Anomaly Detection Approaches for Communication Networks

Marina Thottan, Guanglei Liu, Chuanyi Ji

Abstract In recent years network anomaly detection has become an important area for both commercial interests as well as academic research. Applications of anomaly detection typically stem from the perspectives of network monitoring and network security. In network monitoring, a service provider is often interested in capturing such network characteristics as heavy flows that use a link with a given capacity, flow size distributions, and the number of distinct flows. In network security, the interest lies in characterizing known or unknown anomalous patterns of an attack or a virus.

In this chapter we review three main approaches to network anomaly detection: statistical methods, streaming algorithms, and machine learning approaches with a focus on unsupervised learning. We discuss the main features of the different approaches and discuss their pros and cons. We conclude the chapter by presenting some open problems in the area of network anomaly detection.

1 Introduction

In recent years network anomaly detection has become an important area for both commercial interests as well as academic research. Applications of anomaly detection typically stem from the perspectives of network monitoring and network secu-

Marina Thottan
Bell Labs, Alcatel-Lucent, 600-700 Mountain Avenue, Murray Hill, NJ 07974, USA, e-mail: marinat@alcatel-lucent.com

Guanglei Liu
Division of Mathematics and Sciences, Roane State Community College, 276 Patton Lane, Harriman, TN 37748 USA, e-mail: liug@roanestate.edu

Chuanyi Ji
School of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive, Atlanta, GA 30332 USA, e-mail: jic@ece.gatech.edu

rity. In network monitoring, a service provider is often interested in capturing such network characteristics as heavy flows that use a link with a given capacity, flow size distributions, and the number of distinct flows. In network security, the interest lies in characterizing known or unknown anomalous patterns of an attack or a virus.

A general definition of a network anomaly describes an event that deviates from the normal network behavior. However since there are no known models available for normal network behavior, it is difficult to develop an anomaly detector in the strictest sense. Based on the inherent complexity in characterizing the normal network behavior, the problem of anomaly detection can be categorized as model based and non-model based. In model based anomaly detectors, it is assumed that a known model is available for the normal behavior of certain specific aspects of the network and any deviation from the norm is deemed an anomaly. For network behaviors that cannot be characterized by a model, non-model based approaches are used. Non-model based approaches can be further classified based on the specific implementation and accuracy constraints that have been imposed on the detector.

In network monitoring scenarios where a statistical characterization of network anomalies is required, a known model is not a must. However, a statistical anomaly detector must have access to large volumes of data that can provide the required samples, from which accurate estimates of a network normal behavior can be made. However, with increasing speeds of network links, the frequency of sampling that is necessary for achieving a desired accuracy can be infeasible to implement. For example on an OC-768 link, packets arrive every 25 ns. Online monitoring of these packets requires per packet processing along with a large of amount of state information that must be kept in memory. This is a heavy burden on the limited SRAMs that are available on the router line cards. Thus sampling rates are resource constrained. Therefore under these circumstances it is more appropriate to use anomaly detection that can process long streams of data with small memory requirements and limited state information. Thus an online detection of anomalies with processing resource constraints corresponds to making some specific queries on the data and this is better handled by discrete algorithms that can process streaming data. In comparison with statistical sampling, streaming peruses every piece of data for the most important information while sampling processes only a small percentage of the data and absorbs all the information therein [52].

Machine learning approaches can be viewed as obtaining adaptively a mapping between measurements and network states, normal or anomalous. The goal is to develop learning algorithms that can extract pertinent information from measurements and can adapt to unknown network conditions and/or unseen anomalies. In a broad sense, statistical machine learning is one class of statistical approaches. A choice of a learning scenario depends on the information available in measurements. For example, a frequent scenario is that there are only raw network measurements available, and thus unsupervised learning methods are used. If additional information is available, e.g. from network operators, known anomalies or normal network behaviors, learning can be done with supervision. A choice of a mapping depends on the availability of a model, the amount and the type of measurements, and complexity of learning algorithms.

In this chapter we review all three approaches to network anomaly detection: statistical methods, streaming algorithms, and machine learning approaches with a focus on unsupervised learning. We discuss the main features of the different approaches and discuss their pros and cons. We conclude the chapter by presenting some open problems in the area of network anomaly detection.

2 Statistical Approaches for Network Anomaly Detection

In this section, we review statistical approaches for anomaly detection. Fig. 1 illustrates the general steps involved in statistical anomaly detection. 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 detection performance [48]. 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 techniques can be applied here, e.g., Wavelet Analysis, Covariance Matrix analysis, and Principal Component Analysis. The main challenge here is to find computationally efficient techniques for anomaly detection with low false alarm rate. In the final step, decision theories such as Generalized Likelihood Ratio (GLR) test can be used to determine whether there is a network anomaly based on the deviations observed.

In a broader context, statistical anomaly detection can also be viewed from the machine learning perspective, where the goal is to find appropriate discriminant functions that can be used to classify any new input data vector into the normal or anomalous region with good accuracy for anomaly detection. One subtle difference between statistical anomaly detection and machine learning based methods is that statistical approaches generally focus on statistical analysis of the collected data, whereas machine learning methods focuses on the "learning" part. In this section

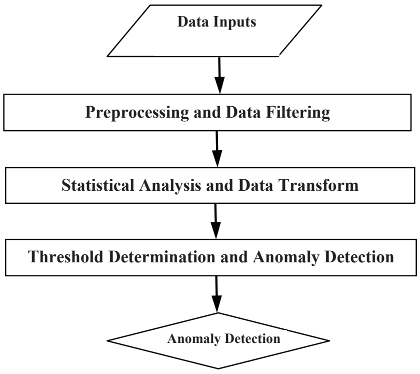
Data Inputs
Preprocessing and Data Filtering
Statistical Analysis and Data Transform
Threshold Determination and Anomaly Detection
Anomaly Detection

Fig. 1 Statistical Approach for Network Anomaly Detection

we solely focus on statistical anomaly detection due to its importance and popularity and review the general machine learning based methods in section 4.

2.1 Change-Point Detection

Statistical sequential change-point detection has been applied successfully to network anomaly detection. In [49, 48], 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 authors compute a network health function by combining the abnormality indicators of each individual MIB variable. This network health function can be used to determine whether there is an anomaly in the network.

