, NMSE and MAPE values are the lowest for the arithmetic mean size (0.0494; 0.0887; 0.1845; 0.2856), indicating higher levels of accuracy.

We also use this methodology to estimate the best case scenario for the Home 1 in the static approach. It generates a table similar to Table III (omitted to avoid redundancy of information).

DOI: 10.1002/nem

Table III. Descriptive Statistics for Campus 2 dataset

Dataset		Mean			Std. Dev.	Variance	NMSE	MAPE
		Arithmetic	Square Error	Std. Error				
1	Dropbox	45.994	0.000	0.486	54.123	2929.4	0.000	0.000
2	Trivial	45.994	2929.4	0.000	0.000	0.000	1.000	337.98
Approach		**Sliding window size arithmetic mean**						
3	PMA	45.990	143.076	0.482	53.634	2876.71	0.0494	37.93
4	EMA	45.989	185.282	0.476	52.988	2807.81	0.0887	39.04
5	WMA	45.995	395.621	0.466	51.838	2687.19	0.1845	54.61
6	SMA	45.993	581.035	0.463	51.546	2657.02	0.2856	66.68
Approach		**Sliding window size standard deviation**						
7	PMA	45.988	208.785	0.474	52.834	2791.46	0.0745	38.36
8	EMA	45.990	278.076	0.468	52.045	2708.71	0.1026	41.93
9	WMA	45.999	594.747	0.455	50.679	2568.38	0.2313	57.43
10	SMA	45.993	931.031	0.449	50.063	2506.40	0.3714	71.31
Approach		**Sliding window size variance**						
11	PMA	45.988	268.790	0.471	52.468	2752.61	0.0974	41.43
12	EMA	45.990	381.512	0.457	50.829	2583.66	0.1475	46.19
13	WMA	46.003	1175.951	0.420	46.837	2193.73	0.5356	94.46
14	SMA	45.984	1908.381	0.405	45.121	2035.95	0.9372	134.69



(a) Home1 (b) Campus2

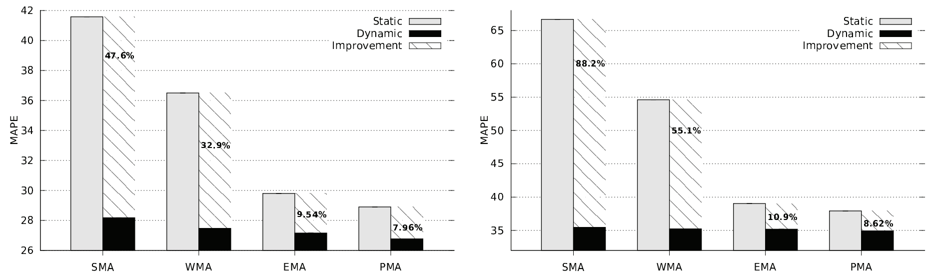
Figure 5. NMSE results from Dropbox datasets

Among different sliding window size assessed, the best result was achieved by the sliding window with arithmetic mean size, as illustrated in Table III.

4.4.2. Dynamic Approach: For the dynamic approach assessment we use Algorithm 1 for calculating the sliding window size. Furthermore, in the dynamic approach, the predictor models were evaluated from the two Dropbox traffic traces (Home 1 and Campus 2). Figure 5 illustrates the NMSE accuracy of the predictor models. All predictor models were tested in their original version with a static window size as well as with our dynamic window size methodology. Although our focus is on the comparison between predictor models operating with a static window size and a dynamic window size, we observe that SMA consistently provides the worst results, irrespectively of the window size methodology used. On the other extreme we have PMA, which provides the best overall results.

With respect to the comparison between the static and dynamic approach, our results show that all the predictor models achieve better results with our dynamic window size methodology. This is further evidenced in the NMSE results of Figure 5, which shows that all the predictors are improved from as little as 6.96% for the best predictor model (PMA) to as much as 428% for the worst predictor identified (SMA).

