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

To Detect and Prevent the anomaly in Network Traffic Based on Statistical approach and α-stable Model

[1]Anup Bhange
[1]M.tech Scholar, Dept CSE
[1] Patel Institute of Technology
Bhopal

[2] Amber Syed
[2]Asst.Prof, Dept CSE
[2]Patel Institute of Technology
Bhopal

Abstract:--
Detecting network traffic anomalies is crucial for network operators as it helps to identify security incidents and to supervise the availability of networked services. Although anomaly recognition has received significant attention in the literature, the automatic classification of network anomalies still remains an open problem. Network administration and security is now one of the most energetic examines areas, among which, research on perceive and identifying anomalies has attracted a lot of interest. Researchers are still worried to find a useful and lightweight method for anomaly recognition purpose. In this Paper our discussion on an approach for anomaly recognition.
With novel types of attacks come into view frequently, rising bendable and adaptive security oriented approaches is a severe challenge. Also discussed anomaly detection in network traffic which may consist with to detect anomaly namely flood and flash crowd. Using Techniques are to protect target systems and networks against malicious activities.
In this 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 method to detect anomalies in network traffic, based on a non restricted α -stable first-order model and statistical hypothesis testing. Here we focus on detecting and preventing two anomaly types, namely floods and flash-crowd .Here we use NS2 simulator to calculate above result.

Keywords: Statistical Approach, α-Stable Distribution, Network traffic representation

I. INTRODUCTION:

For the period of the last years, the presence of cyber assault has grown radically throughout the Internet [1]. Consequently, the detection of anomalies on network traffic data has been studied extensively. Despite significant research efforts, anomaly recognition systems have not been yet broadly adopted by network operators mainly because: 1) they produce a large number of false positives; 2) they use complex detection method that are often unintelligible, i.e., a "black box", to network operator; and 3) they require learning a labeled "ground truth" dataset from the target network.

Internet traffic size is vital for monitoring trends, network planning and anomaly traffic detection. In general, simple packet- or byte-counting methods with SNMP have been widely used for easy and useful network administration. In count, the passive traffic amount approach that gathers and inspects packets at routers or dedicated machines is also popular. However, traffic measurement will be harder in the next-generation Internet with the features of high-speed links or new protocols such as IPv6 or MIPv6.

Traffic quantity at high speed links is challenging because of fast packet-processing requirement. Though packet-level measurement can explain the complete traffic characteristics, it is not easy to support high-speed line rates of multi-gigabit per second. Moreover, impartial systems for packet-level traffic monitoring will be luxurious for the wide deployment and easy management in a large-scale network. Hence, ISPs or big A Ses will generally prefer the flow-level traffic measurement advance that could be easily embedded into routers or switches to dedicated packet-level traffic monitoring systems. Currently, flow-level measurement modules at routers such as Cisco Net Flow [1] have become popular, because flow-level measurement could produce useful traffic statistics with a appreciably small amount of measured data.

Traffic anomalies such as attacks, flash crowds, large file transfers and outages occur fairly frequently in the Internet today. Large enterprise networks often have a security operations center where operators endlessly monitor the network traffic hoping to detect, identify and treat anomalies. In smaller networks, these tasks are carried out by general network administrators who are also carry out other day-to-day network maintenance and planning activities. Despite the recent growth in monitoring technology and in intrusion recognition systems, correctly detecting anomalies in a timely fashion remains a challenging task. One of the reasons for this is that many of today's security solutions yield tackle that gather and analyzes traffic from one link at a time. Similarly many research efforts consider anomaly

recognition on a per link basis [3]. To identify traffic anomalies one typically seeks to characterize, or build a model, of what constitutes normal behavior. After filtering out normal looking traffic, anomaly detection methods analyze the residual traffic pattern for deviations.