In [51], 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 ergodic 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.

The work in both [49] and [51] models network dynamics as a quasi-stationary or stationary process. However, it is well known that under certain network scenarios the traffic dynamics exhibit non-stationary behaviors [34]. Thus better traffic modeling methods that can capture the non-stationary behavior could lead to improved anomaly detection with lower false alarm rates. In addition, not all detected abrupt changes correspond to network anomalies. Therefore, accurate characterization of anomalies in terms of abrupt changes in network dynamics is essential for effective anomaly detection [49].

2.2 Wavelet Analysis

Wavelet analysis has been applied to modeling of non-stationary data series because it can characterize the scaling properties in the temporal and frequency domains. In [39], Miller et al. apply wavelet transform techniques to anomaly detection in geophysical prospecting. Although in a different setting, the anomaly detection problem considered in [39] has some similarities to the problem of network anomaly detection. In both cases, one seeks to detect anomalous situations using limited and/or partial measurements. For example, in Geophysical prospecting, the data available for anomaly detection are usually scattered radiation collected at medium boundaries [39], while in network anomaly detection, limited data is a common scenario. In addition, there is the stringent requirement for computationally efficient anomaly detection with low alarm rate in both cases.

In [5], Barford et al. successfully apply wavelet techniques to network traffic anomaly detection. The authors develop a wavelet system that can effectively isolate both short and long-lived traffic anomalies. The wavelet analysis in [5] mainly focuses on aggregated traffic data in network flows. In [26], Kim et al. extend the work in [5] by studying IP packet header data at an egress router through wavelet analysis for traffic anomaly detection. Their approach is motivated by the observation that the out-bound traffic from an administrative domain is likely to exhibit strong correlation with itself over time. Thus, in [26], the authors study the correlation among addresses and port numbers over multiple timescales with discrete wavelet transforms. Traffic anomalies are detected if historical thresholds are exceeded in the analyzed signal. Wavelet analysis has proved to be an effective anomaly detection method. However, in [45], Soule et al. find that wavelet based methods for residual traffic analysis does not perform well compared to the simple GLR method. This raises the interesting question as to when wavelet analysis is useful for network anomaly detection [45].

2.3 Covariance Matrix Analysis

In [54], Yeung et al. develop a covariance matrix method to model and detect flooding attacks. Each element in the covariance matrix corresponds to the correlation between two monitored features at different sample sequences. The norm profile of the normal traffic can then be described by the mathematical expectation of all covariance matrices constructed from samples of the normal class in the training dataset. Anomaly can be detected with threshold-based detection schemes. The work in [54] uses second-order statistics of the monitored features for anomaly detection and is independent of assumptions on prior data distribution.

In [47], the covariance matrix method is extended, where the sign of the covariance matrices is used directly for anomaly detection. Compared to the classification and clustering methods in [54], detecting anomalies by comparing the sign of the covariance matrixes saves computation costs while maintaining low false-alarm rates. In a separate work [36], Mandjes et al. consider anomaly detection in voice over IP network based on the analysis of the variance of byte counts. The authors derive a general formula for the variance of the cumulative traffic over a fixed time interval, which can be used to determine the presence of a load anomaly in the network.

By employing second order features, covariance matrix analysis has been show to be a powerful anomaly detection method. One interesting direction in this area is to find what variables best characterize network anomaly and improve detection performance.

2.4 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality-reduction approach of mapping a set of data points onto new coordinates. Given a data matrix $\mathbf{Y}_{n\times m}$, $n \geq m$, PCA returns a set of k ($k \leq m$) ortho-normal vectors that defines a k-subspace. The k-subspace in theory characterizes the maximum variance in the original data matrix [30]. The spirit of PCA based anomaly detection is to separate the normal behavior from anomalies through dimensionality-reduction, which as discussed earlier can also be viewed from a Machine Learning perspective.

Lakhina et al. pioneered the application of Principal Component Analysis (PCA) to network-wide anomaly detection [30, 31, 32]. The basic idea of using PCA for traffic anomaly detection is that: the k-subspace obtained through PCA corresponds to the normal behavior of the traffic, whereas the remaining $(n-k)$ subspace corresponds to either the anomalies or the anomalies and the noise. Each new traffic measurement vector is projected on to the normal subspace and the anomalous subspace. Afterwards, different thresholds can be set to classify the traffic measurement as normal or anomalous. The source of the anomalous traffic can then be pinpointed by determining the ingress and egress points of different traffic flows. In the series of work [30, 31, 32], Lakhina et al. show that PCA can serve as a powerful method of network-wide anomaly detection with low false alarm rates.

The work in [30] assumes that all the data is processed off-line for anomaly detection, which cannot scale for large networks and may not detect anomalies in time. This motivates the work in [23], where Huang et al. consider on-line network anomaly detection on the basis of PCA. In [23], Huang et al. propose detection architectures based on distributed tracking and approximate PCA analysis. The idea is that only limited/partial data is transmitted though the network to a coordinator and anomaly detection is done at the coordinator with limited/partial view of the global states. Furthermore, using stochastic matrix perturbation theory, the authors develop a theoretical formulation on how to trade off quantization due to limiting the frequency and size of data communications against the overall detection accuracy. Experiments in [23] show that by choosing a 4% missed detection rate and a 6% false alarm rate, the proposed detection scheme can filter out more than 90% of the traffic from the original signal.

In [1], Ahmed et al. consider another online anomaly detection method based on kernel recursive least squares. The basic idea is that, given the low dimensionality of network traffic, regions occupied by the traffic features can be represented with a relatively small dictionary of linearly independent feature vectors. In addition, the size of the dictionary is much smaller than the size of the traffic measurements, thus facilitating the implementation of on-line anomaly detection. It has been demonstrated in [1] that the proposed algorithm can achieve similar detection performance to that in [31], but has the advantages of faster detection time and lower computational complexity.

