Predicting Traffic in the Cloud: A Statistical Approach

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

***Abstract*—Monitoring and managing traffic are vital elements to the operation of a network. Traffic prediction is an essential tool that captures the underlying behavior of a network and can be used, for example, to detect anomalies by defining acceptable data traffic thresholds. In this context, most current solutions are heavily based on historical time data, which makes it difficult to employ them in a dynamic environment such as cloud computing. We propose a traffic prediction approach based on a statistical model where observations are weighted with a Poisson distribution inside a sliding window. The evaluation of the proposed method is performed by assessing the Normalized Mean Square Error of predicted values over observed values from a real cloud computing dataset, collected by monitoring the utilization of Dropbox. Compared with other predictors, our solution exhibits the strongest correlation level and shows a close match with real observations.**

***Keywords*—Network traffic analysis, network traffic prediction, sliding window, Poisson process, Dropbox.**

I. Introduction

Cloud computing is at the core of the always connected paradigm, in which users access their data any time and anywhere requiring only a device with Internet access. This has increased the continuing demand for ubiquity and more powerful resources, making cloud computing a solution that perfectly matches need with efficiency. A recent study published in *The New York Times* [1] estimates that companies in the U.S.A. using cloud computing can save $12.3 billion per year by 2020, in addition to making annual reductions in carbon emissions equivalent to 200 million barrels of oil.

Cloud computing resorts to different delivery models that define what kind of services are provided to the end user. These are usually classified as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) which respectively provide software, platforms and infrastructure resources to the consumer. These service models place different levels of management requirements in the cloud environment.

Collecting statistics of the network is a useful management task that enables traffic patterns to be understood and strategies to be planned in order to prevent future problems. When these statistics accumulate over time, inferences can be made with respect to the future behavior of the network traffic and when an abnormality occurs the administrator will have enough time to act before the problem worsens. However, in cloud computing environments this task is complicated. For instance, the fact that the same infrastructure is shared by different users makes any attempt to block traffic or blacklist devices extremely risky since, in this case, a legitimate user may suffer the same problems caused by a denial-of-service attack. For these reasons, efficient management and analysis of data traffic in virtualized environments becomes essential to decision-making, thereby increasing the reliability of the services provided [2].

One of the main challenges of monitoring a cloud computing environment is the dynamics of network topologies with high communication rates between heterogeneous hosts [3]. This calls for traffic prediction mechanisms that do not depend on large historical data. In order to address this need, we propose a statistics-based solution that enables prediction of network traffic in cloud computing by resorting to observations of recent traffic only. This proposal is tailored for an IaaS delivery model, by performing an analysis of traffic at physical devices in the lower end of the infrastructure. This analysis is performed independently at each physical device, therefore reducing the complexity in terms of volume of data to analyze when compared to centralized solutions.

There are many benefits to the accurate prediction of network traffic. For instance, it can be used as a tool for extracting the baseline of network traffic to capture the underlying traffic trend [4] and facilitate techniques such as traffic shaping for improved Quality of Service [5]. Moreover, it allows the detection of anomalies by blending network traffic forecasts with tolerable operational thresholds [6]. The method proposed in this paper is designed to facilitate security and monitoring services at the lower end of the infrastructure.

The remainder of the paper is organized as follows. Section II covers some of the most prominent related work. Section III describes the statistical prediction approach 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 management of computer networks. Although most research efforts are focused on classical methods strongly based on historical data such as time series and neural networks, we also analyze previous works that have a short

dependency on historical data.

Among the several estimation techniques considered, some are used with the objective of predicting the behavior of network traffic. We now discuss those publications that have been found as correlated to our proposal, even when not concerned specifically with scalability and dynamics of virtualized environments. These works are divided according to their dependency historical data.

A. Long-range Dependency

In [7], Li and Lim identify a noticeable behavior of traffic, which is called "burstiness" or "packet trains", defined by peak-to-average transmission rate. This behavior is characterized by a long repetition of intervals of time in which firstly no packets are transmitted, and afterwards a wave of packets is sent. In this work, the network traffic is studied from the perspective of fractal time series. Using this approach, it is possible to project time series predicting the future behavior of the network. This approach is used to study specific parameters (such as the Hurst parameter) and relies on playback of offline traffic, taking into consideration traffic properties such as long-range dependency and heavy-tailed distribution. These properties relate to large historical data, therefore making this approach unsuitable for real-time traffic monitoring in dynamic environments.

