

# An Empirical Evaluation of Entropy-based Traffic Anomaly Detection

George Nychis, Vyas Sekar, David G. Andersen, Hyong Kim, Hui Zhang  
Carnegie Mellon University

## ABSTRACT

There is considerable interest in using entropy-based analysis of traffic distributions for anomaly detection. These approaches are appealing since they provide more fine-grained insights into traffic structure than traditional traffic volume analysis. While previous work has demonstrated the benefits of entropy-based anomaly detection, there has been little effort to comprehensively understand the detection power of using entropy-based analysis of multiple traffic distributions in conjunction with each other. We consider two classes of distributions: flow-header features (IP addresses, ports, and flow-sizes), and behavioral features (degree distributions measuring the number of distinct destination/source IPs each host communicates with).

Somewhat surprisingly, we observe that the entropies of the address and port distributions are strongly correlated with each other and provide very similar detection capabilities. The behavioral and flow size distributions are less correlated and detect incidents that do not show up as anomalies in the port and address distributions. Further analysis using synthetically generated anomalies also suggests that the port and address distributions have limited utility in detecting scan and bandwidth flood anomalies. Based on our analysis and understanding of the correlations among the different distributions, we discuss important implications for entropy-based anomaly detection.

## 1. INTRODUCTION

There has been recent interest in the use of entropy-based metrics for traffic analysis [32] and anomaly detection [15, 4, 10, 6, 17, 30]. The goal of such analysis is to capture fine-grained patterns in traffic distributions that simple volume based metrics cannot identify. Studies have shown that entropy based methods can detect anomalies that may not manifest as large deviations in traffic volume [15].

Many traffic features (e.g., flow size, ports, addresses) have been suggested as candidates for entropy based anomaly detection. However, there has been little work in understanding the analysis capabilities provided by a set of entropy metrics used in conjunction with one another. For example, it is unknown whether the dif-

ferent features complement each other, or if they detect the same results and are redundant.

The goal of this paper is to provide a better understanding of the use of entropy-based methods in anomaly detection. We consider two types of distributions based on *flow-header* features and *behavioral* features. The flow-header features are addresses (source and destination), ports (source and destination), and the flow size distribution (FSD). These have been suggested as candidates for detecting worms, scans, and DDoS attacks [15, 6, 14]. The behavioral features are the in and out-degree distributions, where the degree of an end-host  $X$  is the number of distinct IP addresses that  $X$  communicates with. The behavioral distributions capture the structure of end-host communication patterns.

To provide insight in to the detection capabilities of the traffic features, we use a month-long traffic trace collected within a large university network as our primary dataset. In addition, we corroborate specific parts of our analysis with other datasets from Internet2 [8], Geant [7], and a (different) university department. Our key results are:

- Across all datasets, port and address distributions are highly correlated, with pairwise correlation scores greater than 0.95. The degree distributions and FSD are weakly correlated with each other and with the port/address distributions.
- The correlation between the source (destination) port and source (destination) address distribution arises due to the nature of the underlying traffic patterns. However, the correlations across the source and destination distributions stem from the uni-directional nature of flow-level measurements available today.
- The anomalies detected by the port and address distributions overlap significantly. In our dataset, almost all the anomalies detected by these distributions are *alpha flows* [15]. In contrast, host degree distributions and FSD identify distinct anomalies (e.g., abnormal scan, DoS, and P2P activity) that are not detected by the port and address distributions.

- Experiments with synthetically generated anomaly scenarios show that FSD and the degree distributions detect scanning events whereas the port and address distributions do not (confirming the trace results). For bandwidth flooding and DDoS events, port and degree distributions detect only high-magnitude events that would have appeared as traffic volume anomalies anyway.

These observations have important implications for entropy-based analysis. First, we should select candidate distributions with care. While ports and addresses have been commonly suggested [15] as good candidates for entropy-based anomaly detection, our measurements question this rationale. Our results also suggest a natural metric for choosing traffic features for entropy based anomaly detection: select traffic distributions that inherently complement one another and thus provide different views into the underlying traffic structure. Second, we need to potentially move beyond the traditional uni-directional flow semantics exported by many tools available today (e.g., Netflow), since it can artificially skew the properties of the underlying distributions. Thus, it is prudent for administrators to use bi-directional flow collection tools whenever possible. Finally, we discuss how to use the correlations to design a better anomaly detection system. Our preliminary results show that using time-series anomaly detection on the correlation scores can expose new anomalies that do not manifest in the raw time-series analysis.

## 2. PRELIMINARIES

First, we define the notion of normalized entropy used for anomaly detection. Next, we briefly describe the timeseries analysis techniques for anomaly detection that we borrow from existing work. We then introduce the seven traffic distributions we evaluate.

### 2.1 Entropy

Let  $X$  denote a random variable representing the distribution of values a particular traffic feature (e.g., the source address or destination port of a flow) can take. Let  $x_1 \dots x_N$  denote the range of values that  $X$  can take, and for each  $x_i$  let  $p(x_i)$  represent the probability that the random variable  $X$  takes the value  $x_i$ , i.e.,  $p(x_i) = Pr[X = x_i]$ . The entropy [25] of the random variable  $X$  is defined as<sup>1</sup>:

$$H(X) = - \sum_{i=1}^N p(x_i) \log p(x_i) \quad (1)$$

**Normalized Entropy:** Let  $N_0$  be the number of distinct items present in a given measurement interval. Intuitively, the entropy is a measure of the diversity of the

<sup>1</sup>All logarithms in this paper are to the base 2 and  $0 \log 0 = 0$

distribution. The entropy attains its minimum value of zero when all the items are identical and its maximum value of  $\log(N_0)$  when each item appears exactly once. Since different measurement intervals might observe a different number of distinct items, we normalize  $H$  to be between zero and one by computing the normalized entropy:  $H / \log N_0$ . This measures the relative randomness within each measurement interval and allows us to quantitatively compare entropy values across time. For the remainder of the discussion we will use this definition of normalized entropy.

### 2.2 Traffic Features

We study seven empirical traffic distributions. Five of these are obtained from flow-header features: source address, destination address, source port, destination port, and the flow size distribution (FSD). The remaining two are based on inter-host communication behavior: the in-degree and out-degree distribution across hosts, where the degree of host  $X$  is the number of distinct IP addresses that  $X$  communicates with.

Prior work on using flow-header features in traffic analysis uses unidirectional flow information commonly exported by most routers [21, 22]. Our dataset contains bidirectional flow records [2]; we explicitly convert each bidirectional flow record into two unidirectional flows. Consider Figure 1 as an example. The first entry would be interpreted as two flows, one flow from 244.0.0.1 to 128.2.240.216 with 6 packets, the other flow from 128.2.240.216 to 244.0.0.1 with 2 packets. The behavioral metrics use *directional* flows to distinguish between incoming and outgoing connections for each host.

#### Examples using Figure 1:

- *Address* 0.2.122.42 sourced 65 packets and was the destination of 77 packets
- *Port* 22 sourced 70 packets and was the destination of 60 packets
- The *traffic volume* is 74 source packets and 76 destination packets
- 3 hosts had an *in degree* of 1, 2 hosts had an *out degree* of 1
- 2 flows had a *flow size* of 6 packets

#### Computing normalized entropy:

- *Addresses:* For each source (destination) IP address  $x_i$ , we calculate the probability

$$p(x_i) = \frac{\text{Number of pkts with } x_i \text{ as src (dst) address}}{\text{Total number of pkts}}$$

The normalization factor is  $\log(N)$ , where  $N$  is the number of active source (destination) addresses observed during the measurement interval.

