A Comparative Study of Two Network-based Anomaly Detection Methods

Kaustubh Nyalkalkar*, Sushant Sinha†, Michael Bailey* and Farnam Jahanian*
*Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA
†Yahoo!, Bangalore, India
{*knyalkal, sushant, mibailey, farnam*}*@umich.edu*

***Abstract*—Modern networks are complex and hence, network operators often rely on automation to assist in assuring the security, availability, and performance of these networks. At the core of many of these systems are general-purpose anomaly-detection algorithms that seek to identify normal behavior and detect deviations. While the number and variations of these algorithms are large, two broad categories have emerged as leading approaches to this problem: those based on spatial correlation and those based on temporal analysis. In this paper, we compare one promising approach from each of these categories, namely entropy-based PCA and HHH-based wavelets.**

I. Introduction

Network operators face an enormous task in maintaining large networks. A number of general-purpose, network-based anomaly detection systems have been proposed to assist them in this effort. These network-based anomaly detection systems continuously collect different metrics (e.g., flow-count, packet-count, byte-count on different links) from the network as their input and try to isolate *anomalies*—which can be broadly defined as events of particular interest to network operators. While the number and diversity of such algorithms is quite large, two general classes of anomaly detection algorithms, namely spatial correlation and temporal analysis, have emerged. In spatial correlation, normal behavior is characterized by the correlations between different metrics, or between identical metrics measured at different physical locations. Significant uncorrelated changes are then identified as anomalies. With temporal analysis, the normal behavior of a metric is characterized by analyzing its time series, and significant deviations from the predicted future behavior are flagged as anomalies. The promising methods from these classes are entropy-based principal component analysis (PCA) [1] and hierarchical heavy hitter (HHH) based time series analysis [2] respectively.

The availability of different classes, the various algorithms in each, along with the different network-metrics and tunable parameters leave the network operators with a confusing array of choices. For the automated methods to be truly effective, operators need guidelines about which methods are appropriate for their requirements, and also their ideal parameters. Furthermore, it need not be the case that a particular class, algorithm, or set of parameters is uniformly better than another. It is likely the case that different methods will be better suited at identifying disjointed sets of interesting network events (e.g., port-scans versus denial of service attacks). Comparative studies can not only help identify which methods and metrics are better than others, but can also identify the conditions where the accuracy is compromised in order to help determine the future directions for improvement or to inspire new hybrid approaches. While PCA has been studied extensively [3], [1], [4], [5], we are not aware of any similar evaluation of an HHH-based anomaly detection system that provides operators with guidelines about its use. Furthermore, there is no comparative evaluation of these methods across the temporal and spatial domains, nor between these two methods in particular, to show when and how the accuracy of these methods might differ and where they might be improved.

In this paper, we characterize a temporal-based method that uses wavelets on HHH time series data and then compare its performance against a spatial-based detector using entropy as its metric and PCA for correlation. The intent of this comparison is to enlighten operators as to which methods work better under which scenarios and to show the design considerations that affect the detection performances. Using an evaluation framework based on trace-driven simulation similar to other work [3], [6], [7], we inject anomalies of varying magnitudes and thus compare the detection performance of the methods over a range of operating parameters. In order to evaluate the detections with respect to different types of anomalies, we have designed injection experiments for scenarios that are interesting to network operators: portscans, DDoS, surges in traffic (called boost), and (partial) outages in traffic (called drop).

II. Related Work

Among temporal analysis techniques, time-domain modeling methods, like smoothing and Box-Jenkins ARIMA modeling [8], can potentially face problems in incorporating the periodic behavior present in network traffic (e.g., daily, weekdays, weekends), while frequency-domain methods, such as Fourier analysis, can miss sudden changes in the time domain. A good compromise between both of these methods is wavelets [9], which provides good resolution in both the time as well as the frequency domain. For spatial correlation, PCA is the basic technique [1], [3]. Anomalies can get buried when the traffic-data is aggregated, and hence, methods that operate at finer aggregates have been proposed. Some methods work with heavy-hitters (HHs) as the aggregate of choice; e.g., sketches [10] and hierarchical HHs (HHHs) [2]. HHHs are attractive because

they preserve some of the semantic information (like source, destination and service) of the underlying flows.