DOI: 10.1002/nem



(a) Home1 (b) Campus2

Figure 6. MAPE results from Dropbox datasets

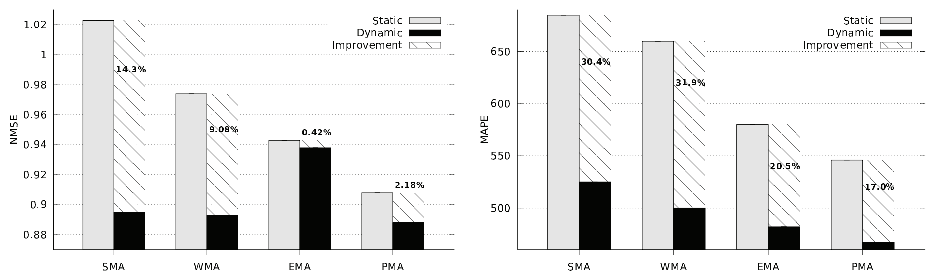
It is worth noticing that Figure 5a presents WMA with better result than EMA and this is not confirmed in Figure 6a. When the predictor model is assessed by NMSE, the data normalization process tends to improve the results of the predictor with the highest variance (see Equation 2). In this case, the WMA presents better results than EMA because its predicted data shows higher variance σ^2. In order to avoid the problem of larger variance of data, we also evaluate the Dynamic Window Size Algorithm by MAPE.

Figure 6 shows the performance of the predictor models in terms of error percentage. It is illustrated that in both cases (Home 1 and Campus 2) the error rate decreases using the dynamic window size methodology. The overall MAPE results show that the prediction results are improved for all predictors, from 7.96% (PMA) to 88.2% (SMA).

4.5. Data Center Case Study

We divide this evaluation reports in two parts. In first step we perform the prediction over the Data Center dataset with the static approach, *i.e.*, the window size remains invariable during the prediction data. In the next step, we show the dynamic approach results which calculates the sliding window size according to Algorithm 1.

4.5.1. Static Approach: It is important to note that the datasets assessed in this section also generate a table such as Table III for the static approach. However, in order to avoid redundancy of information, this assessment will be summarized in a graph. In Figure 7 we present the prediction result. In Figure 7a, we can see the bars in gray that represent the NMSE achieved for the SRD predictors (SMA, WMA, EMA and PMA). The same way that Dropbox case study, PMA presents



(a) NMSE (b) MAPE

Figure 7. NMSE and MAPE results for Data Center

Prepared using nemauth.cls
DOI: 10.1002/nem

the best result close to 0.91 for NMSE. Figure 7b shows the MAPE performance, where PMA again presents better results in comparison with other predictors assessed in this work.

4.5.2. Dynamic Approach: The Data Center was also evaluated by the Algorithm 1 in order to generate results from a dynamic window size perspective. Figure 7 illustrates several information about accuracy of the prediction models.

We observe that all predictors improve significantly their results after using the dynamic window size approach to forecast the time series, as confirmed by the NMSE and MAPE results of Figure 7. The EMA predictor has shown the smallest improvement with 0.42% and the SMA shows the best improvement with 14.3% regarding NMSE. For MAPE, Figure 7b depicts improvements from 17% (PMA) up to 31.9% for WMA.

Throughout this work, several descriptive statistics to measure the improvement of the solution were presented. Focusing on the level of accuracy, we found that NMSE provides optimistic performance results when the time series present high σ^2 values, whereas MAPE proved to be better suited for measuring data with high volatility, by measuring the error in terms of percentage of the real data over the prediction data.

Both metrics concede that the moving average approach represents a SRD solution that computes a local average of data at the end of the time window, on the assumption that this is the best estimate to represent the current mean value around which the data are ranging. These models are suitable if the time series change suddenly, as happens with cloud computing traffic. In this case, an anomaly may be easily diluted inside the time window without compromising the prediction in whole [10]. Moreover, after using DyWiSA with the α optimization, all predictor models have decreased the time and the workload to compute the prediction of the data.

