Online Traffic Prediction in the Cloud: A Dynamic Window Approach

Bruno L. Dalmazo, João P. Vilela, Marilia Curado
CISUC, Department of Informatics Engineering
University of Coimbra, Coimbra, Portugal
{dalmazo, jpvilela, marilia}@dei.uc.pt

***Abstract*—Traffic prediction is a fundamental tool that captures the inherent behavior of a network and can be used for monitoring and managing network traffic. Online traffic prediction is usually performed based on large historical data used in training algorithms. This may not be suitable to highly volatile environments, such as cloud computing, where the coupling between observations decreases quickly with time. We propose a dynamic window size approach for traffic prediction that can be incorporated with different traffic predictions mechanisms, making them suitable to online traffic prediction by adapting the amount of traffic that must be analyzed in accordance to the variability of data traffic. The evaluation of the proposed solution is performed for several prediction mechanisms by assessing the Normalized Mean Square Error and Mean Absolute Percent Error of predicted values over observed values from a real cloud computing data set, collected by monitoring the utilization of Dropbox.**

***Index Terms*—Cloud computing, network traffic prediction, short-range dependence, sliding window algorithm.**

I. Introduction

Fomented by cloud computing, many services and products are currently offered exclusively through the Internet. More than ever, neglecting the management of these networks may cause irreparable economic harm to businesses and their customers. Network administrators of these cloud-supporting networks must then monitor and analyze those networks in order to collect relevant information about network traffic that may be used to support decision-making.

Network traffic monitors constantly keep statistics of network connectivity and applications' availability, facilitating the detection of problems in hosts, networks or services. Some of the metrics usually considered with respect to network traffic are the throughput, response time, jitter and lost data. After capturing all this information, it is possible to analyze and identify suspicious patterns of network traffic. These patterns help to plan strategies to prevent similar problems that may happen in the future [1]. When these statistics accumulate over time, it is possible to make inferences about the future behavior of network traffic. Thus, when an abnormality is expected, the network administrator will have time to act even before the problem arises.

Constant monitoring can generate a large overhead in the management system, and thus increase the associated cost to supervise the entire network infrastructure. Reducing the service and operating costs is crucial for a successful management. In this context, the management system requires an efficient technique which consumes minimal resources and time from support staff. Solving the problem of processing large amounts of data for traffic prediction represents an important achievement in order to avoid unnecessary overhead and minimize these costs. Moreover, dynamically reducing the amount of information to process is also relevant to other applications, as traffic shaping for improved Quality of Service [2] or to conceive more accurate simulation models [3].

However, characterizing 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 [4]. Most monitoring tools currently available provide a graphical interface of network statistics, from which problems can be identified and remedied [5]. This challenge is even greater in cloud computing because its traffic may suffer sudden changes [6], [7], and the elastic and scalable nature of cloud environments may be easily confused with traffic anomalies, hampering the management of the network.

The effective monitorization of computer networks must be constantly performed to favor the detection of possible issues as they happen or even before-hand, therefore calling for online traffic prediction mechanisms. 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 [8].

In order to address this issue, we propose a dynamic window size methodology for traffic prediction. The size of the window defines the amount of traffic that is considered for traffic prediction, and varies according to bounded historical data, therefore making it suitable to environments where Short-range Dependence (SRD) is deemed necessary, such as cloud computing. This mechanism can be applied to determine the scope of data to be analyzed for any traffic prediction approach.

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

II. 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. 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.

A. Short-range dependency traffic prediction

Maria Papadopouli *et al.* [3] 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 requires a large amount of historical data. They emphasize some advantages of SMA, such as its simplicity, low complexity and ease of application.

Aiping Li *et al.* [9] 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 [10], Frank Klinker describes mathematical tools to identify and predict market trends. In particular, it shows that the Exponential Moving Average (EMA) can be used for efficient forecast of network traffic with short historical data.