- *Ports:* For each source (destination) port  $x_i$ , we calculate the probability:

$$p(x_i) = \frac{\text{Number of pkts with } x_i \text{ as src (dst) port}}{\text{Total number of pkts}}$$

StartTime(sec)	EndTime(sec)	Proto	Left_IP:Port	Flow_Dir	Right_IP:Port	Src_Pkts	Dst_Pkts	Src_Bytes	Dst_Bytes
12802	12803	UDP	244.0.0.1:3180	->	128.2.240.216:139	6	2	372	120
12820	12824	TCP	0.2.122.42:9478	->	109.173.146.198:22	29	30	1566	3772
12824	12824	TCP	130.20.143.124:80	<-	0.2.122.42:2118	2	1	846	60
12824	12825	TCP	0.2.122.42:2341	->	109.173.146.198:22	31	40	1662	4201
12825	12825	UDP	244.0.0.1:2718	->	0.2.122.42:139	6	3	372	150

Figure 1: Format of the flow level data used for the analysis showing some sample flow records.

The normalization factor is  $\log(P)$ , where  $P$  is the total number of distinct active source (destination) ports for the interval.

- *Flow size distribution:* For each actual value that flow size (measured in packets) takes, we calculate the probability:

$$p(x_i) = \frac{\text{Number of flows with flowsize } x_i}{\text{Total number of flows}}$$

The normalization factor is  $\log(F)$ , where  $F$  is the number of distinct flow sizes we observe within the measurement interval.

- *Behavioral distributions:* For a host  $X$ , the *out-degree* is the number of distinct IP addresses that  $X$  contacts, and the *in-degree* is the number of distinct IP addresses that contact  $X$ . Each  $X$  is an active internal IP address inside the network under consideration (e.g., in our dataset we only consider hosts inside the university): these are the only hosts for which we have a complete view of both incoming and outgoing traffic. For each value of out-degree (in-degree)  $x_i$ , we calculate the probability

$$p(x_i) = \frac{\text{Number of hosts with out-degree } x_i}{\text{Total number of hosts}}$$

The normalization factor for the out-degree distribution is  $\log(D)$ , where  $D$  is the number of distinct out-degree (similarly in-degree) values observed during the measurement interval.

We bin the traffic stream into fixed-length measurement epochs (e.g., five minutes). For each measurement epoch, we compute the normalized entropy values for the different distributions. We then apply timeseries anomaly detection on the timeseries of normalized entropy values.

### 2.3 Anomaly Detection Methods

Our goal is not to evaluate the timeseries analysis methods themselves; rather, our goal is to demonstrate that our observations regarding the properties of entropy-based anomaly detection are independent of the underlying anomaly detection technique used. We use two independent methods for timeseries anomaly detection. We do so primarily to avoid any biases arising from a particular anomaly detection technique.

**Wavelet analysis:** Barford et al. propose wavelet analysis for detecting anomalies in traffic timeseries data [3].

Their technique treats measurement data (e.g., number of packets per five minute interval) as a generic time-series signal. Using wavelet decomposition, the time-series is decomposed into low, mid-range, and high-frequency components. The low frequency components capture expected traffic trends (e.g., the mean and the diurnal traffic patterns), while the mid to high frequency components capture instantaneous variations in the time-series (i.e., deviations and anomalies from the expected patterns).

After this decomposition, the mid-range and high frequency components are normalized to have unit variance. Next, they compute the local variability of the high frequency and mid frequency components using sliding windows of different sizes. The size of the sliding window determines the duration of anomalies that can be detected. The local variability of the high frequency and mid frequency parts are then added together. The final anomaly detection step is a simple threshold function applied to this combined signal.

**Heuristic threshold-based detection:** As an alternative to wavelet based detection, we consider a heuristic technique adapted from prior work [23, 24]. The high-level goal is to estimate the mean and standard deviation of the timeseries signal using historical data. For each future observation, we compute a deviation score:  $Score = \frac{|Observation - Mean|}{Stddev}$ . This score captures how far away from the mean value a particular observation is, expressed relative to the standard deviation. We flag an anomaly whenever any observation has a score greater than some threshold  $\alpha$ .

If the mean and standard deviation are calculated with historical traffic data that contains large anomalies, then these values are likely to over-estimate the true statistics. Therefore, this heuristic may miss anomalies in future observations, because the model of traffic accommodates more anomalies than it should. To avoid this bias, we introduce an iterative cleaning technique for learning the mean and standard deviation given possibly noisy training data. The approach works as follows. In each iteration, we compute the mean and the standard deviation. For the current iteration, we find anomalous data points, i.e., those that are greater than  $\alpha = 3$  standard deviations away from the mean. We remove these anomalies from consideration for further iterations. The iteration continues until the mean and standard deviation obtained are *stable*, meaning that

the values do not differ significantly across subsequent iterations. In our experiments, we find that this process usually terminates in 4-5 iterations and drops fewer than 3% of the data points.

### 3. MEASUREMENT RESULTS

**Dataset:** Our primary dataset uses Argus [2] flow level data captured in February 2005 at Carnegie Mellon University<sup>2</sup>. The dataset contains traffic to and from over 100,000 active IP addresses involving roughly 92 TB of total traffic over 2.5 billion flows. IP addresses in the dataset were anonymized preserving a one-to-one mapping between actual and anonymized IP addresses [31]. Application ports were not anonymized. The traffic feature distributions we study are unchanged by the anonymization.

The dataset is split into five minute non-overlapping intervals consisting of flows that completed within the interval. Each flow record includes the connection time, the protocol used, the connection state, flow direction, and source/destination pairs for the IP address, port, packet count, and byte count. The format of the data with some example flow records is shown in Figure 1. As discussed earlier, to obtain the in- and out-degree distributions, the flow record must indicate the flow’s direction. However, in some cases the directionality is not evident from the flow record (e.g., UDP flows, long-lived TCP flows that extend beyond the flow timeout). In such cases, we use application port numbers to infer flow direction<sup>3</sup>. If the directionality is still ambiguous, we arbitrarily select the left host in the flow record as the originator.

#### Roadmap:

- We compute pair-wise correlation coefficients for the different entropy timeseries in Section 3.1 to understand if the different traffic feature distributions are structurally correlated.
- We compute pair-wise correlation coefficients for the anomaly deviation scores in Section 3.2.
- We analyze the overlap between the set of anomalies detected by the different traffic features in Section 3.3 to understand if the different entropy metrics provide similar detection capabilities or if they differ significantly.
- We use a heuristic approach for identifying the traffic flows contributing to the anomalies in Section 3.4. By analyzing the anomalous flows, we

<sup>2</sup>The router observes all traffic between university hosts and external Internet hosts. It also observes a significant fraction of internal inter-departmental traffic

<sup>3</sup>Rationale: If a host is running a well-known application service, then it is likely to be the server-host in the connection. Since the client typically initiates a connection to the server in client-server transactions we assume that the host that does not use the well-known port is the originator of the connection.

explain why some of the anomalies are detected by several entropy metrics, while other anomalies are unique to specific metrics. We discover several interesting anomalies, including possible botnet activity, the arrival of a P2P “supernode”, and a possible outbound spoofed DoS attack.

### 3.1 Correlations in Entropy Timeseries

Table 1 shows the pairwise correlation scores between the entropies of different distributions. We find strong correlations ( $> 0.95$ ) between the address and port distributions. The remaining metrics show low or no correlation. Figure 2 shows the entropy timeseries values over the entire month-long trace. The visual confirmation of the correlations is just as striking as the values themselves. Additionally, we observe that many of the spikes and deviations in the timeseries plots are also highly correlated. We will revisit these anomalies in the subsequent discussions.

	Out Deg	Src Addr	Dst Addr	Src Port	Dst Port	FSD
InDeg	0.102	0.100	0.097	0.000	0.007	0.414
OutDeg	-	-0.034	-0.033	-0.054	-0.015	-0.018
SrcAddr	-	-	0.994	0.962	0.956	0.307
DstAddr	-	-	-	0.966	0.969	0.286
SrcPort	-	-	-	-	0.989	0.171
DstPort	-	-	-	-	-	0.181

**Table 1: Correlation of entropy data with no measurement anomalies.**

Trace (#routers)	Date	Avg	SDev	Min	Max
CMU (1)	Mar 1-31, 2008	0.980	0.014	0.962	0.998
GA Tech (1)	Feb 12 - Mar 22, 08	0.943	0.020	0.910	0.968
Internet2 (11)	Dec 1-14, 2006	0.841	0.078	0.761	0.988
GEANT (22)	Nov 1-31, 2005	0.897	0.072	0.812	0.987

**Table 2: Corroborating the correlation scores from other traces**

To confirm that these results are not an artifact of our dataset, we perform similar analysis using data from other networks and time periods: Internet2 [8], GEANT [7], Georgia Tech, and CMU-2008. All the datasets are large, consisting of over a hundred thousand flows per 5-minute bin. Most of the traces cover a month long duration. The Internet2 and GEANT traces consist of flow data from each of the vantage points (11 and 22 respectively). Table 2 summarizes the average and standard deviation of the correlation scores among the ports and addresses.<sup>4</sup> The strong correlations we observe are not unique to one dataset. Further, the correlations have been stable from 2005 to present. For the remainder of the paper, we focus on results from the CMU 2005 dataset.