A. Dainotti *et al.* [8] resort to a statistical-based approach to perform network traffic classification by associating network traffic with different categories of network applications. Traffic prediction is performed by taking into account characteristics such as inter-packet times and payload size, as well as their temporal correlation. The proposed solution is focused on packet-level traffic classification based on a Hidden Markov Model. The goal of this work is to use the obtained classification to offer different levels of QoS depending on the class of traffic. It also facilitates enforcing security policies to different applications and identification of malicious traffic flows. All evaluations are performed by analyzing offline traffic from different network topologies.

B. Short-range Dependency

Another study [9] analyzes the flow trend of two types of packet flows: inflow and outflow of data packets. This work highlights some problems that could occur, namely the volatility clustering problems and its effect on deteriorating the accuracy of short-term predictions. The proposed model is enhanced by Adaptive Support Vector Regression to form a linear combination of two models (Adaptive Neuro-Fuzzy Inference System and Nonlinear Generalized Autoregressive Conditional Heteroscedasticity) in order to not only simplify the complexity of the system, but also improve the prediction accuracy by solving the overshoot and volatility clustering problems. This scheme can act as a core component of network traffic analysis in order to help a network manager in providing network traffic control. Due to the several algorithms and statistical calculations employed, this approach is deemed heavy and requires high processing overhead, therefore not being suitable for real-time monitoring in a cloud computing environment.

In the field of forecast but specifically in the fresh food sales area, Wan-I Lee *et al.* [10] do a comparative study among prediction models. Their work characterizes the Simple Moving Average (SMA) as a time series analysis forecasting method. It emphasizes some advantages of SMA, such as its simplicity, low complexity and ease of application. It also gives a basic and efficient tendency index. Based on its simplicity, this method is commonly used to make forecasts from historical data.

Aiping Li *et al.* [11] study anomaly detection methods for high-speed network traffic. The purpose in 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.

Our solution provides a systematic approach for estimating network traffic resorting to a statistical method based on a sliding window with a weighted Poisson process. This work differs from previous works by taking into account the characteristics of the cloud computing environment, as shown in Table I. In particular, it provides high accuracy with low levels of historical dependency and operates in a distributed manner by having each device perform traffic prediction independently. This provides a low complexity traffic prediction solution by reducing the amount of data necessary to process when compared to a centralized system.

III. Statistical Prediction Approach

It is known that there are differences between the normal behavior of network traffic and an anomaly. However, the transition between these two extremes is obscure, i.e., we do not know at which point the network traffic ceases to represent legitimate use and should be considered an anomaly, as illustrated by Figure 1. Studies have shown that the task of identifying anomalous behavior in network traffic is not a

TABLE I: Summarized Related Work

Desirable features	**Model**				
	Ming Li, 2008	**B. Chang, 2009**	**A. Dainotti, 2012**	**Wan-I Lee, 2012**	**Aiping Li, 2012**
Low complexity	×	×	√	√	√
Anomaly detection	×	√	×	×	√
Low historical dependency	×	√	×	√	√
High accuracy	√	√	√	√	×

trivial matter [12]. This challenge is even greater in cloud computing due to the scalability and dynamics of this environment.

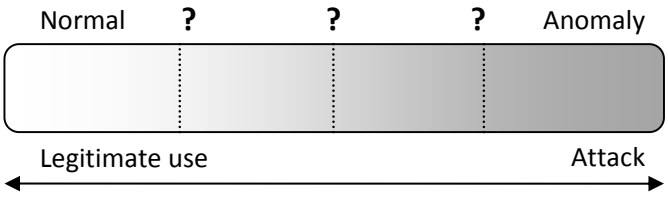

Fig. 1: Network traffic behavior

A. Preliminaries

The characteristics of network traffic have been studied in works such as [7], which indicate that parameters such as the number of packets transmitted present heavy-tailed probability distributions, i.e., their decay is slower than the normal distribution.