In [24], Huang et al. propose to use network-wide analysis to improve detection of network disruptions. By studying the BGP update messages across a backbone network, the authors find that nearly 80% of the network disruptions exhibit some level of correlation across multiple routers in the network. Then Huang et al. apply PCA analysis techniques to the BGP updates and successfully detect all node and link failures and two thirds of the failures on the network periphery. The work in [24] also demonstrates that it is possible to combine the analysis of routing dynamics with static configuration analysis for network fault localization. Thus network-wide analysis techniques could be applied to online anomaly detection. However, as indicated in [24], one remaining open issue is to understand what information best enables network diagnosis and to understand the fundamental tradeoffs between the information available and the corresponding performance.

Although PCA-based approaches have been shown to be an effective method for network anomaly detection, in [41], Ringberg et al. point out the practical difficulty in tuning the parameters of the PCA based network anomaly detector. In [41], the authors perform a detailed study of the feature time series for detected anomalies in two IP backbone networks (Abilene and Geant). Their investigation shows that the false positive rate of the detector is sensitive to small differences in the number of principal components in the normal subspace and the effectiveness of PCA is sensitive to the level of aggregation of the traffic measurements. Furthermore, a large anomaly may contaminate the normal subspace, thus increasing the false alarm rate. Therefore, there remains one important open issue which is to find PCA-based anomaly detection techniques that are easy to tune and robust in practice.

2.5 Kalman Filter

In [45], Soule et al. develop a traffic anomaly detection scheme based on Kalman Filter. Unlike the work in [30, 31, 32], Soule et al. process the link data using a Kalman filter rather than PCA analysis to predict the traffic matrix one step into the future. After the prediction is made, the actual traffic matrix is estimated based on new link data. Then the difference between the prediction and the actual traffic matrix is used to detect 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 promising tool for network anomaly detection together with other more complicated models of non-linear dynamics.

3 Discrete Algorithms for Network Anomaly Detection

In many cases, network anomaly detection involves tracking significant changes in traffic patterns such as traffic volume or the number of traffic connections. Due to the high link speed and the large size of the Internet, it is usually not scalable to track the per-flow status of traffic. In [13], Duffield et al. proposed packet sampling to monitor flow-level traffic. By limiting the number of flows that need to be monitored, sampling can partially solve the scalability problem at the cost of anomaly detection performance. In this area, one important issue is to investigate the tradeoff between the amount of sampled information and the corresponding performance. In [4], Androulidakis et al. investigate selective flow-based sampling to find a good balance between the volume of sampled information and the anomaly detection accuracy. The authors suggest that selective sampling that focuses on small flows can achieve improved anomaly detection efficiency with fewer flows selected.

However, simple sampling cannot fully solve the scalability problem as any packets or flows that are not sampled may contain important information about anomalies. Furthermore, it is likely that this information can only be recovered if these packets or flows are sampled and stored. Thus, sometimes a large number of flows (up to 2^{64} flows based on different combinations of source and destination IP addresses) may need to be sampled to achieve an anomaly detector with good performance [28].

To address the disadvantages of sampling approaches, there has been extensive research in data streaming algorithms for anomaly detection in high-speed networks. A key difference between streaming and sampling is that streaming peruses every piece of data for the most important information while sampling digests a small percentage of data and absorbs all the information therein [52].

Specifically, using streaming techniques, the anomaly detection problem can be formulated as a heavy-hitter detection problem or a heavy-change detection problem. In the heavy-hitter detection problem, the goal is to identify the set of flows that represent a significantly large proportion of the ongoing traffic or the capacity of the link [16]. In the heavy-change detection problem, the goal is to detect the set of flows that have drastic change in traffic volume from one time period to another [28]. Data stream computation has been shown to be a powerful tool to solve these two problems.

3.1 Heavy-Hitter Detection

In the context of network anomaly detection, the goal of heavy-hitter detection is to efficiently identify the set of flows that represent a significantly large proportion of the link capacity or of the active traffic with small memory requirements and limited state information. Specifically, it can be formulated as follows [28]. Consider streams of data $\alpha_1, \alpha_2, \ldots$ that arrives sequentially, where each item $\alpha_i = (a_i, u_i)$ consists of a key $a_i \in \{0, 1, \ldots, n-1\}$ and an update $u_i \in \Re$. Associated with each key a_i is a time varying signal $A[a_i]$, i.e., the arrival of each new data item updates signal $A[a_i]$ such that $A[a_i] = A[a_i] + u_i$. In network anomaly detection, the key can be defined as the source and destination IP addresses, source and destination port numbers, protocol numbers and so on [28]. Thus n can be a very large number, e.g., $n = 2^{64}$ considering all possible source and destination IP address pairs. The updates can be viewed as sizes of packets in traffic flows. Then the problem of heavy-hitter detection is to find those items (keys) satisfying the following: $\frac{A[a_i]}{\sum_{l=0}^{n-1} A[a_l]} > \varepsilon$ or $A[a_i] > \phi$, where ε and ϕ are predetermined thresholds.

The challenge with heavy-hitter detection is that data processing cannot be done on a per-flow basis due to the large bandwidth of the current network. Thus, stream algorithms based on summary structures are used to solve this problem with guaranteed error bounds. In the area of data mining, there has been extensive research of algorithms for heavy-hitter detection. The connection of these algorithms to heavy-hitter detection in computer networks is first made in [16] and [37]. In [16], Estan et al. initiate a new direction in traffic measurement by recommending concentrating on only large flows, i.e., flows whose volumes are above certain thresholds. The authors also propose two algorithms for detecting large flows: sample and hold algorithm and multistage filters algorithm. Theoretical results show that the errors of these two algorithms are inversely proportional to the memory available. Furthermore, in [16], the authors note that network measurement problems bear a lot of similarities to measurement problems in other research areas such as data mining, and thus initiate the application of data streaming techniques to network anomaly detection.