<sup>4</sup>Internet2, GEANT, and Georgia Tech only provide unidirectional Netflow [21] style flow records. Thus, we cannot repeat the degree analysis on these.

### 3.2 Correlations in Anomaly Deviation Scores

Next, we explore if the correlations in the entropy timeseries values also extend to anomalies. We assign anomaly deviation scores to each data point using the techniques described in Section 2.3. (For wavelet analysis, the score assigned is the magnitude of localized variance computed over a sliding window of size six representing a half-hour interval.)

	Out Deg	Src Addr	Dst Addr	Src Port	Dst Port	FSD
InDeg	0.248	0.199	0.188	0.185	0.156	0.507
OutDeg	-	0.179	0.165	0.143	0.122	0.396
SrcAddr	-	-	0.991	0.971	0.964	0.319
DstAddr	-	-	-	0.970	0.971	0.300
SrcPort	-	-	-	-	0.986	0.256
DstPort	-	-	-	-	-	0.220

**Table 3: Pairwise correlations in the wavelet deviation scores.**

Table 3 shows that the port and address distributions are as strongly correlated in deviation scores as they are in terms of the entropy values. Interestingly, the behavioral features become slightly more correlated to the other metrics. For example, the correlation between out-degree and FSD increases by 0.414 (Table 4). We hypothesize the reason for this increase is that the in and out-degree distributions show more stochastic variations than the other distributions. Thus, they tend to be uncorrelated in terms of the timeseries values. However, the wavelet analysis removes the noisy variations and so the deviation scores are more correlated.

	Out Deg	Src Addr	Dst Addr	Src Port	Dst Port	FSD
InDeg	0.146	0.100	0.091	0.185	0.149	0.093
OutDeg	-	0.213	0.198	0.197	0.137	0.414
SrcAddr	-	-	-0.003	0.009	0.008	0.012
DstAddr	-	-	-	0.003	0.002	0.014
SrcPort	-	-	-	-	-0.002	0.085
DstPort	-	-	-	-	-	0.039

**Table 4: Difference in correlation values between the two timeseries detection approaches**

Threshold-based detection produces similar results. The difference in the correlations between threshold-based and wavelet-based deviation scores are shown in Table 4. The correlations across the entropy based metrics occur regardless of the detection method chosen as Table 4 confirms. In the rest of the paper, we only present results from wavelet analysis for brevity.

### 3.3 Overlap in Anomalies Detected

Our next goal is to understand the overlap between the detection capabilities provided by different entropy metrics.

We generate anomalies by flagging observations with anomaly deviation scores greater than a detection thresh-

	Out Deg	Src Addr	Dst Addr	Src Port	Dst Port	FSD
InDeg	0.137	0.197	0.183	0.154	0.126	0.354
OutDeg	-	0.072	0.071	0.026	0.023	0.213
SrcAddr	-	-	0.946	0.864	0.854	0.141
DstAddr	-	-	-	0.858	0.891	0.123
SrcPort	-	-	-	-	0.934	0.087
DstPort	-	-	-	-	-	0.060

**Table 5: Correlation between anomalies using detection threshold  $\alpha = 3$ .**

Anomaly Type	Affected Metrics	Labels
Alpha Flows (Botnet activity)	Addresses, Ports	B, F-H, K-N, P
Scans	FSD	A, D
P2P Supernode Activity	FSD	O
Spoofed DoS	Degree	C, E, I
Measurement Outage	Inconsistent	J, Q, R

**Table 6: Labeled traffic anomalies**

old  $\alpha$ . After generating the alarms we assign an *anomaly overlap score* as follows. We construct a timeseries of anomaly incidents for each entropy metric, where a timeslot (five minute bin) is assigned a 1 if there is an anomaly reported and a 0 if there is no anomaly. This vector of 0-1 values represents the entire sequence of anomalies for each entropy metric. We compute the correlations for every pair of (binary) anomaly vectors. By construction, two metrics with the same set of anomalies have a correlation score of 1, and metrics with opposite sets of alarms have a correlation score of -1. We generate these pairwise correlation scores for different values of  $\alpha$ .

Table 5 presents the correlation scores for  $\alpha = 3$ . We notice that the correlations in deviation scores correspond to the correlations in the detected anomalies. Further, the overlap is independent of the value of  $\alpha$  (not shown). Even at very low deviation scores, where thousands of alarms are generated, the alarms are just as correlated as using larger thresholds.

### 3.4 Understanding Anomalies In-Depth

Why do the anomalies detected by the port and address distributions overlap and why do FSD and degree distributions provide unique detection capabilities? To answer this, we use a heuristic approach to identify the set of traffic flows that contribute to each anomaly. Once we identify these anomalous flows, we explain the effect the flows have on different traffic distributions, and why these manifest as anomalies in some traffic features but not in others.

We identify eighteen major events, indicated by alphabetical labels in Figure 3, and summarized in Table 6. (Some anomalies span multiple timeslots; we cluster anomalies close in time into a single event.) Among these, we find three measurement anomalies that do not exhibit any common characteristics among the distributions. We discover 4 alpha flows [15] that explain the

