ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 3, May2012

Statistical Inference and α-Stable Modeling for Anomaly Detection in Network Traffic

Ravindra Kumar Gupta
Asistant Professor
SSSIST
Sehore (M.P) ,India

Gajendra Singh Chandel
Asistant Professor
SSSIST
Sehore (M.P) ,India

Vijay D. Rughwani
(M.Tech in IT Pursuing)
SSSIST
Sehore (M.P),India

ABSTRACT

ANOMALY detection aims at finding the presence of anomalous patterns in network traffic. Automatic detection of such patterns can provide network administrators with an additional source of information to diagnose network behavior or finding the root cause of network faults. However, as of today, a commonly accepted procedure to decide whether a given traffic trace includes anomalous patterns is not available. Indeed, several approaches to this problem have been reported in the literature. Research proposals in anomaly detection typically follow a four-stage approach, in which the first three stages define the detection method, while the last stage is dedicated to validate the approach. So, in the first stage, traffic data are collected from the network (data acquisition). Second, data are analyzed to extract its most relevant features (data analysis). Third, traffic is classified as normal1 or abnormal (inference); and fourth, the whole approach is validated with various types of traffic anomalies (validation).This project paper aims in detecting two anomaly namely flood & flash crowd anomaly using statistical inference & α-Stable modeling.

Keywords:
Traffic analysis, anomaly detection, statistical models, hypothesis testing, network performance, network reliability.

I INTRODUCTION

We are drowning in the deluge of data that are being collected world-wide, while starving for knowledge at the same time. Anomalous events occur relatively infrequently. However, when they do occur, their consequences can be quite dramatic and quite often in a negative sense. Automatic detection of such patterns can provide network administrators with an additional source of information to diagnose network behavior or finding the root cause of network faults. Research proposals in anomaly detection typically follow a four-stage approach, in which the first three stages define the detection method, while the last stage is dedicated to validate the approach. So, in the first stage, traffic data are collected from the network (data acquisition). Second, data are analyzed to extract its most relevant features (data analysis). Third, traffic is classified as normal1 or abnormal (inference); and fourth, the whole approach is validated with various types of traffic anomalies (validation). In this regard, flood and flash crowd anomalies are of interest to several anomaly detection contributors.

Following the aforementioned four-stage approach, we can mention that data acquisition is typically carried out by polling one or more routers periodically, so that traffic data are collected and stored for posterior analysis in the second stage. Some authors sample data at the packet level, gathering information from headers, latencies, etc., while others prefer to use aggregated traffic as the source of information, often through the use of the Simple Network Management Protocol (SNMP). Sampling data at the packet level provides more information, but at the cost of a higher computational load and dedicated hardware must be employed. Aggregated traffic, on the other hand, gives less information from which to decide for the presence or absence of anomalies, but is a simpler approach and does not need any special hardware. Apart from this dichotomy, however, there seems to be a consensus on how to proceed in this stage.

In the data analysis phase, several techniques can be applied to extract interesting features from current traffic. Some of them include information theory, wavelets, statistics-based measurements , and statistical models . Of these techniques, the use of statistical models as a means to extract significant features for data analysis has been found to be very promising, since they allow for a robust analysis even with small sample sizes (provided that the model is adequate for real data). Moreover, with a traffic model, its set of parameters can be used as extracted traffic features, since any traffic sample is determined by the model parameters.

Existing traffic models range from the classical Poisson model, first introduced for packet networks by Kleinrock , to

most recent models, which state the importance of high variability and long-range dependence in modeling network traffic. Nevertheless, anomaly detection is still often based (at least partially) on classical models, such as Gamma distributions . The fact that these models do not account for high variability may have a negative impact on capturing traffic properties and, as a consequence, on detecting anomalies. High variability manifests itself in the marginal (first-order) traffic distribution and states that traffic is inherently bursty. This results in traffic distributions exhibiting heavy tails which cannot be properly modeled with, e.g., Gaussian functions. Long-range dependence, on the other hand, states that traffic is highly dependent over a wide range of time scales, i.e., its autocorrelation function exhibits a slowly decaying tail.