The need for rigorous evaluations of anomaly detection systems has recently become apparent [11], [12]. In [13], a comparison of a number of state-of-the-art anomaly detectors is presented, albeit only under portscan attacks. Other studies [7], [6] present comparisons using trace-driven simulation, but [7] compares variations of the same basic method (i.e., analysis of residuals) while [6] compares the utility of different entropy-based metrics. Much interest has been generated in the PCA-based detector, as evidenced by quite a few characterization studies [4], [5].

III. Description of the anomaly detectors

In this paper, we compare two prominent techniques for detecting anomalies in network traffic: HHH-based anomaly detection and entropy-based PCA analysis. Note that we are unaware of any previous work that has used wavelet analysis on HHHs; though wavelet analysis [9] and HHH extraction [14], [15], [2] have been studied independently.

A. Wavelet Analysis of Hierarchical Heavy Hitters

The basic idea here is to divide the traffic into a number of buckets of at least a given size (i.e., identify the HHHs) and then monitor the time-series of these buckets using wavelets. For HHHs, hierarchies on the packet attributes of interest (source, service, etc.) are assumed. After choosing a metric (e.g., flow, bytes), an attribute on which the hierarchy is imposed, and an aggregation-threshold ϕ, the HHHs (which are nodes in the hierarchy that constitute at least ϕ-fraction of the total metric count in a given period) are periodically flushed, giving the time series of the HHHs. The HHHs occurring within all intervals of a training-period are added for monitoring.

To perform wavelet analysis on the time series of each HHH, each time series is profiled and analyzed at different time granularities (e.g., 15 mins, 30 mins, 1 hour, etc.) For each time granularity, Haar-wavelet coefficients indicating changes between adjacent counts are computed and then, Z-statistics is used on the coefficients. A user-specified percentage (called $coverage$), denoting the assumed percentage of normal instances in the data, is used to select the value z_{thr} from all of the z-values of the wavelet coefficients obtained during training. During detection, time intervals corresponding to wavelet coefficients with z exceeding z_{thr} are flagged as anomalous.

In the experiments for each anomaly scenario, attributes that were expected to better detect anomalies were selected. The metrics and attributes used in each scenario (described in Section IV-D) are shown in Table I. Note that we use only 1-D HHHs in this paper.

B. Entropy-based PCA

The entropy-based PCA detector by Lakhina *et al.* [1] aims to identify uncorrelated changes in the entropies of different packet feature distributions (i.e., srcip, dstip, srcp and dstp) corresponding to different O-D flows. The top_k parameter is used to identify the *normal* and *residual* subspaces, and the projection of an observation vector onto the *residual* subspace is used to identify anomalies. Specifically, we use *Z-statistics* on the norms of the *residual* vectors obtained during training to determine z_{thr} for a specified $coverage$ (in a fashion similar to wavelets). While our approach is different from that of [1], it does not affect the ranking of the instances and hence, the Precision-Recall graphs obtained later (see IV-C).

Lakhina *et al.* aggregate their flows into O-D flows, but Ringberg *et al.* [4] show that the level of aggregation of flows (ingress routers, OD flows, input links) affects the sensitivity of PCA. Since it is unclear which level of aggregation will afford a fair comparison with HHH-wavelets, we use PCA without aggregation. Next, while Lakhina *et al.* measure entropy in terms of the packet counts, we measure it in terms of the flow counts as well, and present the results of using this metric too.

TABLE I
Metrics and attributes used by 1-D HHHs for the anomaly scenarios

Scenario	Traffic Metric	Attribute
Portscan	flow	srcip
DDoS	flow, bytes	dstip, dstp
Boost	bytes	dstip, dstp
Drop	flow	srcip, dstp

IV. Description of Experiments and Metrics

In this section, we describe the traffic traces used, our approach for injecting and detecting anomalies, the metrics we use for evaluation, and the anomaly scenarios we have modeled.

A. Data

In order to evaluate the anomaly detection systems, we collected NetFlow [16] data from a live academic network, for the period of Jan-Feb 2008. The network traces contained 5.23 million flow records, and a total of 244 billion packets and 153 TB of data were transferred during this period. The measurements were aggregated into 15 minute bins, giving 5,855 samples of time.