In a previous work [11], we propose a systematic approach for estimating network traffic resorting to a statistical method based on a Poisson process (Poisson Moving Average - PMA). In this work, we have used a sliding window with static size to weight past observations by taking advantage of well-known network traffic features such as short-range dependence.

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 scenarios where previous data is available.

B. Online traffic prediction

Yuehui Chen *et al.* [12] 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 [13] 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.

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. In this work we propose a dynamic sliding window mechanism based on SRD for traffic prediction. This solution provides 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.

III. Dynamic Sliding Window Mechanism

The dynamic sliding window is a mechanism to limit the amount of information that is used for traffic prediction, therefore making it suitable [11] for online prediction in a cloud computing context.

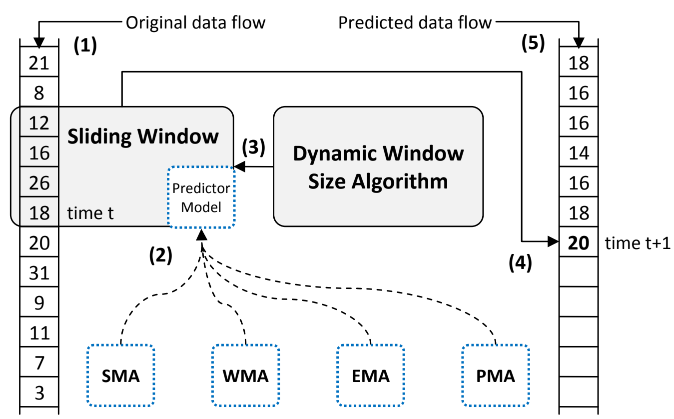


Fig. 1: Elements of the proposed solution and iterations

Figure 1 illustrates the main conceptual components and their interactions. Real-time cloud traffic data (flow 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 flow 2. Possible candidates for the predictor model (described in Section II-A) 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 (flow 3). The next value of cloud data traffic is predicted (flow 4) according to the chosen predictor model, therefore resulting in a sequence of predicted values for the cloud data traffic (flow 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.

A. 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). 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 [11]. 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.

B. Dynamic Window Size Algorithm

Traffic predictors usually operate over all of previous data [12] 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 [6]

This constraint's identification has led to consider the sliding window approach as a forgetting process which allows to limit 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 generates low 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 between the previous and current sliding window.

Algorithm 1 describes the operation of the Dynamic Window Size Algorithm. It resorts to a sliding windows of variable size, which changes according to the variance (σ^2) found in the last sliding window and the current sliding window. A small variance indicates that the predicted data is close to the mean, while a high variance indicates that the predicted data is spread out from the mean. We consider the theoretical maximum variance (σ^2_{max}) to be the variance of the extreme values of a sliding window.

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 window with the average of the current sliding window. In order to avoid unnecessary overhead of the algorithm, we select an α equal to 0.05 which represents 95% confidence interval. In this context, confidence interval 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 5% level.

Algorithm 1 Dynamic window size

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

1: **Start**
2:   **procedure** DYWISA($newAvg$, $oldAvg$, $sWindow$)
3:     $var\ wSize = sWindow.size()$
4:     $var\ direct = newAvg/oldAvg$
5:     $var\ inverse = oldAvg/newAvg$
6:     $var\ ratio = sqrt(direct - inverse)^2$
7:     **if** ($ratio > (1 + \alpha)$) **then**
8:       $var\ volume = f(sWindow)$
9:       **if** ($newAvg > oldAvg$) **then**
10:         $wSize = wSize + volume$
11:       **else**
12:         $wSize = wSize - volume$
13:       **end if**
14:     **end if**
15:     **return** $wSize$
16:   **end procedure**
17: **End**

Let *direct* be a value which measures average changes between the current window and last window, and *inverse* represent its inverse. If the difference between *direct* and *inverse* 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. For this, we use the ratio between the σ^2 and the σ^2_{max} inside a sliding window. This whole process is represented by the function f at line 8 of the 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 follow:

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

Proof. See Appendix A. ∎

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.