Several approaches have been used in the inference stage as well. Classification methods based on neural networks , statistical tests, information theory, and simple thresholding, to cite a few, can be found in anomaly detection literature. There seems to be a common point in all of them, though. The inference stage bases its decisions on the existence of a reference traffic window, which allows the classification method to assess whether the current traffic window is normal (i.e., it is sufficiently similar to the reference window) or abnormal (i.e., significantly different from the reference window). How the reference window is chosen not only has an impact on the final normal versus abnormal classification rate, but it also determines the exact definition of a traffic anomaly. Some approaches assume that an anomaly is an abrupt change in some of the features extracted from traffic, so the reference window is simply the previous-to-current traffic window.

In the validation stage, researchers give quality measures about the detection capability of their method according to a chosen criterion, which is typically the detection rate in terms of false positives and false negatives (i.e., the fraction of normal traffic patterns incorrectly classified as anomalous and the fraction of anomalous traffic patterns incorrectly classified as normal, respectively), although some researchers prefer other quality measures.

II DATA ACQUISITION

In the data acquisition stage, traffic samples are taken at intervals of t seconds, so that data windows of W seconds are continuously filled and passed to the second stage. W should be large enough to have a minimum amount of data when trying to fit a statistical model to them, and short enough to have (at least) local stationarity. This is necessary since we extract a single set of parameters from each time window, which we assume to be constant for W seconds. Traffic stationarity where we find one-hour periods to be reasonably stationary, so, we make use of this assumption. However, in order to ensure the model adequately fits the data we chose a time window length W = 30 minutes. t, on the other hand, should be short enough to, again, provide as many traffic samples as possible to the second stage, but we must also keep in mind that the shorter the t, the more loaded a router will be. Network managers often find unacceptable for a router to spend any significant amount of time in monitoring tasks.

III DATA ANALYSIS

The use of statistical models in the data analysis stage can be advantageous since an adequate model allows for a robust analysis even with small sample sizes. With traffic windows of W=t ¼ 360 samples each, our sample size is rather small, so the use of a model is desirable. This approach has been previously used in various works; however, the model used there does not account for important traffic properties, such as high variability. To deal with the problem, proposes the use of α-stable distributions, unrestricted in their parameter space, as a model for traffic marginal's.

IV INFERENCE

In order to detect anomalies, we first need to define what anomaly is or, in other words, what our method tries to detect. A common approach is to define anomalies as a sudden change in any quantity measured in the previous stage or a significant divergence between the current traffic window and a previously chosen reference . Note the difference between both strategies: the former compares current traffic to recent past traffic, while the latter does not assume recent traffic is necessarily normal. We feel this latter approach is superior since some types of anomalies are detectable this way but would not be otherwise (for instance, slow trends). However, how superior this latter approach is depends directly on the ability of the reference window to represent all kinds of normal traffic at any given circumstance. It is widely known that network traffic exhibits a cycle stationary behavior with periods of days and weeks , generally speaking, traffic patterns that are clearly anomalous in some network, at a given time, can be perfectly normal in some other network or time instant. Thus, the reference window should vary from a router port to another, and from any hour-weekday combination to any other for the anomaly detection system to succeed in real world. Possibly, holidays and other workday interruption periods should also be taken into account. Still, having exactly one reference window for all possible combinations of port, hour, and weekday needs the intervention of an expert who can tell normal traffic apart from anomalous.

V FLOODING & FLASH CROWD ANOMALY IN NETWORK TRAFFIC

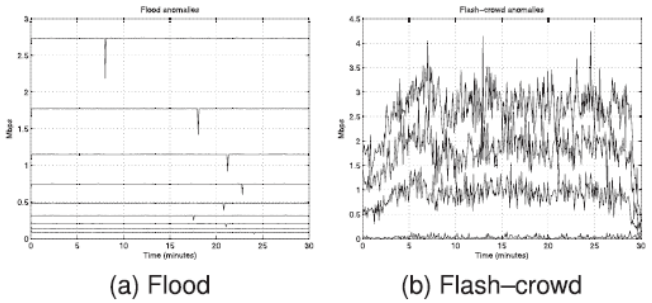

Fig.1. Anomalous patterns for flood and flash-crowd anomalies.