We discuss four different methods for signaling alerts when analyzing residual traffic. The simplest method compares the instantaneous residual traffic to a threshold. The second method considered is a small variation on the deviation score idea presented in [4]. Their key idea is to compare a local (temporally) variance calculation with a global variance assessment. The deviation score used in [4] is computed using output signals of a wavelet transform applied to IP flow level data from a single link. We apply this idea of comparing the local to the global variance on our filtered residual signal. In our third scheme, we apply wavelet analysis only on the filtered traffic (in [4] wavelet analysis is applied directly on the original signal). We signal an alert when the detail signal (now a type of residual) at each of a few different timescales exceeds a threshold. We raise an alarm only if the threshold is exceeded at multiple timescales. The fourth method uses a generalized likelihood ratio test to identify the moment an anomaly starts, by identifying a change in mean rate of the residual signal. These last two methods, introduced here for the first time, are particular applications of known statistical techniques to the anomaly detection domain.

While monitoring the traffic and detecting anomalous activities is important, it is equally important to keep the rate of false alarms low. A high false alarm means that the genuine events will be lost in the "snow" of false events. Suppose that we apply one's statistical anomaly detection method on large networks (with thousands of switches and routers involved and millions of users), even a very small false alarm rate may result in enough false alarms to overwhelm that network operation staff. In the worst case, false alarms undermine anomaly detection, as operation staff tire of reacting to false alarms, and ignore or turn the system off entirely. Currently, researchers are still struggling for a simple but robust method for anomaly detection, with high detection rate and low false alarm.[5]

Although anomaly detection has been addressed in many prior projects, there is the fact that few works have been succeeded in statistically characterized different types of network traffic flow anomalies. Furthermore, most anomaly detection methods are limited to analyzing the entire traffic as one entity, which makes them unable to quantify network anomalies, and their validities are affected when many anomalous activities occur simultaneously. From that we see the need for a method that can effectively detect and classify network anomalies based on flow statistics.

II. RELATED WORK:

Anomaly recognition has been addressed in many prior projects, and previous works have primarily focused on security tasks (detecting DDoS attacks, worms, or other intrusion…). In many cases, providers use very simple method for anomaly recognition, such as fixed threshold, packet capturing and analyzing… for the case of DDoS detection, Cisco and Juniper [6] also embedded in their routers a simple flood attack protection based on threshold technique. For the 3rd party solutions in network traffic anomaly detection such as D-Ward, Multops… they also tried to define thresholds for TCP, UDP and ICMP applications, then an attack or anomalous activities will be detected and given alarm whenever a threshold is go over. These methods are quite limited since there is no fixed threshold for different kind of networks. Also these methods call for qualified network operators to define thresholds and constantly monitor and modify them.

Several works have proposed various anomaly recognition methods (for a survey of results refers to [7]). Among these, a number of network traffic anomaly recognition methods use wavelets [8], Principal Component Analysis (PCA) [9], [10] or Kalman filters [11] to distinguish between normal and anomalous traffic, but do not provide information about the type of detected anomalies. The problem of anomaly extraction has been recently treated

In the literature [12], [13]. Among the existing proposals, the most relevant to our work [12] uses a frequent item-set mining (FIM) algorithm to identify the flows related to an anomaly from a given hint (e.g., an involved IP address) provided by an external anomaly detector.

To the best of our knowledge, only few works have addressed the problem of anomaly classification. Lakhina etal. [14] Cluster the output of a PCA-based anomaly detector to identify anomalies with similar behavior. Human intervention is necessary to find out the correspondence between each reported cluster and the high-level anomaly that it is describing. Tellenbach et al. [15] classify changes to generalized entropy metrics of traffic feature distributions to identify the type of detected anomalies. They demonstrate that this approach can classify with accuracy of around 85% synthetic anomalies. Most related to our approach, Choi et al. [13] make use of parallel coordinate plots to find unique patterns of attacks that are easy to recognize visually by a human expert. In contrast, a discussed technique that can automatically classify network anomalies without requiring later manual inspection.

A. Resource of System Statistics:

Acquire the correct type of network routine data is essential for anomaly recognition. The categories

of anomalies that can be sense are needy on the environment of the network data. In this we converse some achievable sources of network data along with their importance for detecting network anomalies. For the purpose of anomaly recognition, we must distinguish normal traffic behavior. The more precisely the traffic behavior can be sculpt, the better the anomaly recognition scheme will perform.