Long-range Dependency (LRD) in network traffic is normally a result of adding several processes following a long-tail distribution, meaning that the auto-correlation function of current and past observations decays slowly. There are many connections among long-tails and Poisson processes [13]. For instance, this kind of processes offers a very simple explanation of long-range dependency being caused by long-tailed file sizes. Say many users are connected to a single server that processes work at constant rate r. At a Poisson time point, some user begins transmitting work to the server at constant rate which, for specificity, we take to be rate 1. The length of the transmission is random with long-tailed distribution. This example attests that modeling of long-tail traffic may generate accurate results over time, however, as we will see in the Section IV, good results do not depend necessarily of LRD, therefore reducing the needless overhead and amount of data that must be analyzed.

Algorithm 1 Poisson procedure

Input : Lambda parameter
Output: Poisson slices vector
1: **Start**
2: **procedure** POISSON($lambda$)
3: $vector\ vPoisson$
4: $var\ poissonSlice$
5: **for** $(i = (lambda); i > 0; i - -)$ **do**
6: $poissonSlice \leftarrow \frac{e^{-\lambda}(\lambda)^i}{i!}$
7: $vPoisson.add(poissonSlice)$
8: **end for**
9: **return** $vPoisson$
10: **end procedure**
11: **End**

Unlike LRD, in Short-range Dependency (SRD) processes, the coupling between values at different times decreases quickly as the time increases. Either the auto-covariance drops to zero after a certain time interval, or it eventually has an exponential decay. This proposal aims at exploring SRD as a process to reduce the amount of traffic that must be analyzed for traffic prediction, therefore being more suitable to dynamic environments such as cloud computing.

The Poisson distribution is a natural choice for describing the probability of the number of occurrences in a field or continuous interval (usually time or space), such as number of defects per square meter, number of accidents per day or number of network packets per minute. We note that in our study the unit of measure (time) is continuous, but the random variable (number of packets) is discrete. In other words, a Poisson process is used to determine the probable minimum and maximum number of transactions that can occur within a given time period, from a series of discrete values.

Let k be a discrete variable taking the values 0, 1, 2, 3, ... ,∞. If k represents a time interval following the Poisson process with parameter $\lambda > 0$, then:

$$P[N(t) = k] = \frac{e^{-\lambda t}(\lambda t)^k}{k!} \tag{1}$$

where the Poisson parameter lambda (λ) represents the total number of events (z) divided by the number of units (n) of data ($\lambda = z/n$). The unit forms the basis or denominator for calculation of the average. A Poisson process is described in Algorithm 1.

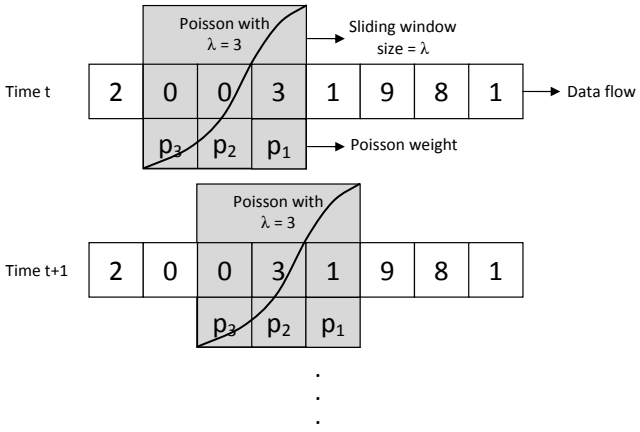

Fig. 2: The operation of sliding window

B. Poisson-based statistical prediction

In order to reduce the complexity of predicting network traffic, we consider time-bounded past information by means of a sliding window of size λ. This window is applied by weighting past observations according to a Poisson distribution with λ sampled values. The example illustrated in Figure 2 shows a sliding window with size three (i.e. $\lambda = 3$). Each value of the data flow is weighted with a portion of the Poisson process, and the most recent value of the data flow receives the highest Poisson slice/weight. Thus, at time t, the sliding window has a set of three values $\{0, 0, 3\}$. In the next turn, at time $t + 1$ the next value to enter inside the window will be 1, and when this occurs the oldest value (0) leaves the sliding window. This process will be repeated as long as there is a data flow from the network.