For HHH-wavelets, three weeks were used to collect the HHHs, and each HHH was trained for a further three weeks, starting from the time it was first observed. For entropy-PCA, three weeks were used for training. In both cases, the last two weeks were used for detection.

In order to confirm that the results in some scenarios (portscan and DDoS) were free from biases on the data used, we obtained a 2^{nd}-fold of data by shuffling the training and detection portions of the traces. To support the results in the evaluation in Section V, we show the graphs obtained for one anomaly instance in one fold only. Unless otherwise mentioned, the trends in the graphs are representative of other anomaly instances and other folds (where applicable).

B. Experimental Setup

The mode of operation for both of the methods consists of training over a part of the given traffic trace, followed by detection over the remaining part, resulting in a hypothesized set of anomalies (*hypos*). The *hypos* depend on the method, the method-specific parameters, and the detection threshold, and may be either True Positives (TPs) or False Positives (FPs).

Our basic method of experimentation was trace-based simulation—we injected events of network operator interest onto the traffic-trace described earlier. We limit our interest only to short-term, sudden changes, and as such, the injections are of 15 min durations, and we look for corresponding anomalies in the time-duration of 30mins/1hr only. First, the methods were run over the base traffic without any synthetic injections to obtain the default set of *hypos* (*base anomalies*) detected by the particular method-parameter-configuration. Next, anomalies (*injs*) of different rates were injected and the corresponding *hypos* were identified. The identification of *base anomalies* allowed us to isolate detections resulting from the injections alone. For different parameters of a method and different injection rates, separate injection experiments were carried out.

Care was taken to ensure that *injs* did not overlap with *base anomalies* or with themselves. During detection, the anomalies corresponding to *injs* were identified by looking for an overlap in time, and if applicable, attributes too.

C. Metrics for Comparing Detection Performance

We compare the detections of the methods by their sensitivities as well as using Precision-Recall graphs.

In order to carry out a sensitivity analysis, we estimate the TP-Rate (also known as Recall) for varying magnitudes of an anomaly. This gives an idea of how sensitive the method is towards detecting that anomaly. However, sensitivity analysis by itself is insufficient to compare two methods.

Precision-Recall curves, which have been used to compare the performances of classifiers [17], provide a better way of comparison. In our context, Precision is the proportion of alarms raised by the method that turn out to be true, and hence, the higher the Precision, the lesser manual effort wasted on the part of the operator in following up on the detections provided by the method. Within our framework, we can now estimate both Precision and Recall from the following definitions:

$$\text{Precision} = \frac{\text{\# } \textit{hypos} \text{ corresponding to } \textit{injs}}{\text{\# } \textit{hypos}} \quad (1)$$

$$\text{Recall} = \frac{\text{\# } \textit{injs} \text{ detected}}{\#injs} \quad (2)$$

Note that, in the injection framework, only estimates of these values are obtained. Recall is an estimate because only the information from injections is used while ignoring TPs that may be detected in the base traffic. Similarly, Precision is actually a lower bound because any TPs present in the base traffic are ignored. In order to obtain the Precision-Recall curves, all of the anomalies (and non-anomalies) were ranked using the ordered tuple $(coverage, z)$, and the Precision-Recall values were progressively calculated for the set of tuples. Initially, the set consisted of only the highest-ranked tuple, and then at each step, the next tuple in rank was added to the set and the Precision-Recall values were recalculated. $Coverage$ lower than 99% wasn't used because lower values can increase the number of generated anomalies which are beyond the capacities of network operators.

TABLE II
MAGNITUDES OF ANOMALIES FOR EACH SCENARIO

Scenario	Parameter	Range
Portscan	scan-rate	50, 100, 250, 500, 1000 scans/s
DDoS	bandwidth	100, 500, 1024, 2560, 5120, 10240 Kbps
Boost	% increase	50, 100, 1000, 2500, 3000, 4000 %
Drop	% drop	25, 50, 75, 100 %

D. Injection Scenarios

We have modeled four scenarios that are of interest to network operators: portscan, DDoS, boost and drop. While portscan and DDoS are obvious scenarios, 'boost' models an increase in a subset of the traffic (e.g., applications gaining popularity, routing changes) and 'drop' models scenarios where a subset of the traffic is dropped (e.g., network congestion, new firewall rules for a specific service). All scenarios except boost are modeled by generating/dropping relevant flow records. DDoS and boost modify the byte counts in the relevant flow records. Multiple injections of two anomaly instances from each scenario were carried out and the magnitude of each anomaly instance was varied as shown in Table II.