B. Network Explore:

Network explore are focused tackle such as ping and hint route that can be used to get specific network constraint such as end-to-end delay and packet loss. Snooping tools provide an instantaneous measure of network behavior. These techniques do not need the cooperation of the network service provider. However, it is possible that the service providers could choose not to allow ping traffic through their firewall. Furthermore, the specialized IP packets used by these tools need not follow the same trajectory or receive the same treatment by network devices as do the regular IP packets. This method also assumes the existence of symmetric paths between given source-destination pairs. On the Internet, this assumption cannot be guaranteed. Thus, performance metrics derived from such tools can provide only a coarse grained view of the network. Therefore, the data obtained from probing mechanisms may be of limited value for the purpose of anomaly detection.

C. Packet Filtering for Flow-Based Statistics:

In packet filtering, packet flows are sampled by capturing the IP headers of a select set of packets at different points in the network Information gathered from these IP headers is then used to provide detailed network performance information. For flow-based monitoring, a flow is identified by source destination addresses and source-destination port numbers. The packet filtering approach requires sophisticated network sampling techniques as well as specialized hardware at the network devices to do IP packet lookup. Data obtained from this method could be used to detect anomalous network flows. However, the hardware requirements required for this measurement method makes it difficult to use in practice.

D. Data from Routing Protocols:

Information about network events can be gain through the use of routing peers. For example by using an open shortest path first (OSPF) peer, it is possible to meet all routing table updates that are sent by the routers. The data collected can be used to build the network topology and provides link status updates. If the routers run OSPF with traffic engineering (TE) extensions, it is possible to get link utilization levels. Since routing updates occur at frequent intervals, any change in link utilization will be updated in near real time. However, since Routing updates must be kept small; only limited information pertaining to link statistics can be propagated through routing updates [17]

III. Anomaly Recognition Methods:

Statistical approach designed for Network Anomaly recognition:

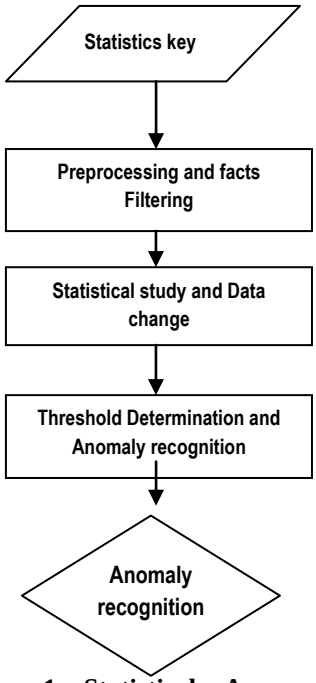

Fig. 1 Statistical Approach for Network Anomaly recognition

Fig. 1 Demonstrates the general steps implicated in statistical anomaly recognition. The first step is to preprocess or filter the given data inputs. This is an important step as the types of data available and the time scales in which these data are measured can significantly affect the recognition performance [5]. In the second step, statistical analysis and/or data transforms are performed to separate normal network behaviors from anomalous behaviors and noise. A variety of method can be applied here, e.g., Wavelet Analysis, Covariance Matrix analysis, and Principal Component Analysis. The main challenge here is to discover computationally efficient techniques for anomaly recognition with low false alarm rate. In the final step, decision theories such as Generalized Likelihood Ratio (GLR) test can be used to conclude whether there is a network anomaly depends on the variation observed. Statistical anomaly recognition can also be viewed from the machine learning perspective, where the goal is to find appropriate discriminate functions that can be used to classify any new input data vector into the normal or anomalous region with good accuracy for anomaly recognition. One subtle difference between statistical anomaly

recognition and machine learning based methods is that statistical approaches generally focus on statistical analysis of the composed data, whereas machine learning methods focuses on the "learning" part.

A. Change-Point Detection:

Statistical sequential change-point recognition has been useful successfully to network anomaly recognition. In [5], Thottan et al. characterize network anomalies with Management Information Base (MIB) variables undergoing abrupt changes in a correlated fashion. Given a set of MIB variables sampled at a fixed time-interval, the compute a network health function by combining the abnormality pointer of each individual MIB variable. This network health function can be used to conclude whether there is an anomaly in the network. In, Wang et al. detect SYN flooding attacks based on the dynamics of the differences between the number of SYN and FIN packets, which is modeled as a stationary erotic random process. The non-parametric Cumulative Sum (CUSUM) method is then used to detect the abrupt changes in the observed time series and thus detect the SYN flooding attacks.