The Poisson distribution is represented by a discrete function observed over time t, the distribution starts at t = 0 and, when t = λ the function has its maximum. For a time interval of size t = λ, let a truncated Poisson distribution of size n be represented by values $p_1, p_2, \ldots, p_n$. To determine a prediction of the expected value of network traffic at time t, $\tilde{y}_t$, our solution uses a Poisson distribution truncated from t = 0 to t = λ. Then, we weight previous values according to the Poisson distribution as follows,

$$\tilde{y}_t = p_1 \times y_{t-1} + p_2 \times y_{t-2} + \ldots + p_\lambda \times y_{t-\lambda} \qquad (2)$$

where the $\tilde{y}_t$ represents the result of the prediction process, namely, the next expected value of the network traffic at time t.

Algorithm 2 Pseudocode for predicting network traffic

Input : Trace from Dropbox
Output: Prediction of network traffic

```
1:  Start
2:      read dbTrace
3:      read lambda
4:      vector vPoisson
5:      vector vPrediction
6:      vPoisson ← poisson(lambda)
7:      for (i = 0; i < dbTrace.size(); i++) do
8:          var nextValue ← 0
9:          for (j = 0; j <= lambda; j++) do
10:             if (i − j >= 0) then
11:                 var tmp ← (vPoisson[j] ∗ dbTrace[i − j])
12:             end if
13:             nextValue ← (nextValue + tmp)
14:         end for
15:         vPrediction.add(nextValue)
16:     end for
17: End
```

Algorithm 2 shows the procedure for predicting traffic. The input (i.e. the data flow in Figure 2) corresponds to the Dropbox trace (*dbTrace*). It is important to remember that the input data could be any other set of cloud data, without necessarily being derived from the monitoring of the Dropbox. Furthermore, we also have to set up the lambda size as illustrated at line 3 of Algorithm 2. This parameter defines the number of Poisson slices to be considered. This is equivalent to the number of samples of data used for the calculation of the prediction. Once we have the vector with the Poisson slices properly shaped, based on the Poisson Process of Algorithm 1, the algorithm estimates the next value for the network traffic for each new value from dataset (line 13, *nextValue*). As output, we have the vector containing the network traffic prediction, represented by *vPrediction*.

IV. Evaluation and Discussion

In this section we perform an evaluation of our proposal applied to a real trace of Dropbox data from [14].

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).

The work described in [14] presents four datasets: Campus 1, Campus 2, Home 1 and Home 2. During the data analysis we observed that the Campus 1, Home 1 and Home 2 datasets exhibit, at most times, a low number of network packets. Campus 2 instead, shows a larger traffic volume compared with the others datasets, therefore being our choice for evaluation.

For the evaluation of our solution, we consider the traffic collected from two university campuses and two points of presence, namely the Campus 2 dataset (see [14]). The dataset was divided in intervals of five minutes each, and the evaluation was performed by applying a sliding window weighted with a Poisson distribution over the raw data.

As previously discussed, the Poisson parameter λ represents the mean of events of the Poisson process. The Poisson process has the interesting property that the expected value (mean of the distribution) is close to the variance of the distribution [15]. For this reason, we have performed the evaluation focused in two sliding window sizes: the mean and the variance of the dataset. In addition, we also have assessed the prediction algorithm with λ parameter equal to standard deviation (i.e. the nearest integer value found in the statistics of Dropbox dataset). In order to make a fair comparison, the tests with other predictors assessed in this work have also used a window with the same size.

The effectiveness of the prediction mechanisms is measured through the Normalized Mean Square Error (NMSE) [16],

$$NMSE = \frac{1}{\sigma^2}\frac{1}{N}\sum_{t=1}^{N}\left(X_t - \hat{X}_t\right)^2 \qquad (3)$$

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.

We also consider the Pearson correlation, a measure of correlation that represents the linear dependence between two variables (e.g. X and Y), giving a value between $[-1; 1]$. This is widely used as a measure of the strength of linear dependence between two variables. A value of 1 implies that a linear equation perfectly describes the relationship between