IV. Evaluation and Discussion

In this section, we perform an evaluation of the dynamic sliding window mechanism applied to all SRD traffic prediction mechanisms of Section II-A.

A. Setup and Metrics

All the measurements and data presented in this paper were collected from March 24, 2012 to May 5, 2012. The evaluated dataset is focused on Dropbox utilization, which is the most widely-used cloud storage system nowadays [14]. The Dropbox dataset encompasses more than 100 metrics about the network traffic. However, for this study we only consider the total number of packets observed from the client (server) to the server (client). In order to evaluate the solution, we used two datasets as described in the [14], 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. The datasets Home 1 and Campus 2 have 18,785 and 2,528 distinct client IP addresses, respectively. The datasets were divided in intervals of five minutes each, and the evaluation was performed by applying a sliding window weighted with the four statistical models described in Section II-A.

The effectiveness of the prediction mechanisms is measured through the Normalized Mean Square Error (NMSE) [15] and Mean Absolute Percent Error (MAPE) [16]. NMSE is defined as:

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

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, 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 [15].

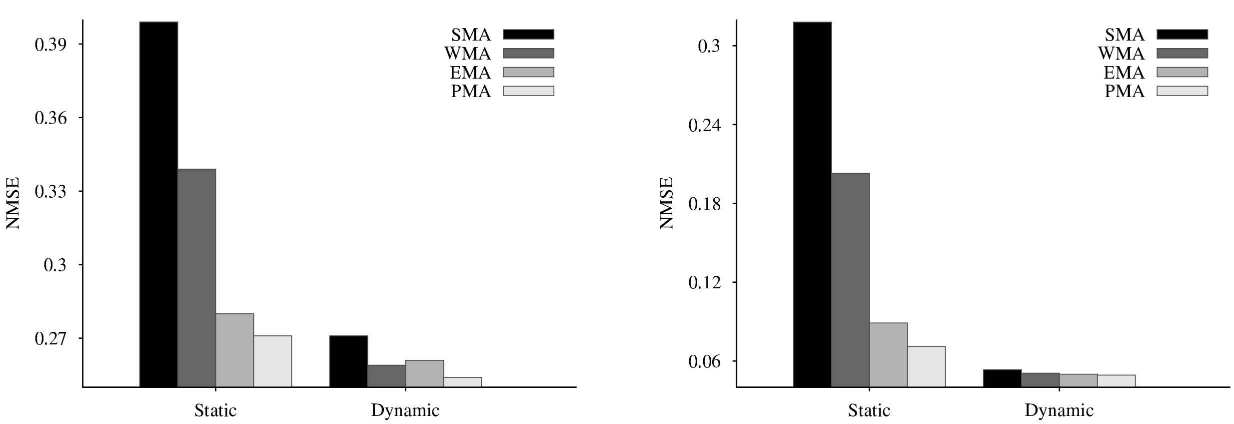
MAPE measures expressed errors as a percentage of the actual data over the prediction data. It is calculated as the average of the unsigned percentage error, and is defined by the formula:

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

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 as well as referenced in NMSE. 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.

B. Results and Discussion

Figure 2 illustrates the accuracy of the predictor models for the two Dropbox traffic traces (Home 1 and Campus 2). All predictor models were tested in its 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.



(a) Home1

(b) Campus2

Fig. 2: NMSE results from Dropbox datasets

With respect to the comparison between the static and dynamic approach, our results show that all predictor models achieve better results with our dynamic window size methodology. This is further evidenced in the NMSE results of Table I, which shows that all predictors are improved from as little as 6.69% for the best predictor model (PMA) to as much as 495.51% for the worst predictor identified (SMA).