B. Kalman Filter:

In [5], Soule et al. develop a traffic anomaly recognition scheme support on Kalman Filter. Unlike the work in Soule et al. process the link data using a Kalman filter rather than PCA analysis to forecast the traffic matrix one step into the future. After the forecast is made, the real traffic matrix is expected based on new link data. Then the difference between the forecast and the actual traffic matrix is used to identify traffic volume anomaly based on different threshold methods. Kalman filter has been applied successfully to a wide variety of problems involving the estimation of dynamics of linear systems from incomplete data. Thus, it is a talented tool for network anomaly recognition together with other more difficult models of non-linear dynamics.

C. Holt-Winters Predict Technique:

Holt-Winters Forecasting is a complicated algorithm that builds upon exponential level. Holt-Winters Forecasting rests on the basis that the pragmatic time series can be rotting into three components: a baseline, a linear trend, and a seasonal effect. The algorithm supposes each of these components evolves over time and this is skilled by applying exponential smoothing to incrementally update the components. The prediction is the sum of the three components: [2]

$XT+1 = nt + DT + Mt+1-m.$ (1)

The update formulas for the three components, or coefficients a, b, c is:

Baseline ("intercept"):

$at = \alpha (yt + ct-m) + (1 – \alpha)(at-1 + bt-1) .$ (2)

Linear Trend ("slope"):

$bt = \beta (at – at-1) + (1 – \beta) bt-1.$ (3)

Trend:

$ct = \gamma (yt – at) + (1 – \gamma) ct-m.$ (4)

As in exponential smoothing, the updated coefficient is an average of the calculation and an estimate obtained solely from the observed value yt, with fractions resolute by a model parameter (α, β, γ). Recall m is the period of the seasonal cycle; so the seasonal coefficient at time t references the last calculate coefficient for the same time point in the seasonal cycle.

The new approximation of the baseline is the observed value attuned by the best available estimate of the seasonal coefficient (ct-m). As the updated baseline needs to account for change due to the linear trend, the forecast slope is added to the baseline coefficient. The new estimate of the slope is simply the difference between the old and the new baseline (as the time interval between comments is fixed, it is not relevant). The new estimate of the seasonal component is the difference between the observed value and the corresponding baseline.

α, β and γ are the adaptation parameters of the algorithm and $0 < \alpha, \beta, \gamma < 1$. Larger values mean the algorithm adapts faster and predictions reflect recent observations in the time series; smaller values means the algorithm adapts slower, placing more weight on the past history of the time series. These values should be optimized when the algorithm is implemented.

D. Rule-Based Approaches:

Inconvenient work in this area of error or anomaly gratitude was depending on expert systems. In expert systems, a complete database grasp the rules of behavior of the injured system are used to conclude if a fault arise, [17]. Rule-based systems are too slow for real-time purpose and are dependent on prior knowledge about the fault conditions on the network. The recognition of faults in this approach depends on symptom that is specific to a particular manifestation of a fault. Examples of these symptoms are excessive utilization of bandwidth, number of open TCP connections, total throughput exceeded, etc. These rule-based systems rely heavily on the expertise of the network manager and do not adapt well to the developing network environment. Thus, it is possible that entirely new faults may escape detection. In, the authors describe an expert system model using fuzzy cognitive maps (FCMs) to overcome this limitation. FCM can be used to obtain an intelligent modeling of the propagation and interaction of network faults. FCMs are constructed with the nodes of the FCM indicate managed objects such as network nodes and the arcs signify the fault propagation model.

E. Finite State Machines:

Anomaly or fault recognition using finite state machines model alarm progression that occur throughout and prior to fault events. A probabilistic finite state machine model is built for a known network fault using history data. State machines are calculated with the intention of not just identify an anomaly but also possibly recognize and diagnosing the problem. The sequence of alarms acquire from the diverse points in the network is modeled as the states of a finite state machine. The alarms are implicit to contain information such as the device name as well as the symptom and time of incidence. The transitions between the states are measured using prior events [17]. A given cluster of alarms may have a number of clarifications, and the objective is to find the best explanation among them. The best justification is obtained by recognize a near-optimal set of nodes with minimum cardinality such that all entities in the set explain all the alarms and at least one of the nodes in the set is the most likely one to be in fault. In this approach, there is an underlying assumption that the alarms obtained are true. No attempt is made to produce the character alarms themselves.