Flood anomalies include attacks, or any other circumstances, which result in a net growth of instantaneous traffic. One can think of flood anomalies as having one or more relatively constant traffic sources added to otherwise normal traffic. DDoS attacks typically give rise to anomalies of this kind. The most common attack is the Denial of Service (DoS) attack. In flooding attacks, a malicious user (or users) sends a large number of SIP messages that overload the SIP server1 and this subsequently.

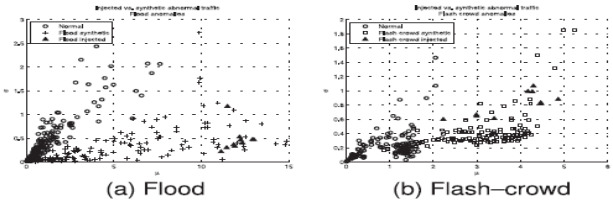

Fig. 2. Distribution of injected versus synthetic anomalous patterns

One of the primary tasks of network administrators is monitoring routers and switches for anomalous traffic behavior such as outages, configuration changes, flash crowds and abuse. Recognizing and identifying anomalous behavior is often based on ad hoc methods developed from years of experience in managing networks. A variety of commercial and open source tools have been developed to assist in this process, however these require policies and/or or thresholds to be defined by the user in order to trigger alerts. The better the description of the anomalous behavior, the more effective these tools become.

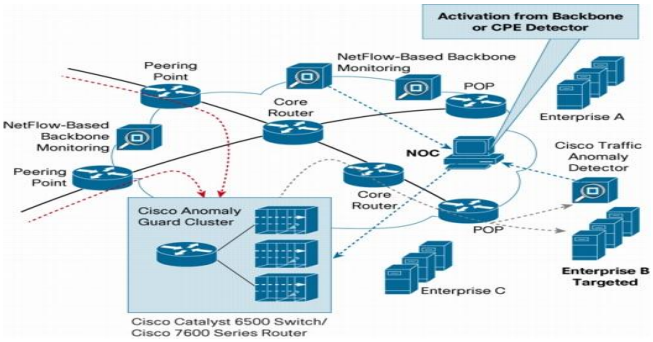

Fig 3. DDoS (Flood) Anomaly Detection

Flash-crowd anomalies encompass traffic patterns which are caused by a net growth of (usually human) users trying to access a network resource. Typical flash-crowd anomalies are related to overwhelming web server usage patterns. Once we have these two pools of "pure anomalies," we randomly choose as many of them as existing normal windows in each port-hour-weekday combination and add them together. This results in three sets of traffic windows for each port, hour, and weekday: a normal set, a flood anomalous set, and a flash-crowd-anomalous set. As stated above, normal windows consist of strictly real traffic, while anomalous ones are synthetic (although they have also been built from real traffic).

VI EXISTING NETWORK TRAFFIC MODELS

Traditionally, network traffic has been modeled as a Poisson process for historical reasons. Indeed, the Poisson model has been successfully used in telephone networks for many years, and so it was inherited when telecommunication networks became digital and started to send information as data packets. Also, this model has a simple mathematical expression , and has only one parameter, which is in turn very intuitive (the mean traffic in packets per time unit).

In the last decade, however, several authors have studied network traffic behavior and proposed other models that overcome the limitations which are inherent to Poisson processes, the most notable ones probably being that the Poisson model has a fixed relationship between mean and variance values (both are equal to λ), and that it does not account for high variability or long-range dependence.

More recently proposed models are usually based on the assumption that network traffic is self-similar in nature, as originally stated. Intuitively, network traffic can be thought of as a self-similar process because it is usually "bursty" in nature and this burstiness tends to appear independently of the time scale. Thus, in Fractional Brownian Motion (FBM) is shown to properly fit ccumulated network traffic data (note that FBM is an autoregressive process and so it can model accumulated traffic, but not instantaneous one), but the

authors impose a strict condition: analyzed traffic must be very aggregated for the model to work, that is, the FBM model is only valid, authors say, when many traffic traces are aggregated, in such a way that the number of aggregated traces is much larger than the length of a single trace (measured in number of traffic samples).