In particular, with a smaller sliding window, oldest values also have lower influence on the predicted network traffic. This indicates that a predictor that prioritizes recent history achieves better results for dynamic cloud computing environments. In addition, SRD solutions are able to provide accurate predictions with relatively low levels of historical data dependency.

5. CONCLUSIONS

Network traffic prediction is a powerful tool that supports several management tasks such as keeping track of resources in the network and how they are assigned, or monitoring the network to spot problems as soon as possible, ideally before users are affected. In this work, we propose a Dynamic Window Size Algorithm along with a methodology for optimization of its threshold for window size changes. Apart from facilitating online traffic prediction due to its short dependency on historical data, the new methodology to exploit the α parameter improved the accuracy of the four traffic predictors considered in this work. Furthermore, all predictor models have also reduced the time and the workload to compute the forecast.

From the observation of the results, we can see that all the Short-range Dependence (SRD) predictors present a considerable improvement after using the α optimization in the Dynamic Window Size Algorithm. Moreover, compared to others predictors, the Simple Moving Average

Int. J. Network Mgmt (2016)
DOI: 10.1002/nem

performed considerably worse, whereas the Poisson Moving Average was more suitable for dynamic cloud environments.

By considering a new *Data Center* dataset with traffic from a diverse set of common cloud services, we were able to provide more general results on the performance of the dynamic window size methodology. These results have shown compliance with earlier results from the *Dropbox* data set, therefore strengthening the validity of the Dynamic Window Size Algorithm for traffic prediction in highly volatile environments. Prospective directions for future work include using the Dynamic Window Size Algorithm to perform anomaly detection of network traffic in virtualized environments.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive suggestions to improve the quality of the paper. This work was partially funded by the iCIS project, under the grant CENTRO-07-ST24-FEDER-002003; and CAPES and CNPq (Brazil) through the Ciência sem Fronteiras Program/2016.

REFERENCES

1. P. Mell and T. Grance. “The NIST definition of cloud computing,” *National Institute of Standards and Technology*, Computer Security Division, Information Technology Laboratory, 2011.
2. B. Jennings and R. Stadler, “Resource management in clouds: Survey and research challenges,” *Journal of Network and Systems Management*, Springer International Publishing, pp. 1–53, 2014.
3. H. Alshaer, “An overview of network virtualization and cloud network as a service,” *International Journal of Network Management*, vol. 25, no. 1, pp. 1–30, 2015.
4. D. Plonka and P. Barford, “Network anomaly confirmation, diagnosis and remediation,” in *47th Annual Allerton Conference on Communication, Control, and Computing. Allerton 2009.*, pp. 128–135, Oct 2009.
5. G. Aceto, A. Botta, W. de Donato, and A. Pescapè, “Cloud monitoring: A survey,” *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.
6. A. Dainotti, A. Pescape, and K. Claffy, “Issues and future directions in traffic classification,” *IEEE Network*, vol. 26, pp. 35–40, january-february 2012.
7. S. Kubler, W. Derigent, E. Rondeau, A. Thomas, and K. Främling, “Embedded Data on Intelligent Products – Impact on Real-Time Applications,” in *Trends in Mobile Web Information Systems*, vol. 183 of *Communications in Computer and Information Science*, pp. 25–34, Springer International Publishing, 2013.
8. H. Ballani, P. Costa, T. Karagiannis, and A. I. Rowstron, “Towards predictable datacenter networks.,” in *SIGCOMM*, vol. 11, pp. 242–253, 2011.
9. K. Vieira, A. Schulter, C. Westphall, and C. Westphall, “Intrusion Detection for Grid and Cloud Computing,” *It Professional*, vol. 12, no. 4, pp. 38–43, 2010.
10. M. L. Proença, G. Fernandes, L. F. Carvalho, M. V. O. de Assis, and J. J. P. C. Rodrigues, “Digital signature to help network management using flow analysis,” *International Journal of Network Management*, 2015.
11. W. Xiong, H. Hu, N. Xiong, L. T. Yang, W.-C. Peng, X. Wang, and Y. Qu, “Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications,” *Information Sciences*, vol. 258, pp. 403–415, 2013.
12. M. Rahmani, K. Tappayuthpijarn, B. Krebs, E. Steinbach, and R. Bogenberger, “Traffic shaping for resource-efficient in-vehicle communication,” *IEEE Transactions on Industrial Informatics*, vol. 5, pp. 414–428, Nov 2009.
13. M. Papadopouli, E. Raftopoulos, and H. Shen, “Evaluation of short-term traffic forecasting algorithms in wireless networks,” in *Conference on Next Generation Internet Design and Engineering. NGI'06.*, pp. 102–109, IEEE, 2006.
14. R. Makhloufi, G. Doyen, G. Bonnet, and D. Gaïti, “A survey and performance evaluation of decentralized aggregation schemes for autonomic management,” *International Journal of Network Management*, vol. 24, no. 6, pp. 469–498, 2014.