TABLE I: NMSE between static and dynamic approach

Dataset	Model	Static	Dynamic	Improvement (%)
Home1	SMA	0.399	0.271	47.23
	WMA	0.339	0.259	30.89
	EMA	0.280	0.261	7.28
	PMA	0.271	0.254	6.69
Campus2	SMA	0.318	0.0534	495.51
	WMA	0.203	0.0507	300.39
	EMA	0.089	0.0499	78.36
	PMA	0.071	0.0493	44.02

It is worth noticing that Figure 2 (a) presents WMA with better result than EMA and this is not confirmed in Figure 3 (a). When the predictor model is assessed by MNSE, 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 the higher σ^2. In order to avoid the problem of larger variance of data, we also evaluate the Dynamic Window Size Algorithm by MAPE.

Figure 3 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's rate decreases using the dynamic window size methodology. The overall MAPE results can be seen in the Table II, which shows that the prediction results are improved for all predictors, from 7.66% (PMA) to 101.21% (SMA).

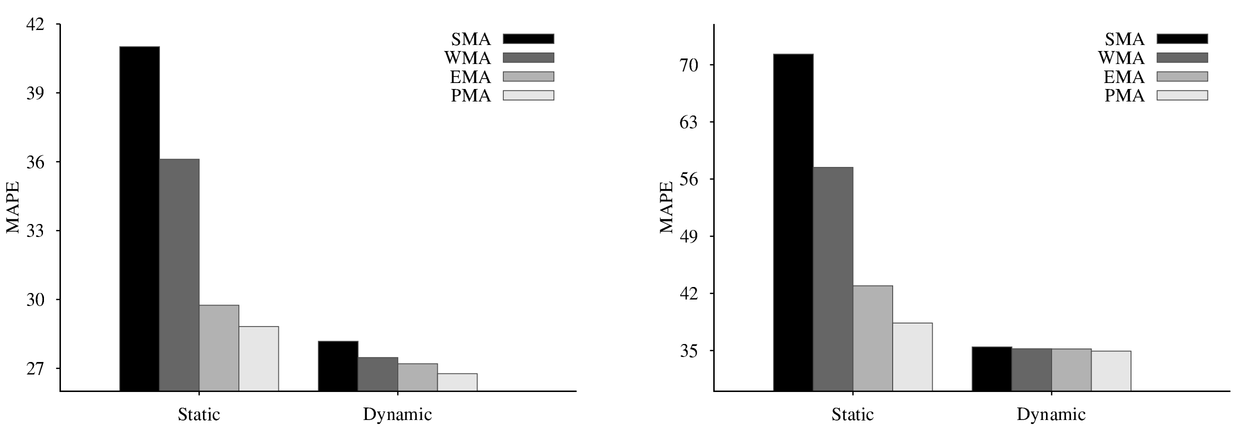
In summary, the moving average approach represents a SRD solution that computes a local average of data at the end of the time window, on the assumption that this is the best estimate to represent the current mean value around which the data are ranging. These models are suitable if the time series changes suddenly as cloud computing traffic does. In this case, an anomaly may be easily diluted inside of the time window without compromising the prediction in whole. From the observation of the results, we can see that all the predictions present a considerable improvement after using the Dynamic Window Size Algorithm. Highlighting the Poisson Moving Average approach, which has shown to be more suitable for dynamic cloud environments than other assessed predictor models. More importantly, the use of the dynamic window size methodology limits the amount of information required for the prediction, therefore allowing these predictor models to be employed for online traffic prediction.