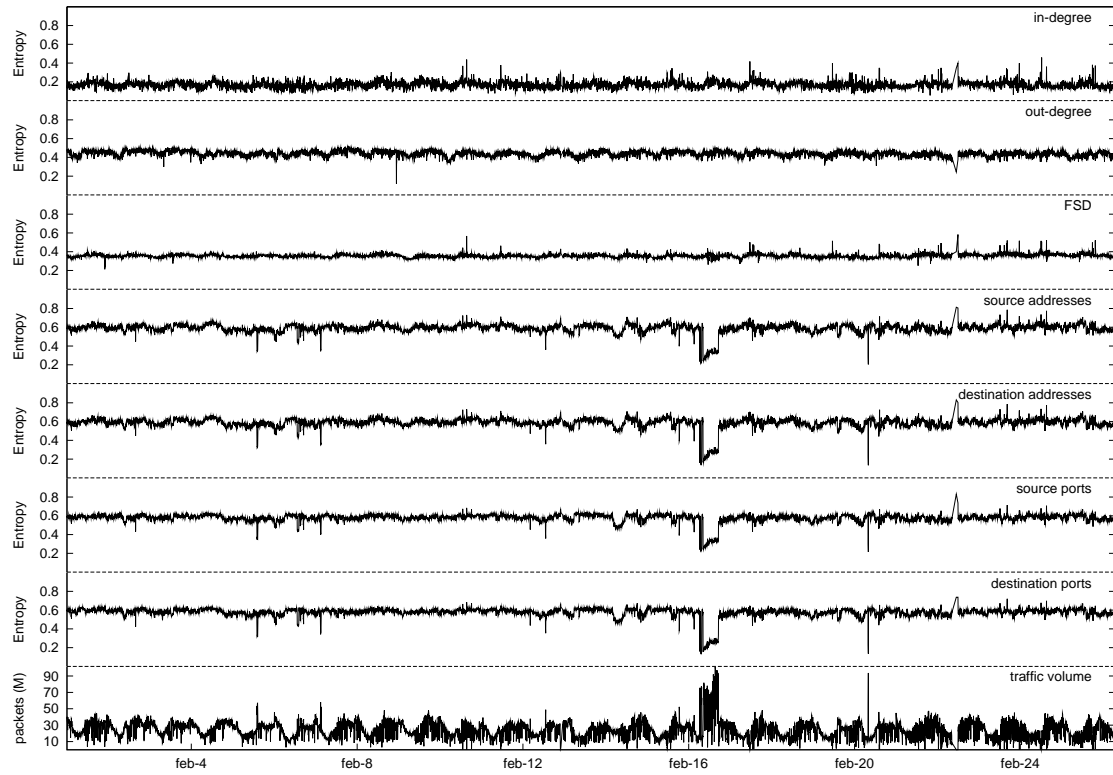


Figure 2: Time series of entropy data for CMU, February 2005

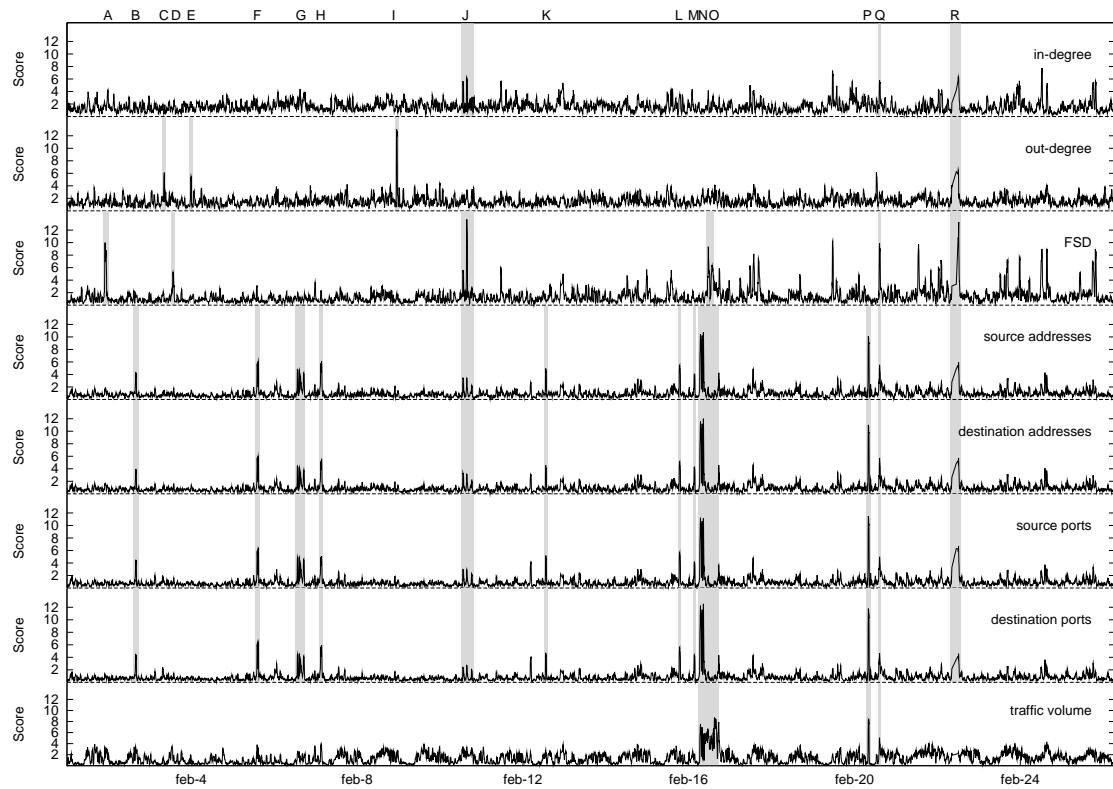


Figure 3: Time series of deviation scores computed with the wavelet analysis technique, labeled with anomalies

strong overlap between the anomalies detected by the address and port distribution. The remaining anomalies are unique to FSD and the in-degree distribution.

**Labeling Method:** In the absence of ground truth for our dataset, we develop a semi-automated approach for discovering anomalous traffic flows. Our approach is a practical heuristic that helps us explain more than 90% of the observed anomalies.

We analyze the *top-k* contributors within each distribution (e.g., destination address, in-degree) for the timeslots when the anomalies occur. The rationale behind this approach is that during an anomaly, the top few contributors to some traffic distributions change significantly. A discrepancy in the top-k set (e.g., if a new value enters the top-k, or if the contribution of the top few values changes significantly), is a possible cause of the anomaly. Next, we identify and remove the traffic flows associated with this discrepancy. We then recompute the entropy and wavelet scores over the remaining flows. If the anomaly subsides, then we posit that these anomalous flows caused the original anomaly.

For example, consider a large bandwidth flood destined to host  $X$ . Since this host receives significantly more traffic than other hosts,  $X$  is likely to appear in the top-k of the destination address distribution. The flows destined to  $X$  are removed and we recompute the entropy and wavelet deviation scores. If the deviation scores for the new entropy values decrease below the detection threshold  $\alpha$ , the bandwidth flood to  $X$  is the likely cause of the anomaly.

**Measurement Anomalies:** Events  $J$ ,  $Q$ , and  $R$  are measurement anomalies, characterized by few or no flow records in our dataset. The measurement anomalies do not show any consistent behavior across the different traffic features.

**Alpha-Flows (likely botnet activity):** In alpha flows, a few ports and addresses (both source and destination) dominate the total traffic volume [15]. This *decreases* the entropy of all the address and port distributions significantly.

Analyzing the flow records, we suspect that these are due to botnet activity: compromised university hosts being used to attack external targets. Events  $F - H$  and  $L - N$ , contain a large volume of UDP traffic destined to a single external host on popular application ports (80,53). These flows are suspicious for three reasons. First, the direction and nature of the traffic is suspicious (we expect web and DNS traffic *to* clients, not *from* them). Second, there is no response traffic from the destination host (it is either overwhelmed or it is dropping these packets). Third, we also found a small amount of TCP traffic on port 6667 (IRC) to and from the sources of the alpha flows just prior to the onset of the alpha flow. The IRC traffic suggests the activity is possibly botnet related (botnets commonly use IRC for

command and control channels). Additionally, eight of these alpha flow incidents share a source IP with another anomaly suggesting that the same compromised host was being used in multiple attacks.

Event  $H$  is particularly interesting in terms of our heuristic for discovering anomalous flows, as it consisted of two independent alpha flow events. Our initial analysis revealed one alpha flow. However, the anomaly persisted after removing the initial alpha flow event; we subsequently discovered the second alpha flow as well.

In our dataset, the alpha flows are the *only* anomalies that the port or address distributions detect. Further, these anomalies are detected by *all* the port and address distributions. This implies, at least in our dataset, that using all the port and address distributions together provides no benefit over using just one of the distributions.

**Peer to Peer Supernode Activity:** Removing the alpha flows for event  $N$  from the traffic does not eliminate a series of concurrent FSD anomalies (collectively labeled  $O$ ). Further analysis of the FSD anomaly reveals that it was caused by an internal host being recruited as a “supernode” in the Kazaa network [11]. During the event, many hosts connect to this supernode creating a significant number of small flows causing a sharp decrease in the entropy of the FSD.

Port and address distributions alone cannot identify two distinct anomalies. This suggests the need to consider distributions that capture different structural properties of the underlying traffic.

**Scan Activity:** In event  $A$ , a single internal host scanned more than 350,000 unique external hosts. The scanner used a fixed source port of 666. The destination ports of the scan are between 0–1024, and each of the 350,000 hosts is scanned exactly once. As there are a large number of small flows, FSD detects the scan. The source address and port distributions do not detect the behavior, as the source does not generate enough packets to bias these distributions significantly. Since only one internal host is involved, the degree distributions are unaffected. Event  $D$  is an outbound scan with a single internal host scanning numerous external hosts on multiple ports. Only FSD detects the scan.

Contrary to conventional wisdom, port and address distributions do not show significant deviations for the scanning anomalies in our data. FSD detects such abnormal scanning activity; we will revisit this in Section 5 using synthetic anomalies.