15. J. Taylor, "Short-term load forecasting with exponentially weighted methods," *IEEE Transactions on Power Systems*, vol. 27, pp. 458–464, Feb 2012.
16. A. Li, Y. Han, B. Zhou, W. Han, and Y. Jia, "Detecting Hidden Anomalies Using Sketch for High-speed Network Data Stream Monitoring," *Applied Mathematics*, vol. 6, no. 3, pp. 759–765, 2012.
17. F. Klinker, "Exponential moving average versus moving exponential average," *Mathematische Semesterberichte*, vol. 58, no. 1, pp. 97–107, 2011.
18. J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology. Wireless VITAE 2009.*, pp. 6–10, May 2009.
19. B. L. Dalmazo, J. P. Vilela, and M. Curado, "Predicting traffic in the cloud: A statistical approach," in *Third International Conference on Cloud and Green Computing (CGC'13)*, pp. 121–126, Sept-Oct 2013.
20. B. L. Dalmazo, J. P. Vilela, and M. Curado, "Online traffic prediction in the cloud: A dynamic window approach," in *The 2nd International Conference on Future Internet of Things and Cloud (FiCloud'2014)*, pp: 9–14, Aug 2014.
21. Y. Chen, B. Yang, and Q. Meng, "Small-time scale network traffic prediction based on flexible neural tree," *Applied Soft Computing*, vol. 12, no. 1, pp. 274–279, 2012.
22. H. Zare Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," in *International Symposium on Information Technology. ITSim 2008.*, vol. 4, pp. 1–6, 2008.
23. W.-K. Kuo and K.-W. Wu, "Traffic prediction and QoS transmission of real-time live VBR videos in WLANs," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7, pp. 36:1–36:21, Dec. 2011.
24. R. Yadav and M. Balakrishnan, "Comparative evaluation of ARIMA and ANFIS for modeling of wireless network traffic time series," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, 2014.
25. H. Kwon, T. Kim, S. Yu, and H. Kim, "Self-similarity based lightweight intrusion detection method for cloud computing," in *Intelligent Information and Database Systems*, *Lecture Notes in Computer Science*, vol. 6592, pp. 353–362, Springer Berlin Heidelberg, 2011.
26. P. Loiseau, P. Gonçalves, G. Dewaele, P. Borgnat, P. Abry, and P. V.-B. Primet, "Investigating self-similarity and heavy-tailed distributions on a large-scale experimental facility," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1261–1274, Aug. 2010.
27. A. S. Weigend and N. A. Gershenfeld, eds., *Time series prediction: Forecasting the future and understanding the past*, Westview Press, 1994.
28. S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting methods and applications* (3rd edn). John Wiley & Sons, 2008.
29. I. Drago, M. Mellia, M. M. Munafò, A. Sperotto, R. Sadre, and A. Pras, "Inside Dropbox: Understanding Personal Cloud Storage Services," in *Proceedings of the 12th ACM SIGCOMM Conference on Internet Measurement.* Berlin, Germany. IMC'12, pp. 481–494, 2012.
30. T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, New York, NY, USA, pp. 267–280, ACM, 2010.