F. Pattern Matching:

A new approach projected and execute by Maxion and others [17] explain anomalies as variation from normal behavior. This approach effort to deal with the inconsistency in the network surroundings. In this approach, online learning is used to build a traffic profile for a given network. Traffic profiles are built using symptom-specific feature vectors such as link utilization, packet loss, and number of collisions. These profiles are then categorized by time of day, day of week, and special days, such as weekends and holidays. When newly acquired data fails to fit within some confidence interval of the developed profiles then an anomaly is declared.

Generalized Likelihood Ratio test:

The standard approach for identify a change in a random process is the CUSUM (Cumulative Summation) method and its variation [3]. The main perception behind the CUSUM technique is that when a modify happen the log-likelihood ratio of an observation yi, defined as si = log L1(y) L0(y), shifts from a harmful value to a positive one (as after the change hypothesis H1 becomes more likely). This means that the log-likelihood of observing a sequence of N observations {yN−1 0}, defined as

SN−1 =PN−1

i=0 si, that was declining with N, begins to increase after the change. The minimum value of Sj gives an estimate of the change point. Therefore a simple statistical test for change detection consists of testing whether:

Sk − min

0≥j≥k

Sj > T,

Where Sk is the log-likelihood ratio distinct previously and T is a threshold. After a change has been detected, the time of change can be projected as: ˆtc = arg min 0≥j≥k {Sj}. The previously explain CUSUM algorithm has been extensively used for anomaly recognition. However it suffers from a key drawback.

It is stated in the context of a simple hypothesis, where the alternative hypothesis H1 should be completely defined, i.e. the level of the change or in other terms the intensity of the anomaly should be known a priori. However in practical settings, this is accurately unknown as by definition anomalies are not predictable.

A solution for this issue is afforded by the General Likelihood Ratio Test. In this advance the level of change in the CUSUM algorithm is substitute by its maximum likelihood estimate. To describe the approach let's fix a scenario.

Suppose an anomaly happen and this results in a shift in the mean of the remaining process. After the shift, the estimation error will no longer be a zero mean random variable of variance (is assumed to be known), but instead is translated to a mean μ, that is unknown, and the same variance. The GLR algorithm uses a window of estimation error { j+N−1j} and applies for each i, j ≥ i ≥ j+N−1 the following test. It first estimates the mean of the estimation error over the window {i, j + N − 1} as

ˆμ =1

j + N − 1 − i

j+XN−1

l=i

III. EXPERIMENTAL SETUP:

In this section we present the experimental setup of our research work not complete result. As mentioned we use the NS2 to calculate the result. Basically we focus on to detecting and preventing flood and flash crowd anomaly in network. Here we consider the 10 nodes in network and sending the packet at regular interval of time and providing the proper threshold to calculate the anomaly in network. The generalized ratio test can be used to divide the anomalous network. And draw the result through graph.

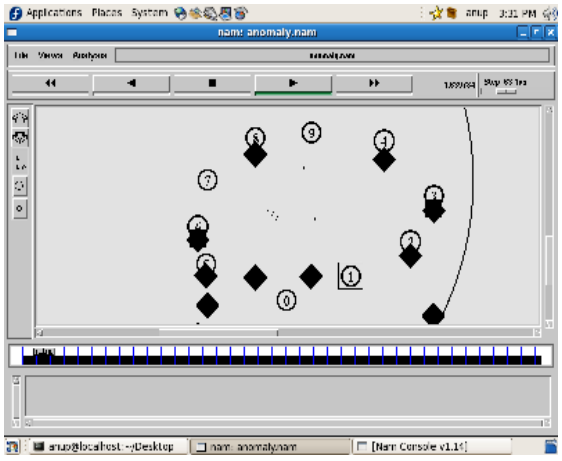

Fig.1 Design transmission of packet to node and packet dropping

IV. α-Stable Distribution as a Model for Network Traffic:

In this sect ion, we will review some statistical distributions which have been previously used to model network traffic, and see how the α-stable model can contribute to enhance traffic modeling. We will do this by looking at Poisson and Gaussian models in detail and stating some traffic properties we found in our data, which should be inherent to traffic coming from any data network. Then, we will see why neither Poisson nor Gaussian models can accommodate to these properties and try to answer the question of whether the α-stable model does.