TABLE II: Descriptive Statistics

Dataset		Mean			Std. Deviation	Variance	NMSE
		Arithmetic	Square Error	Std. Error			
1	Dropbox	19.379	0.000	0.273	30.445	926.901	0.000
2	Trivial	19.379	926.902	0.000	0.000	0.000	1.000
Approach		**Sliding window size arithmetic mean**					
3	Poisson	19.376	41.168	0.267	29.726	883.662	0.044
4	WMA	19.380	82.523	0.263	29.326	860.046	0.089
5	SMA	19.379	117.072	0.262	29.233	854.596	0.126
Approach		**Sliding window size standard deviation**					
6	Poisson	19.375	49.590	0.266	29.630	877.966	0.054
7	WMA	19.381	127.845	0.259	28.853	832.490	0.137
8	SMA	19.380	196.316	0.257	28.638	820.143	0.211
Approach		**Sliding window size variance**					
9	Poisson	19.383	103.639	0.260	29.017	842.027	0.112
10	WMA	19.384	588.133	0.205	22.901	524.457	0.635
11	SMA	19.369	935.342	0.188	20.923	437.777	1.009

X and Y, with all data points lying on a line for which Y increases as X increases. A value of -1 means that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables.

B. Results and Discussion

In Table II, we present statistics about the dataset used and results for several prediction methods. 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 our prediction mechanism (Poisson) as well as two others (SMA and WMA) described in Section II. For each set of predictors, results are provided for three different values of λ: mean, standard deviation and variance.

While for most metrics the results of the remaining predictors are not far from those obtained by our Poisson approach, we highlight the results achieved by our approach in terms of NMSE, where our approach 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. Table III attests that the Poisson predictor has the strongest correlation among the predictors assessed in this work.

The assessment was compiled from 42 consecutive days of monitoring. However, a small demonstration of the behavior of different predictions performed with the traffic dataset containing information from Dropbox is illustrated in Figures 3, 4, 5 and 6 (all these figures have the average as sliding window size). Due limited space and better viewing of the results, we only provide predictions for a limited time period. However, the observable match between real values and predictions held for remaining time periods.

TABLE III: Pearson Correlation

	Dropbox	Poisson	WMA	SMA	Trivial
Dropbox	1,00	0,924	0,903	0,729	-

It is worth mentioning that the best results were obtained for lower values of λ. This happens because the average of all elements of the dataset is around 19.38, which is aligned with the property of the Poisson process of the expected value being close to the mean of the distribution [15]. In particular,