V. EVALUATION

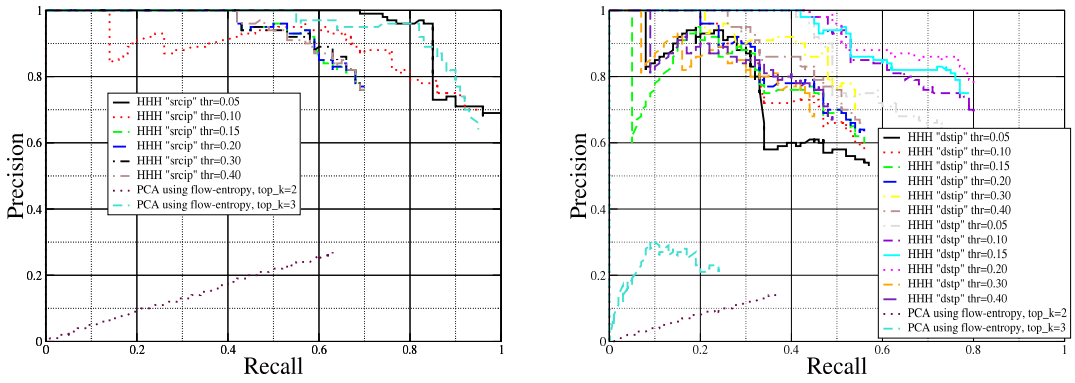
We first present a comparison of the accuracy of the two anomaly-detection methods and then comment on the effectiveness of the parameters and other algorithmic design considerations on the accuracy of the detection methods.

A. Comparison of Accuracy of Detection of HHH-wavelets and Entropy-PCA

The accuracy of detection of the two methods were compared using Precision-Recall graphs for each scenario. For entropy-PCA, the 'entropy w.r.t. packets' metric did not yield good results. For both methods, the respective parameters viz. ϕ and top_k were varied.

Under portscans (Figure 1(a)), neither method outperforms the other consistently across different folds and anomaly-instances: the best cases of HHH and entropy-PCA perform roughly similarly. However, under DDoS (Figure 1(b)), HHHs for all parameters generally perform better than entropy-PCA, with some rare exceptions in the worst-case HHH methods. Under 'boost' and 'drop', the Precision and Recall are so low that no meaningful graphs can be generated.

These results demonstrate two points. First, no existing method is consistently better than the other in detecting all types of anomalies. Next, there are scenarios where neither method does a good job of detecting anomalies.



(a) Portscan (b) DDoS

Fig. 1. The Precision vs. Recall graphs for the DDoS and Portscan scenarios, with different parameterizations, for both compared methods. For Portscans, both methods are comparable, while for DDoS, HHH-wavelets is clearly better than entropy-PCA.

TABLE III
BEST AND WORST CASE MAGNITUDES OF ANOMALIES DETECTED BY THE DIFFERENT METHODS. (N.D. = NOT DETECTED)

Scenario	Attr	HHH Best	HHH Worst	Entr-PCA Best	Entr-PCA Worst
Portscan (scans/s)	srcip	50	250	50	250
DDoS (Mbps)	dstp	0.5	2.5	5	≥10
	dstip	2.5	2.5		
Boost (%)	dstip	2500	N.D.	N.D.	N.D.
	dstp	N.D.	N.D.		
Drop (%)	srcip	≥100	N.D.	N.D.	N.D.
	dstp	N.D.	N.D.		

B. Parameterization and Algorithmic Design Considerations

In this section, we explore a few questions that operators interested in deploying these methods might ask. First, we present a comparison of the methods based on their sensitivities towards detection of anomalies. Next, we try to understand the effectiveness of the method-specific parameters with respect to the detection performance.