V. Network traffic representation

Conventionally, network traffic has been model as a Poisson process for past reasons. Indeed, the Poisson model has been successfully utilize in telephone networks for many years, and so it was native when telecommunication networks became digital and started to send in order as data Packets [1]. Also, this model has a simple mathematical expression [1], and has only one parameter, λ, which is in turn very natural (the mean traffic in packets per time unit). In the last decade, however, several authors have considered network traffic behavior and proposed other models that conquer the limitations which are inherent to Poisson processes, the most notable one probably being that the Poisson model has a fixed relationship between mean and variance values (both are equal to λ).

More recently proposed models are usually found on the hypothesis that network traffic is self–similar in nature, a statement that was made in [26] for the first time. Naturally, network traffic can be contemplation of as a self–similar process because it is usually "busty" in nature, and this burstiness tends to emerge separately of the used time scale. Thus, in [26] FBM [26] is shown to fit accumulated network traffic data.

A proper model for instantaneous network traffic must be flexible enough to adapt to some properties seen in sampled traffic, namely: The amount of traffic accumulated at time t1 is less than, or equal to the amount of traffic accumulated at time t2, for every t1 < t2; that is, traffic increments are greater than, or equal to zero.

The fact that at time t there is a certain amount of traffic C does not imply in any way that at time t+1 the amount of traffic lies anywhere near C, due to the inherent nature of network traffic, which is often bursty and tends to show peaks from time to time. The latter property says that the variation in traffic from one time tick to the next one can be very large, On the other hand, the first aforementioned property makes symmetric distributions (Gaussian and Poisson distribution are symmetric) inappropriate, because if traffic data concentrates near the vertical axis, the model would allow negative traffic increments, and this can never be the case. Accordingly, if

Traffic data concentrates near the maximum transmission rate; a symmetric model would allow traffic increments to be larger than physically possible.

A. The α–stable Representation*:*

α–stable distributions can be consideration of as a superset of Gaussians and originate as the solution to the Central Limit Theorem when 2nd–order moments do not exist [24], that is, when data can abruptly change by huge amounts as time passes by. This fits nicely to the second of the talk about properties seen in network traffic. Moreover, α–stable distributions have an asymmetry parameter which allows their PDF to vary between totally left–asymmetric to totally right–asymmetric. While Poisson and Gaussian distributions are always symmetric. This parameter makes α–stable distributions fit logically to the first traffic property, even when average traffic is practically 0 or very near the maximum theoretical network throughput. In addition, α–stable distributions give an explanation to the restriction imposed in [26] about the need to aggregate so many traffic traces for them to converge to a Gaussian distribution. According to the Generalized Central Limit Theorem [27], which contains the infinite variance case, the sum of n α–stable distributions is another α–stable distribution, although not necessarily a Gaussian one. Since traffic data often has a huge variance (though obviously not infinite), and Under the hypothesis that it is α–stable, then the sum of a few traces will be α–stable but not Gaussian. However, after summing so many traces enough to overcome the enormous variance, the final histogram will converge to a Gaussian curve, as the traditional Central Limit Theorem states.

VI Conclusion:

This paper has presented idea about the statistical anomaly detection of network traffic. Here paper studied a statistical approach to analysis the distribution of network traffic to recognize the normal network traffic behavior This paper also discussed a method to recognize anomalies in network traffic, based on a non-restricted α-stable model and statistical hypothesis testing.