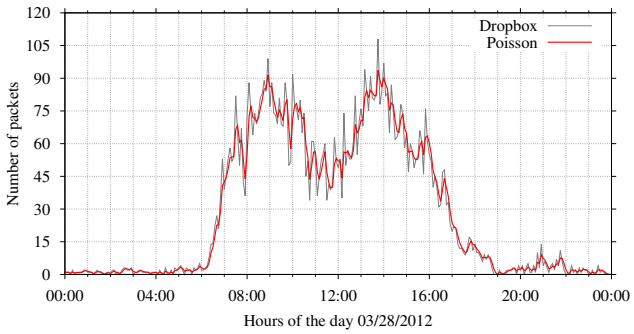

Fig. 3: Poisson sample of prediction

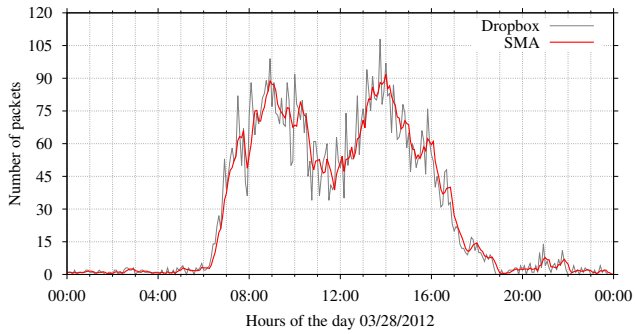

Fig. 4: SMA sample of prediction

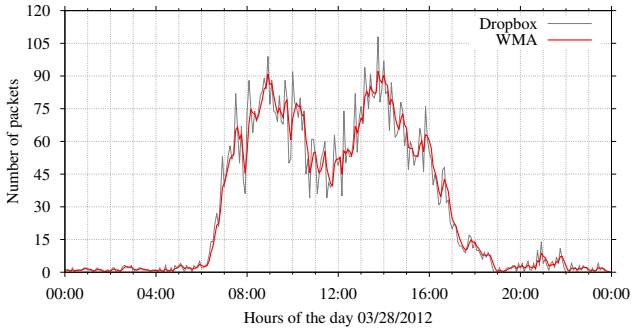

Fig. 5: WMA sample of prediction

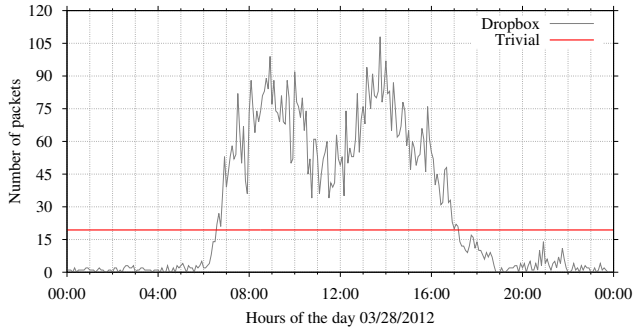

Fig. 6: Trivial predictor sample of prediction

the prediction improves as λ approaches the average. With a smaller sliding window, oldest values also have fewer influence on the predicted network traffic. This indicates that a predictor that prioritizes recent history is better suited to the dynamics of cloud computing environments, by resorting to short-range dependency. These results have shown that our solution is able to provide accurate predictions with relatively low levels of historical data dependency.

V. Conclusions

In this paper we have proposed a statistical approach for predicting network traffic in cloud computing environments. Taking advantage of well-know network traffic features such as short-range dependency, our model resorts to a Poisson distribution within a sliding window for weighting past observations. Our results have shown compliance with real data traces obtained for Dropbox. In addition, our approach resulted in accurate prediction with low levels of historical data dependency and compares favorably with other predictors. In particular, we were able to achieve the lowest values of Normalized Mean Square Error as well as the strongest Pearson correlation between the real values and predictions. Prospective directions for future work include considering an approach based on a dynamic sliding window, and using this methodology to perform anomaly detection of network traffic in virtual environments.

Acknowledgment

This work was partially funded by the project CMU-PT/RNQ/0015/2009, TRONE - Trustworthy and Resilient Operations in a Network Environment; the iCIS project, under the grant CENTRO-07-ST24-FEDER-002003; and CAPES and CNPq (Brazil) through the Ciência sem Fronteiras Program/2013.

References

[1] E. M. Gilmer, "Is There a Silver Lining for the Environment in Cloud Computing?" *The New York Times*, vol. 10 August, 2011.

[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[3] 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.

[4] W. Yang, D. Yang, Y. Zhao, and J. Gong, "Traffic flow prediction based on wavelet transform and radial basis function network," in *2010 International Conference on Logistics Systems and Intelligent Management. Harbin, China*, vol. 2, 2010, pp. 969–972.

[5] 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.

[6] A. Yaacob, I. K. T. Tan, S. F. Chien, and H. K. Tan, "Arima based network anomaly detection," in *Second International Conference on Communication Software and Networks, 2010. ICCSN '10. Singapore*, Feb. 2010, pp. 205–209.

[7] M. Li and S. Lim, "Modeling network traffic using generalized cauchy process," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 11, pp. 2584–2594, 2008.

[8] A. Dainotti, W. De Donato, A. Pescape, and P. Salvo Rossi, "Classification of network traffic via packet-level hidden markov models," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. New Orleans, Louisiana*, 2008, pp. 1–5.

[9] B. Chang and H. Tsai, "Improving network traffic analysis by foreseeing data-packet-flow with hybrid fuzzy-based model prediction," *Expert Systems with Applications. Tarrytown, NY, USA*, vol. 36, no. 3, pp. 6960–6965, 2009.

[10] W. Lee, C. Chen, K. Chen, T. Chen, and C. Liu, "A comparative study on the forecast of fresh food sales using logistic regression, moving average and bpnn methods," *Journal of Marine Science and Technology*, vol. 20, no. 2, pp. 142–152, 2012.

[11] 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.

[12] P. Yan-hui and W. Tao, "Network traffic emulation based on representative network behavior and protocol," in *1st International Conference on Information Science and Engineering (ICISE)*. IEEE, 2009. Jiangsu, China, pp. 1777–1780.

[13] S. Resnick, *Heavy-tail phenomena: probabilistic and statistical modeling*. Springer, 2006.

[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. C. Cameron and P. K. Trivedi, "Regression-based tests for overdispersion in the poisson model," *Journal of Econometrics*, vol. 46, no. 3, pp. 347–364, 1990.

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