In [37], Manku et al. propose the Sticky Sampling algorithm and the Lossy Counting algorithm to compute approximate frequency counts of elements in a data stream. The proposed algorithms can provide guaranteed error bounds and require small memory. Thus, these algorithms also provide powerful tools for solving the heavy-hitter detection problem in high-speed networks. In [6], Cormode et al. introduce the Count-Min sketch method to heavy-hitter detection. Sketch is a probabilistic summary data structure based on random projections. The authors note that it is an open problem to develop extremely simple and practical sketches for data streaming applications. In [8], Cormode et al. extend the study of heavy-hitter detection algorithms by identifying performance bottlenecks and tradeoffs in order to make these algorithms useful in practice. Using generated and actual IP packet data, the authors find that approximate complex aggregates are effective in providing accuracy guarantees with less processing load. In addition, adaptive implementations of "heavy-weight" approximations such as sketches are beneficial. The work in [8] provides excellent insights on how to make these heavy-hitter detection algorithms fast and space-efficient so that they can be used in high-speed data stream application.

Nevertheless, as indicated in [28], heavy-hitters are flows that represent a significantly large proportion of the ongoing traffic or the capacity of the link, but they do not necessarily correspond to flows experiencing significant changes. In terms of network anomaly detection, heavy-change detection is usually more informative than heavy-hitter detection. In addition, the solutions for heavy-hitter detection discussed here usually use data structures that do not have linearity property. Thus, they do not provide the more general ability to perform aggregate queries [43].

3.2 Heavy-Change Detection

In the context of network anomaly detection, the goal of heavy-change detection is to efficiently identify the set of flows that have drastic change in traffic volume from one time period to another with small memory requirements and limited state information. Specifically, it can be formulated similar to the heavy-hitter detection problem [28, 10]. Consider streams of data $\alpha_1, \alpha_2, \ldots$ that arrives sequentially, where each item $\alpha_i = (a_i, u_i)$ consists of a key $a_i \in \{0, 1, \ldots, n-1\}$ and an update $u_i \in \Re$. Associated with each key a_i is a time varying signal $A[a_i]$, i.e., the arrival of each new data item updates signal $A[a_i]$ such that $A[a_i] = A[a_i] + u_i$. We break time into discrete time intervals, $I_1, I_2, \ldots$, and define the value of $A[a_i]$ at time interval $k, k = 1, 2, \ldots, t$ as $s_{i,k}$. The problem of heavy-change detection is to find those items (keys) satisfying the following: $|s_{i,j} - s_{i,k}| > \varepsilon$ or $\frac{|s_{i,j}|}{max\{|s_{i,k}|,1\}} > \phi$, where ε and ϕ are predetermined thresholds. Note that here the changes can be defined using different measures of differences such as absolute difference, relative difference and so on [10].

Clearly heavy-change detection is a harder problem than heavy-hitter detection. In heavy-change detection, one data stream computation technique called sketch has shown great potential. The basic idea is to summarize the input streams so that per-flow analysis can be avoided. In [28], Krishnamurhty et al. first apply sketch to the heavy-change detection problem. With sketch-based change detection, input data streams are summarized using k-ary sketches. After sketches are created, different time series forecast models can be implemented on top of the summaries. Then the forecast errors are used to identify whether there is significant changes in the stream. The sketch-based techniques uses a small amount of memory and has constant pre-record update and reconstruction costs, thus it can be used for change detection in high-speed networks with a large number of flows. However, the k-ary sketch based change detection has one main drawback: the k-ary sketch is irreversible, thus making it impossible to reconstruct the desired set of anomalous keys without querying every IP address or querying every address in the stream if these IP addresses are saved.

To address these problems, in [43, 44], Schweller et al. develop change detection schemes based on reversible sketch data structures. The basic idea is to hash intelligently by modifying the input keys and hashing functions so that keys with heavy changes can be recovered [44]. Using reverse hashing schemes, the authors can efficiently identify the set of all anomalous keys in the sketch. In addition, the authors introduce the bucket index matrix algorithm for accurate multiple heavy-change detection. Empirical results show that the reverse hashing is capable of detecting heavy changes and identifying the anomalous flows in real time. In [17], Gao et al. extend the work in [28, 43, 44] by considering an optimal small set of metrics and building two-dimensional sketches for flow-level traffic anomaly detection. The authors also implement a high-speed on-line intrusion detection system based on Two-Dimensional Sketches, which is shown to be capable of detecting multiple types of attacks simultaneously with high accuracy.

In a separate work [9, 10], Cormode et al. introduce the deltoids concept for heavy-change detection, where a deltoid is defined as an item that has a large difference. The authors propose a framework based on a structure of Combinational Group Testing to find significant deltoids in high speed networks. It is also shown that the proposed algorithms are capable of finding significant deltoids with small memory and update time, and with guaranteed pre-specified accuracy. As commented in [43], deltoids can be considered as an expansion of k-ary sketch with multiple counters for each bucket in the hash table at the cost of memory requirements.

Sketch-based schemes are capable of detecting significant changes efficiently with high accuracy. Nevertheless, there still remain important challenges before the wide deployment of sketches-based network anomaly detection, e.g., how to identify minimal set of metrics for monitoring to be recorded by sketches given the wide variety of attacks/anomalies exhibited in the current Internet.

4 Machine Learning for Anomaly Detection

Machine learning approaches attempt to obtain an anomaly detection that adapts to measurements, changing network conditions, and unseen anomalies. What is a machine learning view of network-anomaly detection? Let $X(t) \in \Re$ (X in short) be an n-dimensional random feature vector at time $t \in [0,t]$. Consider the simplest scenario that there are two underlying states of a network, $\omega_i, i = 0,1$, where $\omega_0 = 0$ corresponds to normal network operation, and $\omega_1 = 1$ corresponds to "unusual or anomalous" network state. Detecting anomaly can be considered as deciding whether a given observation x of random feature vector X is a symptom of an underlying network state ω_0 or ω_1. That is, a mapping needs to be obtained between X and ω_i for $i = 0,1$. Note that an anomalous network state may and may not correspond to abnormal network operation. For example, flash crowd, which is a surge of user activities, may result from either an installment of new software under a normal network operation or an abnormal network state due to DDOS or Worm attacks.