Let us consider why this restriction is necessary. First of all, we used our collected data to try and see if this constraint was needed in our particular network, and saw that it was indeed the case. Note that there are some traffic peaks, or "bursts," scattered among the data, which otherwise tend to vary in a slower fashion. Recalling that instantaneous contributions to FBM are Gaussian random variables, we can calculate a histogram of traffic data like , which show typical cases of instantaneous traffic distribution in routers 1 and 2, along with Poisson, Gaussian, Gamma, and α-stable curves fitted to real data.

Poisson, Gaussian, and Gamma curves were fitted using maximum Likelihood (ML) algorithm, while the α-stable curve was fitted with the algorithm described in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2011.14. Clearly, one can see the marginal distribution of sampled data differs considerably from Poisson, Gaussian, and Gamma probability density functions (PDFs), especially in the case of. This happens due to the extreme values present in the data, which alter mean and variance estimates considerably.

All of this means that a single traffic trace cannot be modeled as an FBM because traffic marginals are not Gaussian. However, once many traffic traces are aggregated, the resulting data do follow a Gaussian distribution, and so, the FBM model is valid. This happens as a consequence of the Central Limit Theorem which loosely states that the sum of many identically distributed random variables converges to a Gaussian distribution. Note, however, that FBM can model the self-similarity properties of traffic, i.e., it includes a time evolution model which accounts for the long-range dependence that data usually exhibit.

On the other hand, the first aforementioned property states the obvious fact that network traffic has compact support between 0 and the M. Compact support makes symmetric distributions (Gaussian distributions are symmetric) inappropriate, because if the traffic histogram concentrates on very low transmission rates, the model would allow negative traffic increments to occur with a non-negligible probability and this can never be the case. Accordingly, if traffic data concentrate near the maximum transmission rate, a symmetric model would allow traffic increments to be larger than physically possible, again,. with a non-negligible probability. This also affects the Gammadistribution, since its tail always lies on its right side

V α-STABLE MODEL

α -stable distributions can be thought of as a superset of Gaussian functions and originate as the solution to the Central Limit Theorem when second order moments do not exist, that is, when data can suddenly change by huge amounts as time passes by. This fits nicely to the high variability property seen in network traffic (the Noah effect). Moreover, **α** -stable distributions have an asymmetry parameter which allows their PDF to vary from totally left asymmetric to totally right-asymmetric (this is almost the case in Fig.4), while genuine Gaussian distributions are always symmetric. This parameter makes **α** -stable distributions naturally adaptable to the first traffic property (compact support) even when average traffic is virtually 0 or very near the maximum theoretical network throughput.

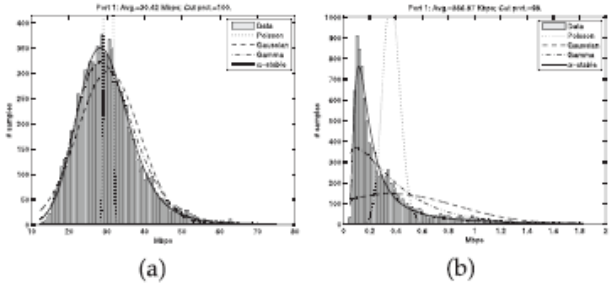

Fig. 4. A typical histogram of traffic passing through: (a) router 1 and (b) router 2 (10,000 samples each, taken in June 2010 and February 2011, respectively) along with Poisson (dotted), Gaussian (dashed), Gamma (dash-dot), and **α** -stable (solid) curves fitted to the data.

In addition, **α** -stable distributions give an explanation to the restriction imposed about the need to aggregate many traffic traces for them to converge to a Gaussian distribution. According to the Generalized Central Limit Theorem, which includes the infinite variance case, the sum of n **α** -stable distributions is another **α**-stable distribution, although not necessarily Gaussian. Since traffic data exhibit the Noah effect, we can assume infinite variance. Then, under the hypothesis that marginal's are **α**-stable, the sum of a few traces will be **α** -stable but not necessarily Gaussian. However, after summing sufficiently many traces, the final histogram converges to a Gaussian curve. This occurs because any real measurement cannot be infinite, even if an infinite variance model proves to reflect reality best.