**Possible DoS using spoofed addresses:** In anomalies  $C$ ,  $E$ , and  $I$ , a very large number of “hosts” (we speculate that these are spoofed) have out-degree 1. This decreases the entropy of the out-degree distribution. The majority of the “hosts” with out-degree 1 connect to the same external destination on port 6667. The set of source addresses in these flows spans the

entire /16 of the university address space. Oddly, the source ports of the flows fall within a small range of port numbers. Also, the destination host sends a single packet response, but the flows seem to terminate abruptly. This leads us to believe that an internal host may be sending attack traffic with spoofed source addresses (within the same subnet) to avoid egress filtering.<sup>5</sup>

The presence of these anomalies unique to the out-degree reiterate the need to choose distributions that provide different views into traffic structure.

### 3.5 Summary of Measurement Results

- The entropy of the port and address distributions (source and destination) are strongly correlated.
- The behavioral distributions (in- and out-degree) and the FSD exhibit low correlations with each other and with the port and address distributions.
- The correlations in the deviation scores mirror the correlations in the entropy values using both anomaly detection methods.
- The anomalies detected by port and address distributions overlap significantly. Further analysis reveals that most of these are alpha flows.
- The degree distributions and FSD detect unique anomalies that are not captured by other distributions. For example, FSD detects interesting scan and P2P behaviors, even when the anomalies are embedded in larger alpha flow anomalies.

## 4. UNDERSTANDING CORRELATIONS IN PORT AND ADDRESS DISTRIBUTIONS

The correlations between the port and address distributions persist across time and across traces from different institutions. We therefore present further analysis to understand the cause and nature of these correlations. As we will see later, understanding the correlations raises important implications for the design of entropy-based anomaly detection systems.

Let us recap how the entropy values were computed in the previous section. First, we obtain the empirical distribution of packet counts over an unidirectional flow abstraction across different values within a specific feature (such as destination ports). Next, we compute the distribution entropy and normalize it by the log of the number of distinct items within the distribution.

The first observation is that the distributions are not correlated simply because they are identical – each distribution is unique. The next question is whether the correlations are an artifact of the way the normalized entropy values are computed. (Note that our approach follows the norm for computing distribution entropy in

<sup>5</sup>Since we only have anonymized flow level traces, we could not further validate this hypothesis.

the literature [15, 32].) We ruled out immediate factors such as (1) entropy normalization, (2) packet vs. byte counts, and (3) time scale of computing correlation values. In all these cases, the correlations are unaffected by the specific data processing steps we take; i.e., the strength the correlations is the same whether we consider (1) normalized or “raw” entropy values, (2) packet or byte counts for computing the empirical distributions, and (3) different time windows to compute the correlation scores (hour vs. day. vs. week).

Once we eliminate these factors, we posit that the correlations are due to (a) a fundamental property of the underlying traffic patterns and/or (b) the unidirectional flow accounting model. For (b), note that this is not specific to our analysis and data. This is often the only type of flow measurements available to network operators today – the notion of bi-directional flow measurements is often ill-defined (e.g., UDP has no connection establishment semantics), and is obscured by practical data collection constraints. For example, packet sampling makes it is hard to identify which IP address originated the connection since the router may not sample the first packet. Similarly, asymmetric routing may mean that a router may not observe both directions of a session and thus cannot generate bi-directional flow reports.

To understand (a), we consider the CMU-2005 dataset since it has additional bi-directional annotations allowing us to eliminate effects from (b). In this case, bi-directional annotations are available because the data is collected at the gateway between the internal network and the Internet, and because the collection infrastructure [2] directly supports bi-directional flows.

### 4.1 Source Addresses vs. Source Ports

**Cause:** *The distributions are structurally similar due to properties of host behavior and network traffic.*

Consider the following hypothetical scenario. The range of source ports (i.e., the possible values that the source port field can take) is infinite. Each host initiates exactly one connection per measurement epoch and chooses a random source port uniformly between 0 and  $\infty$ . In this case, the distribution of source addresses and ports are *identical* and thus the entropy values will be perfectly correlated.

In reality, actual network traffic is different from this hypothetical model: the source port range is limited to  $2^{16}$ , hosts can be involved in more than one connection, and each source port may be chosen by multiple source addresses. However, these factors are not significant enough to cause the distributions to become significantly different and hence uncorrelated.

First, we find that nearly 80% of the hosts in the CMU-2005 trace (68K out of roughly 85K hosts) have a single connection. If source ports are chosen at random,

80% of the contribution to source addresses and source port distributions will be exactly the same – a unique source address will be associated with a unique source port since there are (roughly) 65k unique source ports and 68k unique hosts with a single connection. This means that the finite source port range does not create a significant gap between our hypothetical model and reality.

Next, we consider the effect of multiple hosts selecting the same set of source ports. This is conceptually the same as having a larger number of active unique source addresses relative to the source port range (both increase the number of hosts associated with a specific port). To understand this effect, we replace the actual source ports in the CMU-2005 trace with port numbers chosen uniformly at random within a smaller range. Reducing the source port range from  $2^{16}$  to  $2^{10}$  reduces the correlation between addresses and ports to 0.875; reducing it  $2^8$  still produces a correlation of 0.732. This shows that this effect is not significant enough to reduce the correlation. Thus, even a network which observes significantly more unique source addresses (e.g., Internet2) shows high correlation.

When multiple hosts use the same source ports, the entropies of the port and address distribution are affected similarly, but in differing magnitudes. However, the impact (whether the entropy increases or decreases) on the entropy values is still similar. For example, suppose 1000 new hosts appear in a measurement epoch with each host generating a single connection. This will increase the entropy of source addresses. Even if some of the source ports chosen by these hosts are the same, in aggregate the entropy of the source port distribution still increases. The magnitude of increase relative to the increase in the entropy of the source address distribution will depend on the extent of the overlap in port selection. Since both increase, they are positively correlated – the relative magnitudes of the increase affect the strength of the correlation. As our experiments above with reducing the size of the port range show, the difference in magnitudes does not skew the correlations too much.

## 4.2 Destination Ports vs. Destination Addresses

**Cause:** *Distributions are structurally similar; spread across hosts and application ports is similar to the source distributions.*

The correlation between the entropy of destination addresses and ports is similar to source addresses and ports. (The number of distinct destination addresses and destination ports will be smaller relative the sources. This lowers the absolute entropy compared to the source distributions but it does not affect the correlations.) There are two factors that can potentially skew the correlations: (1) each destination address runs multi-

ple services and (2) most destination addresses run a small number of services. For (1), it is rare for a host to provide more than one service and a single service is often associated with a single port. In our trace, 90% of destination addresses are associated with a single port, 96% are associated with 2 ports or less, and 99% are associated with 3 ports or less.

For (2), if all destination addresses are associated with a small number of application ports (e.g., all servers run on port 80), then the behavior of the destination distributions would differ significantly from the source distributions we considered above. However, the spread of destination addresses and ports is in fact similar to the source distributions. This can be attributed to the large number of network applications with unique destination ports (SVN, iChat, AIM, etc.) and to peer-to-peer applications.

## 4.3 Source vs. Destination Distributions

**Cause:** *The uni-directional flow abstraction used in traffic auditing.*

Source entities and destination entities (e.g., source port vs. destination port) are correlated because of the unidirectional nature of the flow measurements. Each packet contributes to both source and destination pairs. With a strictly bi-directional abstraction, source packets would only contribute to the true source port, and destination packets to the true destination port. With a uni-directional model, if the destination of the connection responds, the packets will be counted toward the destination port *in the source port distribution* as well since the destination now appears as the source in a different uni-directional flow. (Each traffic session between a pair of communicating hosts will manifest as two separate uni-directional flows.) This causes the uni-directional distributions to be approximately the union of their bi-directional pairs.

We use the 2005 CMU trace and recompute the correlations of the entropy values under the bi-directional accounting model. The correlations between source ports and source addresses, as well as destination ports and destination addresses, are not impacted. However, the correlation between a source entity and a destination entity is close to zero (i.e., the entities are uncorrelated). This confirms that the correlation between these pairs is due to the uni-directional flow abstraction.