Due to the complexity of networks, such a mapping is usually unknown but can be learned from measurements. Assume that a set D of m measurements is collected from a network as observations on X, i.e., $D = \{x_i(t)\}_{i=1}^{m}$, where $x_i(t)$ is the i-th observation for $t \in [0,t]$. $x_i(t)$ is called a training sample in machine learning. In addition, another set $D_l = \{y_q\}_{l=1}^{k}$ of k measurements is assumed to be available in general that are samples $(y_q = 0,1)$ on ω_i's. y_q's are called labels in machine learning. A pair $(x_i(t), y_i)$ is called a labeled measurement in machine learning, where observation $x_i(t)$ is obtained when a network is in a known state. For example, if measurement $x_i(t)$ is taken when the network is known to operate normally, $y_i = 0$ and $(x_i(t), 0)$ is considered as a sample for normal network operation. If $x_i(t)$ is taken when the network is known to operate abnormally, $y_i = 1$ and $(x_i(t), 1)$ is considered as a "signature" in anomaly detection. In general $x_i(t)$ is considered as an unlabeled measurement, meaning the observation $x_i(t)$ occurs when the network state is unknown. Hence, three types of network measurements are: (a) normal data $D_n = \{x_i(t), 0\}_{i=1}^{k-u}$, (b) unlabeled data $D = \{x_j(t)\}_{j=1}^{m}$, and (c) anomalous data $D_l = \{x_r(t), 1\}_{r=1}^{u}$. A training set consists of all three types of data in general although a frequent scenario is that only D and D_n are available. Examples of D include raw measurements on end-end flows, packet traces, and data from MIB variables. Examples of D_n and D_l can be such measurements obtained under normal or anomalous network conditions, respectively.

Given a set of training samples, a machine learning view of anomaly detection is to learn a mapping $f(.)$ using the training set, where $f(.) : Training\ Set \ \rightarrow\ \omega_i$, so that a desired performance can be achieved on assigning a new sample x to one of the two categories. Fig. 2 illustrates the learning problem.

A training set determines the amount of information available, and thus categorizes different types of learning algorithms for anomaly detection. Specifically, when only D is available, learning and thus anomaly detection is unsupervised. When D and D_n are available, learning/anomaly detection can, be viewed as unsupervised. When D_l and D_n are both available, learning/anomaly detection becomes supervised, since we have labels or signatures. $f(.)$ determines the architecture of a "learning machine", which can either be a model with an analytical expression or a computational algorithm. When $f(.)$ is a model, learning algorithms for anomaly detection are parametric, i.e., model-based; otherwise, learning algorithms are non-parametric, i.e., non-model-based.

4.1 Unsupervised Anomaly Detection

We now focus on unsupervised learning for anomaly detection, where D_l is not available, a mapping $f(.)$ is learned using raw measurements D and normal data D_n. Unsupervised learning is the most common scenario in anomaly detection due to its practicality: Networks provide a rich variety and a huge amount of raw measurements and normal data.

One unsupervised learning approach for anomaly detection is *behavioral-based*. That is, D_n together with D is learned to characterize a normal network behavior. A deviation from the norm is considered as an anomaly.

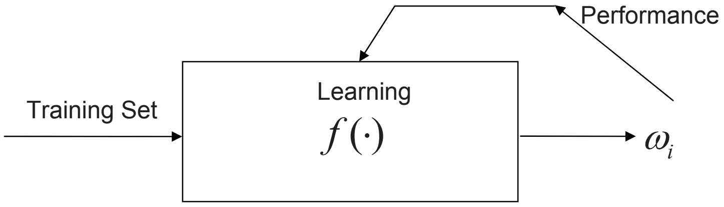

Fig. 2 Machine Learning View of Anomaly Detection

4.1.1 Adaptive Threshold-Based Anomaly Detection

Early work [38, 22, 48] in this area begins with a small network, e.g. interface ports at a router, and chooses aggregated measurements, e.g. MIB variables that are readily available from most of the network equipments. Measurements are assumed to be collected a portion at a time within a moving window across a chosen time duration [22, 48]. Both off-line and on-line learning have been considered, and simple algorithms are experimented. For example, [48, 22] select model-based learning, where X(t) from normal operations is modeled as a second-order AR process, i.e., $f(X) = a_0 + a_1X + a_2X^2 + \varepsilon$. $a_i, i = 0, 1, 2$ are parameters and learned using measurements collected from a moving window of a chosen time-duration, and ε is assumed to be Gaussian residual noise. A likelihood ratio test is applied. If the sample variance of the residual noise exceeds a chosen threshold, an observation x is classified as an anomaly. The model has been successful in detecting a wide-range of anomalies such as flash-crowds, congestion, broadcast storms [48], and later worm attacks [11]. In addition, these anomalies have been detected proactively before they cause catastrophic network failures [22, 48]. One disadvantage of such an approach is the difficulty of choosing an appropriate time scale where AR processes can accurately characterize normal behavior of the MIB variables. The AR model with Gaussian noise is also not applicable to more complex temporal characteristics, e.g. bursty and non-Gaussian variables.

More recently a broader class of raw measurements has been selected for anomaly detection. For example, IP packet traces are used to detect faults/anomalies related to load changes (see [5] and references therein). The number of TCP SYN and FIN (RST) per unit time is used to detect DDOS attacks [51]. Route updates is used to understand routing anomalies and/or BGP routing instability [56, 55, 24]. Aggregated rates are used to detect bottlenecks in 3G networks [40]. These measurements are of different types but are all aggregated data (counts) rather than individual packets. Aggregated measurements facilitate scalable detection but do not contain detailed information, e.g. in packet headers that are useful for end-end flow-based measurements.

More general algorithms have also been developed that go beyond simple models. One type of algorithm aims at better characterizing temporal characteristics in measurements. For example, wavelets are applied to BGP routing updates [55], and four other types of measurements (outages, flash crowds, attacks, and measurement errors) [5]. In this case, wavelets obtain time scales at which anomalies can be better exemplified. Another type of algorithm aims at better detection. For example, non-parametric hypothesis-testing is used to accommodate non-Gaussian statistics in DDOS attacks [51]. Generalized likelihood ratio test is applied to BGP measurements to detect anomalies upon worm attacks [11].

4.1.2 Clustering

Clustering is another class of algorithms in unsupervised anomaly detection. Clustering algorithms [12] characterize anomalies based on dissimilarities. The premise is that measurements that correspond to normal operations are similar and thus cluster together. Measurements that do not fall in the "normal clusters" are considered as anomalies.