α-stable distributions are characterized by four parameters. The first two of them, **α** and β, provide the aforementioned properties of heavy tails (**α**) and asymmetry (β), while the remaining two, σ and μ, have analogous meanings to their counterparts in Gaussian functions (standard deviation and mean, respectively). Note that, while they have analogous senses (scatter and center), they are not equivalent because **α** -stable distributions do not have, in general, a finite mean or variance, i.e., $E\{X\} \neq \mu$ and $STD\{X\} \neq \sigma$. The allowed values for **α** lie in the interval {0; 2}, being **α** = 2 the Gaussian case, while β must lie inside [-1; 1], -1

means totally left-asymmetric and 1totally right-asymmetric). The scatter parameter (σ) must be a nonzero positive number and μ can have any real value.
Despite their potential advantages, however, we also state some reasons why **α** -stable distributions are difficult to use. First, the absence of mean and variance in the general case makes impossible the use of many traditional statistical tools. Moreover, as mentioned before, these distributions do not have (to the best of our knowledge) a known closed analytical form to express their PDF nor their CDF, so powerful numerical methods are needed for tasks which are almost trivial with (for example) the Gaussian distribution, such as estimating their parameters for a given data set or even drawing a PDF. Also, the fact that they have four parameters, instead of just two, introduces two new dimensions to the problem, which can make processing times grow faster than in the Gaussian approach. In our experiments, however, this is not an issue with recent hardware. The Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2011.14 describes the mathematical methods we used in dealing with **α** -stable distributions.

VI IMPLEMENTATION

This project is implemented in NS2. NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkelywritten in C++ and OTcl. NS is primarily useful for simulating local and wide area networks. Although NS is fairly easy to use once you get to know the simulator, it is quite difficult for a first time user, because there are few user-friendly manuals. Even though there is a lot of documentation written by the developers which has in depth explanation of the simulator, it is written with the depth of a skilled NS user. **Tcl** is a general purpose scripting language. While it can do anything other languages could possibly do, its integration with other languages has proven even more powerful.

```
Agent/MyAgent set myVal 0
Agent/MyAgent set bottle_neck 10; #set a bottle neck for the transmissions
Agent/MyAgent set RTSEQ 4200; #set some value for RTSEQ
Agent/MyAgent set source00 $sources(0)
Agent/MyAgent set dest00 $dest(0)
Agent/MyAgent set source01 $sources(1)
Agent/MyAgent set dest01 $dest(1)
Agent/MyAgent set source02 $sources(2)
Agent/MyAgent set dest02 $dest(2)
Agent/MyAgent set source03 $sources(3)
Agent/MyAgent set dest03 $dest(3)
Agent/MyAgent set source04 $sources(4)
Agent/MyAgent set dest04 $dest(4)
Agent/MyAgent set source05 $sources(5)
Agent/MyAgent set dest05 $dest(5)
Agent/MyAgent set source06 $sources(6)
Agent/MyAgent set dest06 $dest(6)
Agent/MyAgent set source07 $sources(7)
Agent/MyAgent set dest07 $dest(7)
Agent/MyAgent set source08 $sources(8)
```

Using Statistical Inference we statistically indicate source & destination. Here we declare source node 1,2---& destination source 2,3,4---.Source 1 will send packet to destination 2 only & so on. If any abnormal activity occurs just like source 1 sends packet to destination 7, then anomaly is detected. Predefined transmission is laid down.

VII OUTPUT

```
-2.34/tk8.4.18/unix:$PATH
[root@localhost Desktop]# ns anomaly.tcl
num_nodes is set 10
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
Source Address:5, Dest Address:6
No anomaly found, so processing the packet
I 6 received packet at:5.0004
Delay is: 0.000398637
Source Address:6, Dest Address:1
Anomaly Detected, so the packet would not be sent
Source Address:6, Dest Address:3
Anomaly Detected, so the packet would not be sent
Source Address:6, Dest Address:7
No anomaly found, so processing the packet
I 7 received packet at:5.00286
Delay is: 0.00246267[root@localhost ~]# ls
anaconda-ks.cfg  install.log.syslog      Public
anomaly.tcl      Music                   Templates
Desktop          ns-2.34.sh              Videos
Documents        ns-allinone-2.34        xgraph-12.1
Downloads        ns-allinone-2.34.tar.gz  xgraph-12.1.tar.gz
install.log      Pictures
[root@localhost ~]# cd Desktop/
[root@localhost Desktop]# PATH=/root/Desktop/ns-allinone-2.34/bin:/root/Desktop/ns-allinone-2.34/tcl8.4.18/unix:/root/Desktop/ns-allinone
```