## 5. USING SYNTHETIC ANOMALIES

In this section, we use synthetically generated anomaly events to complement our measurement results. Table 7 presents a taxonomy of the five synthetic anomalies we evaluate. For each type of anomaly, we are interested in:

1. Can the anomaly be detected using any of the feature distributions?

2. Which feature distribution is most effective for detection?
3. How sensitive is the detection to the magnitude of the anomaly?

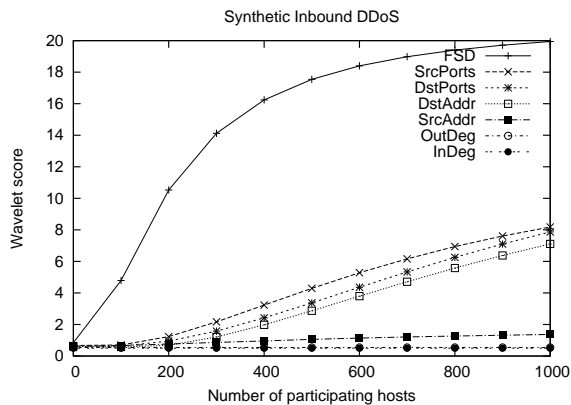
To understand the sensitivity of the detection metrics we vary the scale of the anomaly using a control parameter (e.g., number of source hosts involved in a DDoS or scan attack). In the case of the DDoS and bandwidth floods, we are also interested in comparing entropy-based detection to simple volume based detection. Specifically, we want to identify the smallest anomaly that entropy-based methods detect and understand if, at this magnitude, volume-based analysis would have sufficed.

We fix the duration of the anomaly to be a single epoch (five minutes). For each experiment (i.e., given an anomaly type and the anomaly magnitude), we introduce the anomaly at 50 random locations in the month-long trace. Each data point in the following results represents the mean anomaly deviation score (using wavelet-based detection) over the 50 experiments; the standard deviations were small and are not shown.

For brevity we only consider *inbound* anomalies: the source addresses involved in the incidents are outside the university network and the destination addresses are inside. The results for outbound anomalies are qualitatively similar (except that the roles of in and out-degree get reversed).

### 5.1 Inbound DDoS Flood

**Finding:** *FSD detects DDoS floods at  $6\times$  lower magnitudes compared to port and address distributions.*



**Figure 4: Anomaly scores for inbound DDoS flood**

Each DDoS event is characterized by a single destination address receiving a large volume of single-packet flows (to overwhelm the bandwidth and processing capacity of the server and routers). Figure 4 shows the anomaly scores as a function of the number of hosts

participating in the DDoS attack. Each attack source generates 10 kilobits per second of attack traffic, using a fixed packet size of 57 bytes and a single flow per packet. The attack flows are destined to port 80 on a randomly chosen host inside the university. We have repeated the experiments varying the destination port and the choice of destination address (picking a high-volume, random, and low-volume host) and found similar results.

With just 100 hosts generating 10 kilobits per second of attack traffic each within a five minute interval, there would be  $100\text{hosts} * 300\text{sec} * 10\text{Kbps} / 57\text{bytes} \approx 650,000$  flows of size 1. Thus, the change in the FSD can easily detect the anomaly even with a small number of participating hosts. The destination port and destination address distributions detect the anomaly (i.e., the score is  $\geq \alpha = 3$ ) only when more than 600 attack sources are involved. The anomaly is also detected by source ports since the distribution of traffic across source ports appears random (with 600 hosts participating the normalized source port entropy exceeds 0.97). The degree distributions are unaffected by this anomaly.

### 5.2 Bandwidth Flood

**Finding:** *Destination port and address can detect the anomaly. But, the anomalies that can be detected are large enough that simple volume-based analysis would suffice.*

In a bandwidth flood, a small number of hosts send large amounts of traffic to a single destination. The key differences with respect to the DDoS attack are that the number of hosts involved is an order of magnitude smaller than the DDoS, and that each attack flow is a single high-volume flow. (All attack packets from a single source have the same source port and are thus aggregated into a single flow, as opposed to a DDoS attack where each packet has a random source port.) We vary the number of hosts involved in flooding a single target IP address. The rate of traffic from each host varies uniformly in the range of 300 to 400 Kilobits per second with a fixed packet size of 57 bytes. The target IP is chosen at random within the university with a specific destination port (e.g., port 80 on a webserver). Again, the results were independent of the choice of port and destination host.

Figure 5 shows that the behavioral features, FSD, source port, and source address are unaffected. Since each source generates a single flow (and the size of each flow is random), FSD is not effective at detecting this anomaly. As expected, destination ports and addresses exhibit the greatest deviation. However, a simple back-of-the-envelope calculation suggests that traffic volume can detect this anomaly as well. With 40 hosts (the smallest magnitude that exceeds a detection threshold  $\alpha = 3$  using destination ports) sending 300 Kilobits per second each, an additional 8 million packets are

Anomaly Type	SrcAddr	DstAddr	SrcPort	DstPort	FlowSize
Inbound DDoS Flood	Random	Fixed	Random	Fixed	Fixed (10 Kbps), 1 flow per packet
B/W Flood	Random	Fixed	Random	Fixed	Random (300-400 Kbps), 1 flow per host
Single Scanner	Fixed	Random	Random	Fixed	1-3 packets (10% response rate)
Multiple Scanners	Random	Random	Random	Fixed	1-3 packets (10% response rate)
Port Scan	Fixed/Random	Fixed	Random	Fixed	1-3 packets (10% response rate)

Table 7: Taxonomy of synthetic anomalies used in our evaluation

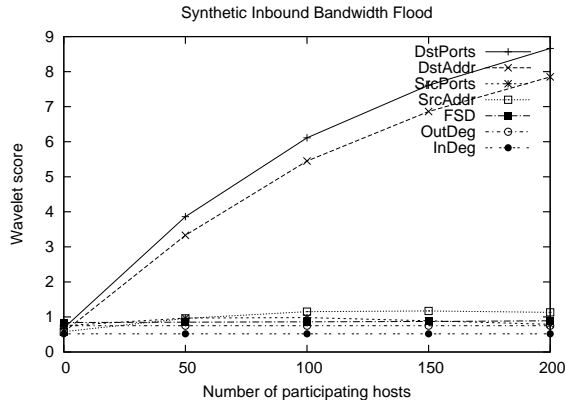


Figure 5: Anomaly scores for inbound bandwidth flood

introduced. In the CMU-2005 dataset, this is roughly 25% of the peak aggregate traffic volume observed in a day, which should be detected by simple volume-based anomaly detection.

### 5.3 Network Scans

Scan traffic is usually made up of a large number of flows to a specific destination port that is being targeted for exploits. We consider two types of scanning activity: a single host scanning the entire university address space and distributed scanning activity from a set of random source addresses. One reason to select addresses and ports as candidate traffic features is that scan traffic typically affects these distributions. We find, however, that these metrics are ineffective for detecting scanning activity. In contrast, FSD and in-degree detect even low magnitude scanning activity.

We set the destination port for the scans to be 445 (associated with many known vulnerabilities [20]). The results are independent of the choice of destination port. To model the properties of real scanning activity (e.g., fraction of hosts that respond and flow sizes of the probes and responses), we sample 10,000 inbound scan flows to port 445 from the traffic trace. We observe that scans receive responses to probes approximately 10% of the time, and that in these cases the probe flows have a flow size of 3 packets, else they are single-packet flows (just a SYN packet).

#### 5.3.1 Single Scanner

**Finding:** Entropy-based anomaly detection cannot de-

tect single scanners with scan rates less than 200 scans per second.