1) Sensitivity Analysis: Figures 2-4 show the variation of TP rate of HHHs with the threshold-parameter ϕ as well as the magnitudes of the anomalies, for some of the scenarios, with $coverage$ fixed at 99%. We see that in general, at any threshold, as the magnitude of the anomaly increases, the TPs detected increase. After the magnitude of the anomaly crosses a certain value, nearly all injections are detected irrespective of the threshold used. In such cases, the anomalies are large enough to be detected by the root HHH, and hence essentially correspond to detections by wavelets on the complete aggregation of the traffic.

Figures 5(a) and 5(b) also show the variation of TP rate of entropy-PCA with the magnitudes of anomalies, for different top_k and metrics, for some of the scenarios. Note that entropy-PCA with 'entropy w.r.t. packets' detects next to nothing in all the scenarios.

The best and the worst magnitudes of the anomalies at which the methods provide significant (i.e., $\geq 50\%$) detection rates are summarized in Table III. We see that HHH-wavelets *can* match the sensitivity of entropy-PCA for portscan, and is the more sensitive of the two for DDoS.

2) Effectiveness of ϕ w.r.t. Detection Performance of HHH-wavelets: Intuitively, the aggregation-threshold ϕ provides a parameter with which to tune the behavior of HHH-wavelets. Here, we study its effectiveness with respect to the sensitivity as well as the detection performance of the method.

From Figures 2-4 which show the sensitivity analysis of HHHs, we observe that lowering ϕ *can* enable detection of anomalies of lesser magnitudes i.e., the detection can become more sensitive. However, this is possible only if a HHH corresponding to the anomaly is present lower down in the hierarchy, which cannot always be guaranteed by simply lowering the threshold.

Next, we consider the scenario-wise Precision-Recall graphs (Figures 1(a) and 1(b)) to get a closer look at the performance of the HHH methods at different ϕ-s. While the lowest threshold (0.05) generally performs better, there are exceptions and even opposing trends, with lower thresholds performing relatively better in some instances, and higher thresholds performing relatively better in others. However, this can be explained in terms of the presence of HHHs corresponding to the injected anomalies.

First, we observed that the number of *hypos* and hence, *base anomalies* tends to decrease as ϕ increases. Hence, if different thresholds have the same HHH relevant to the *inj* in the hierarchy, then the method with the higher threshold will perform better because of the resultant higher precision. Second, lower thresholds have a better chance of maintaining a HHH corresponding to the injected anomaly deeper down in the hierarchy, and if such low HHHs exist, then even though the time series of such HHHs are expected to be noisy, the anomalies corresponding to the injections detected in them seem to have higher ranks than *base anomalies*, as well as higher ranks than anomalies detected higher up in the hierarchy. The higher ranks of the anomalies in the low HHHs leads to higher precision values in the beginning of calculation of precision-recall values, as well as better recall later and hence, better overall performance of the lower ϕ-s. Thus, lower thresholds *can* lead to better performance, if there