TABLE II: MAPE between static and dynamic approach

Dataset	Model	Static	Dynamic	Improvement (%)
Home1	SMA	41.01	28.18	45.53
	WMA	36.11	27.47	31.45
	EMA	29.75	27.20	9.38
	PMA	28.82	26.77	7.66
Campus2	SMA	71.31	35.44	101.21
	WMA	57.43	35.22	63.06
	EMA	42.93	35.19	21.99
	PMA	38.36	34.92	9.85

V. Conclusions

Prediction of network traffic is relevant for many management applications such as resource allocation, admission control and congestion control. In this paper we propose a dynamic window size methodology to incorporate with existing traffic prediction mechanisms. Apart from facilitating online traffic prediction due to its short dependency on historical data, the dynamic window size approach improved the accuracy of the four traffic predictors considered when applied to a real



(a) Home1 (b) Campus2

Fig. 3: MAPE results from Dropbox datasets

data set of Dropbox traffic. Prospective directions for future work include using this methodology to perform anomaly detection of network traffic in virtualized environments.

ACKNOWLEDGMENT

This work was partially funded by the project CMU-PT/RNQ/0015/2009, Trustworthy and Resilient Operations in a Network Environment (TRONE); the iCIS project, under the grant CENTRO-07-ST24-FEDER-002003; and CAPES and CNPq (Brazil) through the Ciência sem Fronteiras Program/2014.

REFERENCES

[1] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," *Network, IEEE*, vol. 26, no. 1, pp. 35–40, january-february 2012.

[2] 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, no. 4, pp. 414–428, Nov. 2009.

[3] 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, 2006. NGI'06.* IEEE, 2006, pp. 102–109.

[4] 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*, ser. Communications in Computer and Information Science, M. Matera and G. Rossi, Eds. Springer International Publishing, 2013, vol. 183, pp. 25–34.

[5] D. Plonka and P. Barford, "Network anomaly confirmation, diagnosis and remediation," in *47th Annual Allerton Conference on Communication, Control, and Computing, 2009. Allerton 2009.*, 30 2009-oct. 2 2009, pp. 128–135.

[6] H. Ballani, P. Costa, T. Karagiannis, and A. I. Rowstron, "Towards predictable datacenter networks." in *SIGCOMM*, vol. 11, 2011, pp. 242–253.

[7] 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.

[8] 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, no. 0, pp. 403–415, 2013.

[9] 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.

[10] F. Klinker, "Exponential moving average versus moving exponential average," *Mathematische Semesterberichte*, vol. 58, no. 1, pp. 97–107, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00591-010-0080-8

[11] 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), 2013*, Sept-Oct 2013, pp. 121–126.

[12] 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.

[13] H. Zare Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," in *International Symposium on Information Technology, 2008. ITSim 2008.*, vol. 4, 2008, pp. 1–6.

[14] 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.*, ser. IMC'12, 2012.

[15] A. S. Weigend and N. A. Gershenfeld, Eds., *Time series prediction: Forecasting the future and understanding the past*. Westview Press, 1994.

[16] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting methods and applications*. John Wiley & Sons, 2008.

APPENDIX

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:

$$m = \frac{q_a y_a + q_b y_b}{q_a + q_b} \tag{3}$$

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:

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

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

$$\sigma^2_{max} = \frac{(q_a+q_b)m^2}{q_a+q_b} - \frac{2(q_a y_a + q_b y_b)m}{q_a+q_b} + \frac{q_a y_a^2 + q_b y_b^2}{q_a+q_b} \tag{4}$$

Simplifying the first term in Equation 4 and substituting the second term by Equation 3 into it, we achieve:

$$\sigma^2_{max} = m^2 - 2m^2 + \frac{q_a y_a^2 + q_b y_b^2}{q_a+q_b} \tag{5}$$

Now, isolating the term $q_a y_a$ from the Equation 3 we have:

$$q_a y_a = m(q_a+q_b) - q_b y_b \tag{6}$$

And similarly:

$$q_b y_b = m(q_a+q_b) - q_a y_a \tag{7}$$

Using these two equations (6 and 7) into the Equation 5, 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 $y_b - m$,

$$\sigma^2_{max} = m(y_b - m) - y_a(y_b - m) \tag{8}$$

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 8 is equivalent to the Equation 1. This finally leads to the results presented in Proposition 1.