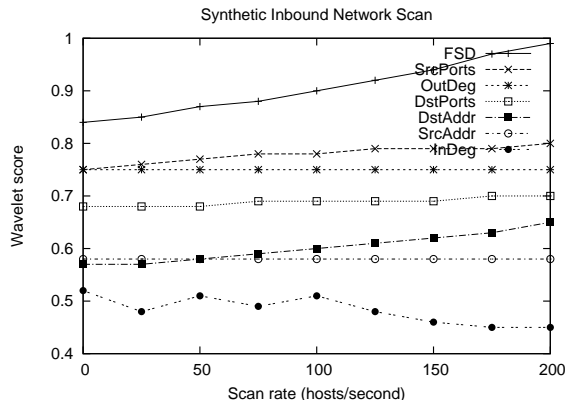


Figure 6: Anomaly scores for a network scan with a single scanning host

Figure 6 shows that a host scanning 200 hosts per second goes undetected. FSD shows the largest deviation, but even it does not generate an anomaly score above 1. Among the other distributions, we observe slight changes in the in-degree (entropy decreases because there is a sudden increase in number of hosts with in-degree 1) and destination addresses (entropy increases since many hosts receive a small volume of traffic and the distribution becomes more uniform). However, the deviations are not large enough to be detected.

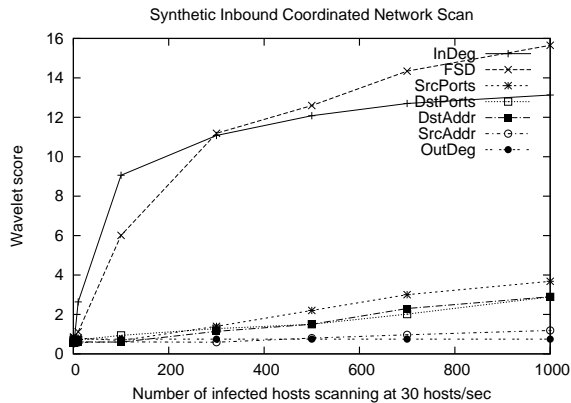
To detect such isolated scan activity, more fine-grained per-host analysis (e.g., flag any host contacting more than  $X$  unique destinations in  $Y$  seconds [1]) and incorporating other aspects of scanning behavior (e.g., failed connections [9]) are necessary.

#### 5.3.2 Multiple Scanners

**Finding:** FSD and in-degree detect scan/worm activity with aggregate rates an order of magnitude lower than port and address distributions.

Such activity is representative of coordinated network scans and worms. In a coordinated scan, multiple hosts (e.g., part of a botnet) scan a particular network. Each participating host scans at a low rate to avoid detection.<sup>6</sup> We fix the scan rate to 30 hosts per second, and

<sup>6</sup>The outbreak of worm activity produces identical behavior. With a random scanning worm, the probability of an incoming scan is  $\text{InfectedHosts} * \text{ScanRate} * \frac{\text{InternalAddressSpace}}{\text{TotalIPAddressSpace}}$ .



**Figure 7: Anomaly scores with the network scan having multiple scanners**

vary the number of hosts generating scanning activity. As in the single scanner case, we assume a 10% response rate.

Figure 7 shows that in-degree and FSD exhibit the strongest deviations. Even with a small number (under 100) of scanners, in-degree and FSD report anomaly scores greater than 9 and 6 respectively. Intuitively, we expect the destination port distribution to identify such scanning activity because the port is invariant across the scan traffic. Surprisingly, it does not do so even when there are 1000 scanners.

FSD and in-degree have another favorable property: they detect scans independent of the ports used by the scan traffic. This implies that even when the vulnerability exists on an application associated with a lot of normal traffic, FSD and in-degree still detect the abnormal scanning activity. However, using the destination port distribution will not detect such scanning activity since the change in the traffic on the popular application port relative to the normal traffic will be small.

## 5.4 Port Scan

**Finding:** *FSD alone is effective for port scan detection.*

We also explored several synthetic port scans. The results are similar to the network scans, except that the degree-based metrics are ineffective. A single scanner with a moderate scan rate (30 scans per second) is not detected by any of the entropy metrics (as in Figure 6). With an increased scan rate ( $> 1000$  scans per second) or with multiple scanners, FSD is the only metric that detects the port scan. The port, address, and degree distributions remain unaffected even with a large number of scanners or a high scan rate.

## 6. IMPLICATIONS

**Choice of features:** Our analysis suggests that the selection of traffic distributions in entropy-based anomaly

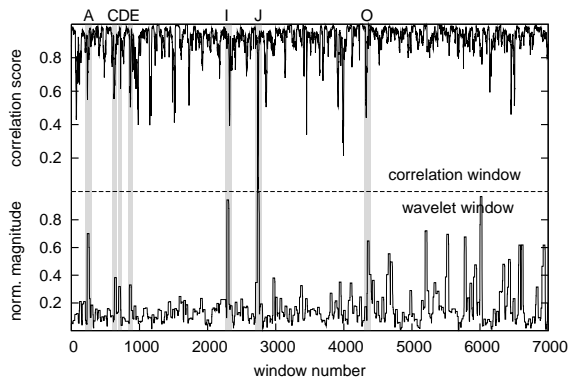
detection should be made judiciously, and in particular we should look beyond simple port and address based distributions. The results also suggest a natural approach for choosing traffic features for entropy based anomaly detection: select traffic distributions that complement one another and provide different views into the underlying traffic structure. For example, the behavioral distributions and the FSD, which are qualitatively different from the port and address distributions, provide distinct and often better anomaly detection capabilities. These complementary distributions can also detect multiple anomalies that occur at the same time.

**Computing distributions:** Section 4 shows that a uni-directional traffic accounting can introduce biases in computing traffic feature distributions. Obtaining bi-directional measurements may involve additional overhead and instrumentation of current traffic monitoring infrastructure, but recent thrusts for bidirectional flow export [28] may help. This may be difficult in large networks when traffic monitors are distributed across the network or when the traffic rates are high enough that sampling is necessary. For enterprise networks, however, the monitor is often a single vantage point co-located with the border router; in this case, bidirectional semantics are easier to obtain and should be preferred.

**Leveraging correlations for anomaly detection:** The stability of correlations in the entropy values during normal time periods suggests a new anomaly detection technique. We compute the correlations over a finite time window  $T$  and detect anomalies over the timeseries of correlation values (computed over a sliding window of size  $T$ ).

In the CMU-2005 dataset, we observe that almost all non-trivial anomalies significantly decrease the entropy correlations. This is shown in Figure 8. We fix  $T = 2$  hours to compute two timeseries of correlation scores: between source address/port and destination address/port. For clarity of illustration, we combine the two timeseries by taking the minimum correlation score across both timeseries. To provide insight in to whether the correlation scores can be leveraged to detect anomalies unique to the other distributions, we compare the correlation timeseries with the maximum wavelet deviation scores of the degree and FSD features.

We include the event labels from Section 3.4 for comparison. Interestingly, nearly all the labeled FSD/degree incidents (the other spikes are measurement anomalies) also exhibit correlation drops in the port and address distributions. Recall from Section 3.4 that these events are not detected as anomalies using the port and address entropy values. Exploring this observation to strengthen anomaly detection is an interesting avenue for future work.



**Figure 8: Timeseries of correlation scores (between ports and addresses) vs. deviation scores over union of FSD and degree distributions**

## 7. RELATED WORK

Network anomaly detection is a broad area of active research. Well known techniques for time-series analysis include the use of wavelets [3], time-series forecasting [23], and other signal processing techniques [27]. These techniques focus primarily on detecting deviations from an expected norm and are basic building blocks for several anomaly detection schemes.

The use of entropy and distributions of traffic features has recently received a lot of attention. Feinstein et al. [6] consider the use of entropy of the distribution of source addresses seen at a network ingress point for DDoS detection. Lakhina et al. [15] augment their PCA framework with entropy based metrics and show that this detects anomalies that cannot be identified using volume based analysis alone. These approaches show the promise of entropy-based anomaly detection. Our work seeks to better understand the selection of traffic feature distributions for entropy based anomaly detection. Since worm payloads contain common substrings that appear frequently, Karamcheti et al. [10] use distributional analysis over packet contents to detect malicious behaviors. We focus on traffic features that can be inferred from flow level information.

Many traffic monitoring and intrusion detection applications use entropy based metrics. Lee and Xiang [17] propose information-theoretic measures for intrusion detection. Entropy has also been used in other contexts in network diagnosis to automatically cluster traffic patterns [32] and to analyze the effectiveness of network monitoring solutions [18]. Wagner et al. [30] use entropy for fast detection of worms by evaluating the compressibility of flow data during attacks.