A similarity measure is a key parameter in a clustering algorithm that distinguishes the normal from anomalous clusters. "Similarity" measures distances among the input x_i's. Euclidian distance is a commonly-used similarity measure for network measurements. When samples lie with distances below a chosen threshold, they are grouped into one cluster [12]. Such an approach has been applied in the following examples: (a) to cluster network flows from the Abilene network [2] to detect abrupt changes in the traffic distribution over an entire network, and (b) to cluster BGP update messages and detect unstable prefixes that experience frequent route changes [55]. More sophisticated algorithms are applied such as hierarchical clustering to group BGP measurements based on coarse-to-fine distances [3]. An advantage of hierarchical clustering is computational efficiency, especially when applied to large network-data sets [3]. Other more sophisticated clustering algorithms improve local convergence using fuzzy-k-mean and swam-k-mean algorithms, and are applied to detecting network attacks [14].

Mixture models provide a model-based approach for clustering [12]. A mixture model is a linear combination of basis functions. Gaussian basis functions with different parameters are commonly used to represent individual clusters. The basis functions are weighted and combined to form the mapping . Parameters of the basis functions and the weighting coefficients can be learned using measurements [12]. For example, [20] uses Gaussian mixture models to characterize utilization measurements. Parameters of the model are estimated using Expectation-Maximization (EM) algorithm and anomalies are detected corresponding to network failure events. One limitation of clustering algorithms is that they do not provide an explicit representation of the statistical dependence exhibited in raw measurements. Such a representation is important for correlating multiple variables in detection, and diagnosis of anomalies.

4.1.3 Bayesian Belief Networks

Bayesian Belief Networks (mapping $f(.)$) provides the capability to capture the statistical dependence or causal-relations among variables and anomalies.

An early work [22] applies Bayesian Belief Networks to MIB variables in proactive network fault detection. The premise is that many variables in a network may exhibit anomalous behavior upon the occurrence of an event, and can be combined to result in a network-wide view of anomalies that may be more robust and result in a more accurate detection. Specifically, the Bayesian Belief Networks [22] first combines MIB variables within a protocol layer, and then aggregates intermediate variables of protocol layers to form a network-wide view of anomalies. Combinations are done through conditional probabilities in the Bayesian Belief Networks. The conditional probabilities were determined a priori.

Recent work [27] uses Bayesian Belief Networks to combine and correlate different types of measurements such as traffic volumes, ingress/egress packets, and bit rates. Parameters in the Bayesian Belief Network are learned using measurements. The performance of the Bayesian Belief Network compares favorably with that of wavelet models and time-series detections, especially in lowering false alarm rates [27].

[25] provides an example of Bayesian Networks in unsupervised anomaly detection at the application layer. In this work a node in a graph represents a service and an edge represents a dependency between services. The edge weights vary with time, and are estimated using measurements. Anomaly detection is conducted from a time sequence of graphs with different link weights. The paper shows that service-anomalies are detected with little information on normal network-behaviors.

4.1.4 Other Approaches

Using information theoretical measures is another approach for unsupervised anomaly detection. Information theoretical measures are based on probability distributions. Hence information from distributions that are estimated from the data is used in anomaly detection.

[33] provides several information theoretical measures for anomaly detection of network attacks. The measures include entropy and conditional entropy as well as information gain/cost for anomaly detection. The entropy measures are applied to several data sets for network security including call data and “tcpdump” data. The entropy measures also assist in selecting parameters for detection algorithms.

[18] applies information theoretical measures in behavioral-based anomaly detection. First, the Maximum Entropy principle is applied to estimate the distribution for normal network operation using raw data. Second, the relative entropy of the network traffic is applied with respect to the distribution of the normal behavior. The approach detects anomalies that not only cause abrupt changes but also gradual changes in network traffic. (See [18] and references therein for other related work that uses entropy measures for anomaly detection.)

[29] uses entropy measures to flow data, and develops algorithms for estimating the entropy of streaming algorithms. Hidden Markov Model is another approach in unsupervised anomaly detection. For example, [53] uses a Hidden Markov Model to correlate observation sequences and state transitions so that the most probable intrusion sequences can be predicted. The approach is applied to intrusion detection data sets (KDDCUP99) and shown to reduce the false alarm rates effectively.

4.2 Learning with Additional Information

When additional information is available, other learning scenarios can be considered to learning the mapping $f(.)$. For example, [21] applies reinforcement learning in proactive network fault detection based on partially observable Markov Decision Processes. This scenario corresponds to the situation that contains reinforcement signals that guide the learning process (see [12] for details).

When labeled anomalous data D_l is available, supervised learning can be considered to learn the mapping for anomaly detection. Probe-measurements can be used to provide a source of such data during anomalous events since they gain direct information on network status upon failures/attacks. [42] uses probe measurements to develop a dynamic Bayesian Belief Network for adaptive network failure diagnosis. [46] uses simulated failure data to build a Bayesian Belief Network for fault-localization.

Recently, [19] shows that labeled measurements from known applications can be used in supervised learning to extract features for other applications. [15] shows that a large number of raw measurements and a small number of failure data result in efficient inference of large-scale network-service disruptions upon natural disaster.

5 Challenges and Deployment Issues

From the above study of the different anomaly detection approaches that are available today, it is clear that a black box anomaly detector may indeed be a utopian dream [50] for two main reasons: (1) the nature of the information that is fed to the anomaly detector could be varied both in format and range, and (2) the nature of the anomaly, its frequency of occurrence and resource constraints clearly dictates the detection method of choice. In [50] the authors propose an initial prototype anomaly detector that transforms the input data into some common format before choosing the appropriate detection methodology. This is clearly an area where further research is an important contribution, especially for deployment in service provider environments where it is necessary to build multiple anomaly detectors to address the myriad monitoring requirements.

Some of the challenges encountered when employing machine learning approaches or statistical approaches is the multiple time scales in which different network events of interest occur. Capturing the characteristics of multi time scale anomalies is difficult since the time scale of interest could be different for different anomaly types and also within an anomaly type depending on the network conditions. In [38] the authors describe the influence of the regularity of data on the performance of a probabilistic detector. It was observed that false alarms increase as a function of the regularity of the data. The authors also show that the regularity of the data is not merely a function of user type or environments but also differs within user sessions and among users. Designing anomaly detectors that can adapt to the changing nature of input data is an extremely challenging task. Most anomaly detectors employed today are affected by the inherent changes in the structure of the data that is being input to the detector and therefore does affect performance parameters such as probability of hits and misses, and false alarm rates.