VII LIMITATION OF STUDY

α-stable distributions outperform other statistical distributions used elsewhere in anomaly detection, but at the cost of a higher computational cost. Although this extra calculation time should not be an issue in current hardware, classical models will always be easier and faster to use. On the other

hand, a set of reference traffic windows combined with synthetic anomalies reduces the need for a network manager's intervention, but a sufficiently large traffic data collection, able to represent normal traffic at any moment, must be available prior to deploying our detection method.

IX CONCLUSION & FUTURE WORK

In this project paper, detection of flood & flash crowd anomaly in WSN is done by using statistical inference & α – stable model. We follow a four-stage approach to describe each of the pieces from which to build a functional detection system (data acquisition, data analysis, inference, and validation), yielding the final classification results. Our approach uses aggregated traffic as opposed to packet-level sampling so dedicated hardware is not needed. Despite the mentioned contributions, our approach still has some drawbacks. α -stable distributions outperform other statistical distributions used elsewhere in anomaly detection, but at the cost of a higher computational cost.

Although this extra calculation time should not be an issue in current hardware, classical models will always be easier and faster to use. On the other hand, a set of reference traffic windows combined with synthetic anomalies reduces the need for a network manager's intervention, but a sufficiently large traffic data collection, able to represent normal traffic at any moment, must be available prior to deploying our detection method.

Further work in this subject falls in three main areas. First, the proposed model for network traffic is not complete, since we model traffic marginals only. As recent literature shows, traffic correlations should be taken into account if new insights on traffic nature are to be found, so clearly this is open ground for a deeper study in the Data analysis stage. Although the inclusion of a time evolution model adds complexity and computational load to traffic analysis, the long-range dependence property may provide additional information which could prove useful for the inference stage. Also, the GLRT has been chosen in the inference stage since, being a parametric classifier, it is able to take advantage of the traffic model robustness, as well as for its asymptotical UMP characteristics. Nevertheless, other classifiers may prove able to yield better results or reduced calculation times. Second, our method and that reported in differ in quite a few areas, so, there is the open question of exactly how much every difference contributes to the final results. Some of these differences are difficult to measure (e.g., the robustness of a set of normal and anomalous reference windows for all port, hour, and weekday combinations versus the use of a single, normal traffic reference window) but, still, studying the contribution of every single variable seems necessary to further enhance performance figures. And third, the methods proposed in this paper have been tested in conditions, so more testing in production environments shall be carried out, so as to further understand network managers' needs.

X REFERENCES

[1] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry,"Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 1, pp. 56-70, Jan. 2007.
[2] M. Thottan and C. Ji, "Anomaly Detection in IP Networks," IEEE Trans. Signal Processing, vol. 51, no. 8, pp. 2191-2204, Aug. 2003.
[3] C. Manikopoulos and S. Papavassiliou, "Network Intrusion and Fault Detection: A Statistical Anomaly Approach," IEEE Comm. Magazine, vol. 40, no. 10, pp. 76-82, Oct. 2002.
[4] Y. Gu, A. McCallum, and D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," Proc. Internet Measurement Conf., Oct. 2005.
[5] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," Proc. ACM SIGCOMM '04, pp. 219-230, Aug. 2005.
[6] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," Proc. Second ACM SIGCOMM Workshop Internet Measurement, pp. 71-82, Nov. 2002.
[7] A. Ray, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," Signal Processing, vol. 84, no. 7, pp. 1115- 1130, 2004.
[8] S.C. Chin, A. Ray, and V. Rajagopalan, "Symbolic Time Series Analysis for Anomaly Detection: A Comparative Evaluation," Signal Processing, vol. 85, no. 9, pp. 1859-1868, 2005.
[9] A. Wagner and B. Plattner, "Entropy Based Worm and Anomaly Detection in Fast IP Networks," Proc. 14th IEEE Int'l Workshops Enabling Technologies: Infrastructures for Collaborative Enterprises, pp. 172-177, June 2005.
[10] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting Anomalous Network Traffic with Self-Organizing Maps," Proc. Sixth Int'l Symp. Recent Advances in Intrusion Detection, pp. 36-54, 2003.
[11] S.T. Sarasamma, Q.A. Zhu, and J. Huff, "Hierarchical Kohonen Net for Anomaly Detection in Network Security," IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics, vol. 35, no. 2, pp. 302-312, Apr. 2005.
[12] V. Alarcon-Aquino and J.A. Barria, "Anomaly Detection in Communication Networks Using Wavelets," IEE Proc.—Comm., vol. 148, no. 6, pp. 355-362, Dec. 2001.
[13] L. Kleinrock, Queueing Systems, Volume 2: Computer Applications.John Wiley and Sons, 1976.
[14] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson, "Self-

Similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," IEEE/ACM Trans. Networking, vol. 5, no. 1, pp. 71-86, Feb. 1997.
[15] G. Samorodnitsky and M.S. Taqqu, Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance. Chapman & Hall, 1994.
[16] F. Simmross-Wattenberg, A. Trista´n-Vega, P. Casaseca-de-la Higuera, J.I. Asensio-Pe´rez, M. Martı´n-Ferna´ndez, Y.A. Dimitriadis, and C. Alberola-Lo´pez, "Modelling Network Traffic as _-Stable Stochastic Processes: An Approach Towards AnomalyDetection," Proc. VII Jornadas de Ingenierı´a Telema´tica (JITEL), pp. 25-32, Sept. 2008.
[17] G.R. Arce, Nonlinear Signal Processing: A Statistical Approach. John Wiley and Sons, 2005.
[18] J. Jiang and S. Papavassiliou, "Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction," J. Network and Systems Management, vol. 12, no. 1, pp. 51-72, Mar. 2004.
[19] W. Yan, E. Hou, and N. Ansari, "Anomaly Detection and Traffic Shaping under Self-Similar Aggregated Traffic in Optical Switched Networks," Proc. Int'l Conf. Comm. Technology (ICCT '03), vol. 1, pp. 378-381, Apr. 2003.
[20] J. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring," Proc. USENIX 14th System Administration Conf. (LISA), pp. 139-146, Dec. 2000.
[21] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modelling," IEEE/ACM Trans. Networking, vol. 3, no. 3, pp. 226- 244, June 1995.
[22] Internet Traffic Archive, http://ita.ee.lbl.gov/, 2011.
[23] Waikato Internet Traffic Storage, http://wand.cs.waikato.ac.nz/ wits/, 2011.
[24] Cooperative Assoc. for Internet Data Analysis, http://www.caida.org/, 2011.
[25] DiRT Group's Home Page, Univ. of North Carolina, http://wwwdirt.cs.unc.edu/ts/, 2010.
[26] "Metrology for Security and Quality of Service," http:// www.laas.fr/METROSEC/, 2011.
[27] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-Based Change Detection: Methods, Evaluation, and Applications," Proc. Internet Measurement Conf. (IMC), pp. 234-247, Oct. 2003.
[28] DDoSVax, http://www.tik.ee.ethz.ch/ddosvax/, 2010.
[29] S. Stolfo et al., "The Third International Knowledge Discovery and Data Mining Tools Competition," http://kdd.ics.uci.edu/ databases/kddcup99/kddcup99.html, 2011.
[30] G. Cormode and S. Muthukrishnan, "What's New: Finding Significant Differences in Network Data Streams," IEEE/ACM Trans. Networking, vol. 13, no. 6, pp. 1219-1232, Dec. 2005.
[31] Cisco Systems, "Cisco IOS NetFlow," http://www.cisco.com/ web/go/netflow, 2011.
[32] A. Papoulis, Probability, Random Variables, and Stochastic Processes, third ed., McGraw-Hill, 1991.
[33] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self- Similar Nature of Ethernet Traffic (Extended Version)," IEEE/ ACM Trans. Networking, vol. 2, no. 1, pp. 1-15, Feb. 1994.
[34] P. Embrechts and M. Maejima, Selfsimilar Processes. PrincetonUni