Many tools [29, 26] and generative models [12] generate realistic traffic workloads for simulation and testing. The correlations we discover between the entropy values of the port and address distributions, both in the university dataset and in the Internet2 dataset, may

have additional implications for such studies that aim to understand structural properties of IP traffic.

There is a large literature related to the accuracy of estimating distributional properties. These include the work on streaming algorithms for estimating the flow size distribution [14] and distribution entropy [16]. Brauckhoff et al. [4] evaluate how packet sampling affects the fidelity of entropy based anomaly detection and show that sampling does not affect the accuracy of detecting the Blaster worm [20]. In a separate study, Mai et al. [19] suggest that packet sampling degrades performance significantly for anomaly detection. These are orthogonal to our measurement study – our focus is on better understanding the selection of traffic feature distributions for entropy based anomaly detection.

Other research efforts in the space of fine-grained anomaly detection have focused on measurements and analysis of backscatter and honeypot data [33], identifying heavy-hitters [5, 34], fast scanner detection [9] etc. These techniques are complementary to entropy-based analysis of flow level measurements.

## 8. CONCLUSIONS

There is considerable interest in deriving fine-grained traffic metrics for anomaly detection and other traffic classification applications. Entropy-based metrics have been suggested as possible candidates to meet this need. The goal of this measurement study was to better understand the anomaly detection capabilities provided by different entropy based metrics.

We evaluate two classes of entropy based metrics: five based on flow header features and two based on properties of end-host behavior. We find that the port and address distributions are strongly correlated both in their raw values and detection capabilities. The behavioral metrics (in- and out-degree) and the flow size distribution provide detection abilities that are distinct from other distributions. Using synthetically generated anomalies we further confirmed that the port and address distributions have limited utility in anomaly detection: they are ineffective for scanning anomalies, and the flood anomalies they detect are large enough to show as volume anomalies.

Our results have two implications for entropy-based anomaly detection. First, we should look beyond simple port and address based traffic distributions for fine-grained anomaly detection. In particular, we should consider distributions that complement each other in their detection capabilities. Second, to avoid the biases arising from uni-directional auditing, it is prudent to use bi-directional flow abstractions for computing traffic distributions whenever possible. Two interesting directions of future work are: (1) exploring information-theoretic measures for deciding traffic features [13], and (2) leveraging the observed correlations to complement

traditional entropy-based anomaly detection.

## 9. REFERENCES

- [1] Snort. <http://www.snort.org>.
- [2] Argus. <http://qosient.com/argus/>.
- [3] BARFORD, P., KLINE, J., PLONKA, D., AND RON, A. A signal analysis of network traffic anomalies. In *Proc. of IMW* (2002).
- [4] BRAUCKHOFF, D., TELLENBACH, B., WAGNER, A., LAKHINA, A., AND MAY, M. Impact of traffic sampling on anomaly detection metrics. In *Proc. of ACM/USENIX IMC* (2006).
- [5] ESTAN, C., SAVAGE, S., AND VARGHESE, G. Automatically Inferring Patterns of Resource Consumption in Network. In *Proceedings of ACM SIGCOMM* (2003).
- [6] FEINSTEIN, L., SCHNACKENBERG, D., BALUPARI, R., AND KINDRED, D. Statistical Approaches to DDoS Attack Detection and Response. In *Proc. of DARPA Information Survivability Conference and Exposition* (2003).
- [7] Geant. <http://www.geant.net/>.
- [8] Internet2. <http://www.internet2.edu>.
- [9] JUNG, J., PAXSON, V., BERGER, A. W., AND BALAKRISHNAN, H. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proc. of the IEEE Symposium on Security and Privacy* (2004).
- [10] KARAMCHETI, V., GEIGER, D., KEDEM, Z., AND MUTHUKRISHNAN, S. Detecting malicious network traffic using inverse distributions of packet contents. In *Proc. of ACM SIGCOMM MineNet 2005* (2005).
- [11] Kazaa. [www.kazaa.com](http://www.kazaa.com).
- [12] KOHLER, E., LI, J., PAXSON, V., AND SHENKER, S. Observed Structure of Addresses in IP Traffic. In *Proc. of IMW* (2002).
- [13] KOLLER, D., AND SAHAMI, M. Toward optimal feature selection. In *Proc. of ICML* (1996).
- [14] KUMAR, A., SUNG, M., XU, J., AND WANG, J. Data streaming algorithms for efficient and accurate estimation of flow distribution. In *Proc. of ACM SIGMETRICS* (2004).
- [15] LAKHINA, A., CROVELLA, M., AND DIOT, C. Mining anomalies using traffic feature distributions. In *Proc. of ACM SIGCOMM* (2005).
- [16] LALL, A., SEKAR, V., XU, J., OGIHARA, M., AND ZHANG, H. Data streaming algorithms for estimating entropy of network traffic. In *Proc. of ACM SIGMETRICS* (2006).
- [17] LEE, W., AND XIANG, D. Information-theoretic measures for anomaly detection. In *Proc. of IEEE Symposium on Security and Privacy* (2001).
- [18] LIU, Y., TOWNSLEY, D., YE, T., AND BOLOT, J. An information-theoretic approach to network monitoring and measurement. In *Proc. of IMC* (2005).
- [19] MAI, J., CHUAH, C.-N., SRIDHARAN, A., YE, T., AND ZANG, H. Is sampled data sufficient for anomaly detection. In *Proc. of ACM/USENIX IMC* (2006).
- [20] MORRISON, J. Blaster revisited. *ACM Queue* vol. 2 no. 4, June 2004.
- [21] Cisco Netflow. <http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml>.
- [22] PHAAL, P., PANCHEN, S., AND MCKEE, N. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks .
- [23] ROUGHAN, M., GREENBERG, A., KALMANEK, C., RUMSEWICZ, M., YATES, J., AND ZHANG, Y. Experience in Measuring Internet Backbone Traffic Variability: Models, Metrics, Measurements and Meaning. In *Proceedings of International Teletraffic Congress (ITC)* (2003).
- [24] SEKAR, V., DUFFIELD, N., VAN DER MERWE, K., SPATSCHECK, O., AND ZHANG, H. LADS: Large-scale Automated DDoS Detection System. In *Proc. of USENIX ATC* (2006).
- [25] SHANNON, C. A mathematical theory of communication. *Bell System Technical Journal* 27 (July 1948), 379–423.
- [26] SOMMERS, J., AND BARFORD, P. Self-configuring network traffic generation, 2004.
- [27] THOTTAN, M., AND JI, C. Anomaly Detection in IP Networks. *IEEE Trans. on Signal Processing* 51, 8 (Aug. 2003), 2191–2204.
- [28] TRAMMELL, B., AND BOSCHI, E. Bidirectional Flow Export Using IP Flow Information Export (IPFIX). RFC 5103, 2008.
- [29] VISHWANATH, K., AND VAHDAT, A. Realistic and responsive network traffic generation. In *Proc. of ACM SIGCOMM* (2006).
- [30] WAGNER, A., AND PLATTNER, B. Entropy Based Worm and Anomaly Detection in Fast IP Networks. In *14th IEEE International Workshops on Enabling Technologies, Infrastructures for Collaborative Enterprises (WET ICE 2005)* (2005).
- [31] XU, J., FAN, J., AMMAR, M. H., AND MOON, S. B. Prefix-preserving IP Address Anonymization: Measurement-based Security Evaluation and New Cryptography-based Scheme. In *Proc. of IEEE ICNP* (2002).
- [32] XU, K., ZHANG, Z., AND BHATTACHARYYA, S. Profiling internet backbone traffic: Behavior models and applications. In *Proc. of ACM SIGCOMM* (2005).
- [33] YEGNESWARAN, V., BARFORD, P., AND ULLRICH, J. Internet intrusions: Global characteristics and prevalence. In *Proc. of ACM SIGMETRICS* (2003).
- [34] ZHANG, Y., SINGH, S., SEN, S., DUFFIELD, N., AND LUND, C. Online detection of hierarchical heavy-hitters. In *Proc. of IMC* (2004).