Sampling strategies for multi time scale events with resource constraints is another area where there is a need for improved scientific understanding that will aid the design of anomaly detection modules. [35] discovered that most sampling methods employed today introduce significant bias into measured data, thus possibly deteriorating the effectiveness of the anomaly detection. Specifically, Mai et al use packet traces obtained from a Tier-1 IP-backbone using four sampling methods including random and smart sampling. The sampled data is then used to detect volume anomalies and port scans in different algorithms such as wavelet models and hypothesis testing. Significant bias is discovered in these commonly-used sampling techniques, suggesting possible bias in anomaly detection.

Often, the detection of network anomalies requires the correlation of events across multiple correlated inputs data sets. Using statistical approaches it is challenging to capture the statistical dependencies observed in the raw data. When using streaming algorithms it is also impossible to capture these statistical dependencies unless there are some rule based engines that can correlate or couple queries from multiple streaming algorithms. Despite the challenges, the representation of these dependencies across multiple input data streams is necessary for the detailed diagnosis of network anomalies. To sum up, there still remain several open issues to improve the efficiency and feasibility of anomaly detection. One of the most urgent issues is to understand what information can best facilitate network anomaly detection. A second issue is to investigate the fundamental tradeoffs between the amount/complexity of information available and the detection performance, so that computationally efficient real-time anomaly detection is feasible in practice. Another interesting problem is to systematically investigate each anomaly detection method and understand when and in what problem domains these methods perform well.

References

1. Ahmed T., Coates M., Lakhina A.: Multivariate Online Anomaly Detection Using Kernel Recursive Least Squares. Proc. of 26th IEEE International Conference on Computer Communications (2007)
2. Ahmed T., Oreshkin B., Coates M.: Machine Learning Approaches to Network Anomaly Detection. Proc. of International Measurement Conference (2007)
3. Andersen D., Feamster N., Bauer S., Balaskrishman H.: Topology inference from BGP routing dynamics. Proc. SIGCOM Internet Measurements Workshop, Marseille, France (2002)
4. Androulidakis G., Papavassiliou S.: Improving Network Anomaly Detection via Selective Flow-Based Sampling. Communications, IET. Vol.2, no.3, 399-409 (2008)
5. Barford P., Kline J., Plonka D., Ron A.: A Signal Analysis of Network Traffic Anomalies. Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurements, 71 - 82 (2002)
6. Cormode G., Korn F., Muthukrishnan S., D. Srivastava D.: Finding Hierarchical Heavy Hitters in Data Streams. Proc. of VLDB, Berlin, Germany (2003)

7. Cormode G., Muthukrishan S.: Improved Data Stream Summaries: The Count-Min Sketch and Its Applications. Tech. Rep. 03-20, DIMACS (2003)
8. Cormode G., Johnson T., Korn F., Muthukrishnan S., Spatscheck O., Srivastava D.: Holistic UDAFs at Streaming Speeds. Proc. of ACM SIGMOD, Paris, France (2004)
9. Cormode G., Muthukrishnan S.: What's New: Finding Significant Differences in Network Data Streams. Proc. of IEEE INFOCOM (2004)
10. Cormode G., S. Muthukrishnan S.: What's New: Finding Significant Differences in Network Data Streams. IEEE/ACM Trans. Netw. 13(6): 1219-1232 (2005)
11. Deshpande S., Thottan M., Sikdar B.: Early Detection of BGP Instabilities resulting From Internet Worm Attacks. Proc. of IEEE Globecom, Dallas, TX (2004)
12. Duda R. O., Hart P., Stork D.: Pattern Classification, 2nd edn. John Willy and Sons (2001)
13. Duffield N.G., Lund C., Thorup M.: Properties and Prediction of Flow Statistics from Sampled Packet Streams. Proc. of ACM SIGCOMM Internet Measurement Workshop (2002)
14. Ensafi R., Dehghanzadeh S., Mohammad R., Akbarzadeh T.: Optimizing Fuzzy K-Means for Network Anomaly Detection Using PSO. Computer Systems and Applications, IEEE/ACS International Conference, 686-693 (2008)
15. Erjongmanee S., Ji C.: Inferring Internet Service Disruptions upon A Natural Disaster. To appear at 2nd International Workshop on Knowledge Discovery from Sensor Data (2008)
16. Estan C., Varghese G.: New Directions in Traffic Measurement and Accounting. Proc. of ACM SIGCOMM, New York, USA (2002)
17. Gao Y., Li Z., Chen Y.: A DoS Resilient Flow-level Intrusion Detection Approach for High-speed Networks, Proc. of IEEE International Conference on Distributed Computing Systems (2006)
18. Gu Y., McCallum A., Towsley D.: Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation. Proc. of IMC (2005)
19. Haffner P., Sen S., Spatscheck O., Wang D.: ACAS: Automated Construction of Application Signatures. Proc. of ACM SIGCOMM Workshop on Mining Network Data, Philadelphia, (2005)
20. Hajji H.: Statistical Analysis of Network Traffic for Adaptive Faults Detection. IEEE Trans. Neural Networks. Vol. 16, no. 5, 1053-1063 (2005)
21. He Q., Shayman M.A.: Using Reinforcement Learning for Pro-Active Network Fault Management. Proc. of Communication Technology. Vol. 1, 515-521 (2000)
22. Hood C.S., Ji C.: Proactive Network Fault Detection. IEEE Tran. Reliability. Vol. 46 3, 333-341 (1997)
23. Huang L., Nguyen X., Garofalakis M., Jordan M.I., Joseph A., Taft N.: Communication-Efficient Online Detection of Network-Wide Anomalies. Proc. of 26th Annual IEEE Conference on Computer Communications (2007)
24. Huang Y., Feamster N., Lakhina A., Xu J.: Diagnosing Network Disruptions with Network-Wide Analysis. Proc. of ACM SIGMETRICS (2007)
25. Ide T., Kashima H.: Eigenspace-Based Anomaly Detection in Computer Systems. Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, 440 - 449 (2004)
26. Kim S. S., Reddy A.: Statistical Techniques for Detecting Traffic Anomalies Through Packet Header Data. Accepted by IEEE/ACM Tran. Networking (2008)
27. Kline K., Nam S., Barford P., Plonka D., Ron A.: Traffic Anomaly Detection at Fine Time Scales with Bayes Nets. To appear in the International Conference on Internet Monitoring and Protection (2008)
28. Krishnamurthy B., Sen S., Zhang Y., Chan Y.: Sketch-Based Change Detection: Methods, Evaluation, and Applications. Proc. of ACM SIGCOMM IMC, Florida, USA (2003)
29. Lall S., Sekar V., Ogihara M., Xu J., Zhang H.: Data Streaming Algorithms for Estimating Entropy of Network Traffic. Proc. of ACM SIGMETRICS (2006)
30. Lakhina A., Crovella M, Diot C.: Diagnosing Network-Wide Traffic Anomalies. Proc. of ACM SIGCOMM (2004)
31. Lakhina A., Papagiannaki K, Crovella M., Diot C., Kolaczyk E., N. Taft N.: Structural Analysis of Network Traffic Flows. Proc. of ACM SIGMETRICS (2004)