References:

[1]. Federico Simmross, Juan Ignacio, Pablo Casaseca-de-la-Higuera, Ioannis A. Dimitriadis" Anomaly Detection in Network Traffic Based on Statistical Inference and α-Stable Modeling" IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 8, NO. 4, JULY/AUGUST 2011

[2] K.-K. R. Choo, "The cyber threat landscape: Challenges and future research directions," Computers & Security, vol. 30, no. 8, pp. 719– 731, 2011
.
[3] Huy Anh Nguyen, Tam Van Nguyen, Dong Il Kim, Deokjai Choi " Network Traffic Anomalies Detection and Identification with Flow Monitoring" 2008 IEEE

[4] Augustin Soule, Kav´e Salamatian, Nina Taft "Combining Filtering and Statistical Methods for Anomaly Detection" USENIX Association Internet Measurement Conference 2005

[5] BARFORD, P., KLINE, J., PLONKA, D., AND RON, A. A signal analysis of network traffic anomalies. ACM Sigcomm IMW (2002).

[6] Marina Thottan, Guanglei Liu, Chuanyi Ji "Anomaly Detection Approaches for Communication Networks"

[7] Huy Anh Nguyen, Tam Van Nguyen, Dong Il Kim, Deokjai Choi " Network Traffic Anomalies Detection and Identification with Flow Monitoring"

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.

[9] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in Proceedings of ACM SIGCOMM Workshop on Internet Measurment, 2002

[10] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in Proceedings of ACM SIGCOMM, 2004.

[11] "Characterization of network-wide anomalies in traffic flows," in Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC), 2004

[12] A. Soule, K. Salamatian, and N. Taft, "Combining filtering and statistical methods for anomaly detection," in Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC), 2005

[13] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian, "Anomaly extraction in backbone networks using association rules," in Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC), 2009.

[14] F. Silveira and C. Diot, "URCA: Pulling out anomalies by their root causes," in Proceedings of IEEE INFOCOM, 2010

[15] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies use traffic feature distributions," in Proceedings of ACM SIGCOMM, 2005
.
[16] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies are using traffic feature distributions," in Proceedings of ACM SIGCOMM, 2005.

[17] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," Computers & Security, vol. 28, no. 5, pp. 276–288, 2009

[18] Marina Thottan and Chuanyi Ji "Anomaly Detection in IP Networks" IEEE TRANSACTIONS ON SIGNAL PROCESSING VOL. 51, NO. 8, AUGUST 2003

[19] S. S. Kim and A. L. N. Reddy. Statistical techniques for detecting traffic anomalies through packet header data. *IEEE/ACM Trans. Netw.*, 16:562{ 575, June 2008. (Cited on page 13.)

[20] A. Soule, H. Ringberg, F. Silveira, and C. Diot. Challenging the supremacy of traffic matrices in anomaly detection. *IMC '07*, pages 105{110, 2007. (Cited on pages 13 and 40.)

[21] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *SIGCOMM '05*, pages 217{228, 2005. (Cited on pages 12, 13, 25, 32, 40, 48, 57, 91 and 96.)

[22] [48] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identi_cation of network anomalies using sketch subspaces. *IMC '06*, pages 147{152, 2006. (Cited on pages 3, 5, 12, 13, 22 and 57.)

[23] S. S. Kim and A. L. N. Reddy. A study of analyzing network traffic as images in real-time. *INFOCOM '05*, pages 2056{2067, 2005. (Cited on pages 14 and 32.)

[24] L. I. Kuncheva. *Combining Pattern Classi_ers: Methods and Algorithms*. Wiley-Interscience, 2004. (Cited on pages 15 and 61.)

[25] S. Shanbhag and T. Wolf. Accurate anomaly detection through parallelism. *Netwrk. Mag. of Global Internetwkg.*, 23(1):22{28, 2009. (Cited on page 16.)

[26] [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

[27] "Metrology for Security and Quality of Service," http://www.laas.fr/METROSEC/, 2011

[28] " Anup Bhange, Amber Syed , Satyendra Singh Thakur" "ANOMALY DETECTION BASED ON DIVERSE APPROACHES IN NETWORK TRAFFIC" IJREAS Volume 2, Issue 2 (February 2012) ISSN: 2249-3905

[29] " Anup Bhange, Amber Syed , Satyendra Singh Thakur" "ANOMALY DETECTION BASED ON DIVERSE APPROACHES IN NETWORK TRAFFIC" **IJREAS Volume 2, Issue 2 (February 2012) ISSN: 2249-3905"**