APPENDICES

A. *Proof for Proposition 1*

If we know the minimum and maximum range, *e.g.* from y_a to y_b, we are able to represent its average m by:

$$m = \frac{q_a y_a + q_b y_b}{q_a + q_b} \tag{A.1}$$

where q_a and q_b are the quantity of y_a and y_b, respectively. Then, if we consider the average and these extreme values as referred before to estimate the maximum variance σ^2_{max} into a sliding window, it may be expressed for:

DOI: 10.1002/nem

$$\sigma^2_{max} = \frac{q_a(m - y_a)^2 + q_b(y_b - m)^2}{q_a + q_b}$$

$$\sigma^2_{max} = \frac{q_a(m^2 - 2my_a + y_a^2) + q_b(m^2 - 2my_b + y_b^2)}{q_a + q_b}$$

$$\sigma^2_{max} = \frac{(q_a + q_b)m^2}{q_a + q_b} - \frac{2(q_a y_a + q_b y_b)m}{q_a + q_b} + \frac{q_a y_a^2 + q_b y_b^2}{q_a + q_b} \tag{A.2}$$

Simplifying the first term in Equation A.2 and substituting the second term by Equation A.1 into it, we achieve:

$$\sigma^2_{max} = m^2 - 2m^2 + \frac{q_a y_a^2 + q_b y_b^2}{q_a + q_b} \tag{A.3}$$

Now, isolating the term $q_a y_a$ from the Equation A.1 we have:

$$q_a y_a = m(q_a + q_b) - q_b y_b \tag{A.4}$$

And similarly:

$$q_b y_b = m(q_a + q_b) - q_a y_a \tag{A.5}$$

Using these two equations (A.4 and A.5) into the Equation A.3, we have:

$$\sigma^2_{max} = -m^2 + \frac{y_a(m(q_a + q_b) - q_b y_b) + y_b(m(q_a + q_b) - q_a y_a)}{q_a + q_b}$$

Evidencing the term $q_a + q_b$ of the equation,

$$\sigma^2_{max} = -m^2 + \frac{m(q_a + q_b)(y_a + y_b) - (q_a + q_b)(y_a y_b)}{q_a + q_b}$$

$$\sigma^2_{max} = -m^2 + m(y_a + y_b) - y_a y_b$$

Evidencing the term $y_b - m$,

$$\sigma^2_{max} = m(y_b - m) - y_a(y_b - m) \tag{A.6}$$

So, we may represent the σ^2_{max} just acknowledging the minimum, the maximum and the average of the data inside the sliding window. In addition, the Equation A.6 is equivalent to the Equation 1. This finally leads to the results presented in Proposition 1.

DOI: 10.1002/nem

AUTHORS' BIOGRAPHIES

Bruno L. Dalmazo has been a Ph.D. student at the University of Coimbra since 2011. He completed his Masters degree in Computer Science in 2011 at the Federal University of Rio Grande do Sul, Brazil. Bruno also received a Bachelors degree in Computer Science in 2008 from the Federal University of Santa Maria, Brazil. His main research interests involve network traffic prediction, as well as security and privacy in cloud computing environments.

João Vilela is an invited assistant professor at the Department of Informatics Engineering of the University of Coimbra, Portugal. He received his Ph.D. in Computer Science in 2011 from the University of Porto, Portugal. Dr. Vilela was visiting the Coding, Communications and Information Theory Group at Georgia Tech, working on physical layer security, and the Network Coding and Reliable Communications Group at MIT, working on security for network coding. His main research interests are in security and privacy of computer and communication systems, with focus on wireless networks and cloud computing.

Marilia Curado is an Assistant Professor at the Department of Informatics Engineering, University of Coimbra, Portugal, where she received a Ph.D. in Informatics Engineering on the subject of Quality of Service Routing. Her research interests are Quality of Service, Mobility, Routing, and Resilience.

DOI: 10.1002/nem