32. Lakhina A., Crovella M., Diot C.: Mining Anomalies Using Traffic Feature Distributions. Proc. of ACM SIGCOMM, Philadelphia, PA (2005)
33. Lee W., Xiang D.: Information-Theoretic Measures for Anomaly Detection. Proc. of IEEE Symposium on Security and Privacy (2001)
34. Leland W. E., Taqqu M. S., Willinger W., Wilson D. V.: On the Self-Similar Nature of Ethernet Traffic, Proc. of ACM SIGCOMM (1993)
35. Mai J., Chuah C., Sridharan A., Ye T., Zang H.: Is Sampled Data Sufficient for Anomaly Detection? Proc. of 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeriro, Brazil. 165 - 176 (2006)
36. Mandjes M., Saniee I., Stolyar A. L.: Load Characterization and Anomaly Detection for Voice over IP traffic. IEEE Tran. Neural Networks. Vol.16, no.5, 1019-1026 (2005)
37. Manku G. S., Motwani R.: Approximate Frequency Counts over Data Streams. Proc. of IEEE VLDB, Hong Kong, China (2002)
38. Maxion R. A., Tan K. M. C.: Benchmarking Anomaly-Based Detection Systems. Proc. International Conference on Dependable Systems and Networks (2000)
39. Miller E. L., Willsky A. S.: Multiscale, Statistical Anomaly Detection Analysis and Algorithms for Linearized Inverse Scattering Problems. Multidimensional Systems and Signal Processing. Vol. 8, 151-184 (1997)
40. Ricciato F., Fleischer W.: Bottleneck Detection via Aggregate Rate Analysis: A Real Case in a 3G Network. Proc. IEEE/IFIP NOMS (2004)
41. Ringberg H., Soule A., Rexford J., Diot C.: Sensitivity of PCA for Traffic Anomaly Detection. Proc. of ACM SIGMETRICS (2007)
42. Rish I., Brodie M., Sheng M., Odintsova N., Beygelzimer A., Grabarnik G., Hernandez K.: Adaptive Diagnosis in Distributed Systems. IEEE Tran. Neural Networks. Vol. 16, No. 5, 1088 - 1109 (2005)
43. Schweller R., Gupta A., Parsons E., Chen Y.: Reversible Sketches for Efficient and Accurate Change Detection over Network Data Streams. Proc. of IMC, Italy (2004)
44. Schweller R., Li Z., Chen Y., Gao Y., Gupta A., Zhang Y., Dinda P., Kao M., Memik G.: Reverse hashing for High-Speed Network Monitoring: Algorithms, Evaluation, and Applications. Proc. of IEEE INFOCOM (2006)
45. Soule A., Salamatian K., Taft N.: Combining Filtering and Statistical Methods for Anomaly Detection. Proc. of IMC Workshop (2005)
46. Steinder M., Sethi A.S.: Probabilistic Fault Localization in Communication Systems Using Belief Networks. IEEE/ACM Trans. Networking. Vol. 12, No. 5, 809- 822 (2004)
47. Tavallaee M., Lu W., Iqbal S. A., Ghorbani A.: A Novel Covariance Matrix Based Approach for Detecting Network Anomalies. Communication Networks and Services Research Conference (2008)
48. Thottan M., Ji C.: Anomaly Detection in IP Networks. IEEE Trans. Signal Processing, Special Issue of Signal Processing in Networking, Vol.51, No.8, 2191-2204 (2003)
49. Thottan M., Ji C.: Proactive Anomaly Detection Using Distributed Intelligent Agents. IEEE Network. Vol. 12, no. 5, 21-27 (1998)
50. Venkataraman S., Caballero J., Song D., Blum A., Yates J.: Black-box Anomaly Detection: Is it Utopian?" Proc. of the Fifth Workshop on Hot Topics in Networking (HotNets-V), Irvine, CA (2006)
51. Wang, H., Zhang, D., Shin, K. G: Detecting SYN flooding attacks. Proc. of IEEE INFOCOM (2002)
52. Xu J.: Tutorial on Network Data Streaming. SIGMETRICS (2007)
53. Yang Y., Deng F., Yang H.: An Unsupervised Anomaly Detection Approach using Subtractive Clustering and Hidden Markov Model. Communications and Networking in China. 313-316 (2007)
54. Yeung D. S., Jin S., Wang X.: Covariance-Matrix Modeling and Detecting Various Flooding Attacks. IEEE Tran. Systems, Man and Cybernetics, Part A, vol.37, no.2, 157-169 (2007)
55. Zhang J., Rexford J., Feigenbaum, J: Learning-Based Anomaly Detection in BGP Updates. Proc. of ACM SIGCOMM MineNet workshop (2005)
56. Zhang Y., Ge Z., Greenberg A., Roughan M.: Network Anomography. Proc. of ACM/USENIX Internet Measurement Conference (2005)