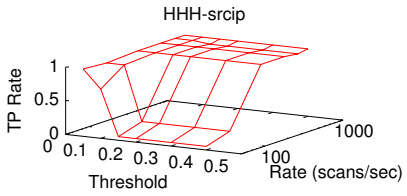


Fig. 2. Portscan: TP Rate vs. Anomaly Rate vs. Threshold

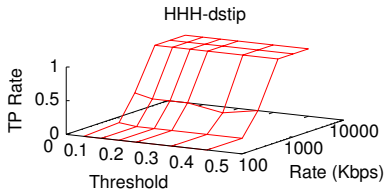


Fig. 3. DDoS(flows): TP Rate vs. Anomaly Rate vs. Threshold

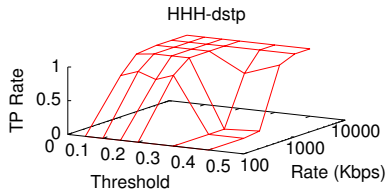
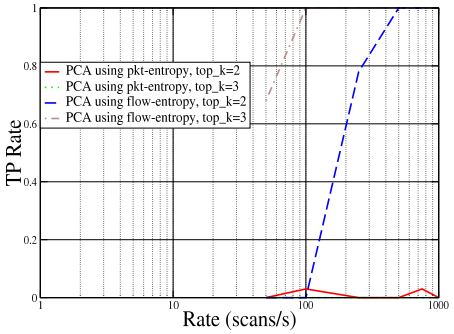
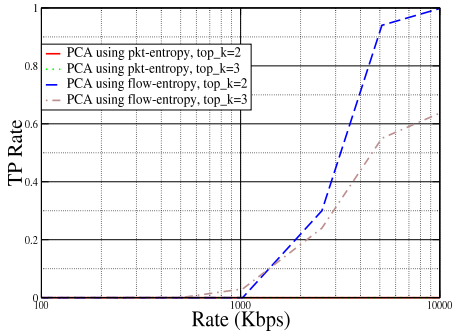


Fig. 4. DDoS(flows): TP Rate vs. Anomaly Rate vs. Threshold



(a) Portscan



(b) DDoS

Fig. 5. Sensitivity analysis of entropy-PCA for different parameters and metrics, for the DDoS and Portscan scenarios.

exist corresponding HHHs lower down in the hierarchy.

3) Effectiveness of top_k w.r.t. Detection Performance of Entropy-PCA: From the entire evaluation, we can see that the performance of entropy-PCA, both in terms of the sensitivity and the precision-recall graphs, is sensitive to small changes in the top_k parameter; this is an extension of the result from [4], which observed that the false-positive rate was sensitive to changes to this parameter.

VI. Conclusions

In this work, we have compared one promising anomaly detection method each from the classes of temporal analysis and spatial correlation: HHH-wavelets, which is a new algorithm that combines the existing techniques of HHHs and wavelets, and entropy-PCA, which is an algorithm already described in the literature. A comparison of the detection accuracy of the methods for different anomaly scenarios shows that: (i) for portscan, no method is better than the other, (ii) for DDoS, HHH-wavelets is better than entropy-PCA and (iii) neither of the methods have significant detections under boost and drop. Additionally, we explore the pragmatic design considerations of the algorithms and obtain the following set of results: (i) under portscan, HHH-wavelets *can* match the sensitivity of entropy-PCA while under DDoS, HHH-wavelets are more sensitive, (ii) we show how lowering ϕ for HHH-wavelets *can* lead to better detection and (iii) we confirm that the effectiveness of entropy-PCA depends strongly on top_k (as was shown in [4]).

References

[1] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies using Traffic Feature Distributions," in *SIGCOMM '05*.

[2] Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund, "Online Identification of Hierarchical Heavy Hitters: Algorithms, Evaluation, and Applications," in *IMC '04*.

[3] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-wide Traffic Anomalies," in *SIGCOMM '04*.

[4] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," in *SIGMETRICS*, 2007.

[5] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for Traffic Anomaly Detection: Problems and Solutions," in *IEEE INFOCOM 2009, Mini-Conference*.

[6] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An Empirical Evaluation of Entropy-based Traffic Anomaly Detection," in *IMC*, 2008.

[7] A. Soule, K. Salamatian, and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection," in *IMC '05*.

[8] G. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, 1976.

[9] A. Ron, D. Plonka, J. Kline, and P. Barford, "A Signal Analysis of Network Traffic Anomalies," *Internet Measurement Workshop 2002*.

[10] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based Change Detection: Methods, Evaluation, and Applications," in *IMC '03*.

[11] H. Ringberg, M. Roughan, and J. Rexford, "The Need for Simulation in Evaluating Anomaly Detectors," *SIGCOMM Comput. Commun. Rev.*, 2008.

[12] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proceedings of the IEEE Symposium on Security and Privacy 2010*, pp. 305–316.

[13] A. Ashfaq, M. Robert, A. Mumtaz, M. Ali, A. Sajjad, and S. Khayyam, "A Comparative Evaluation of Anomaly Detectors under Portscan Attacks," in *RAID*, 2008.

[14] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava, "Finding Hierarchical Heavy Hitters in Data Streams." in *VLDB*, 2003.

[15] ——, "Diamond in the Rough: Finding Hierarchical Heavy Hitters in Multi-dimensional Data," in *SIGMOD'04*.

[16] C. S. Inc., "Netflow services and applications," http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm, 2002.

[17] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 2008.