Computer Networks 55 (2011) 3485–3502

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Accurate network anomaly classification with generalized entropy metrics

Bernhard Tellenbach [a,*], Martin Burkhart [a], Dominik Schatzmann [a], David Gugelmann [a], Didier Sornette [b]

[a] ETH Zurich, ETZ Building, Gloriastrasse 35, 8092 Zurich, Switzerland
[b] ETH Zurich, KPL Building, Kreutzplatz 5, 8032 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 5 August 2010
Accepted 7 July 2011
Available online 21 July 2011

Keywords:
Anomaly detection
Anomaly classification
Generalized entropy
Traffic entropy spectrum (TES)
Tsallis
Netflow

ABSTRACT

The accurate detection and classification of network anomalies based on traffic feature distributions is still a major challenge. Together with volume metrics, traffic feature distributions are the primary source of information of approaches scalable to high-speed and large scale networks. In previous work, we proposed to use the Tsallis entropy based traffic entropy spectrum (TES) to capture changes in specific activity regions, such as the region of heavy-hitters or rare elements. Our preliminary results suggested that the TES does not only provide more details about an anomaly but might also be better suited for detecting them than traditional approaches based on Shannon entropy. We refine the TES and propose a comprehensive anomaly detection and classification system called the *entropy telescope*. We analyze the importance of different entropy features and refute findings of previous work reporting a supposedly strong correlation between different feature entropies and provide an extensive evaluation of our entropy telescope. Our evaluation with three different detection methods (Kalman filter, PCA, KLE), one classification method (SVM) and a rich set of anomaly models and real backbone traffic demonstrates the superiority of the refined TES approach over TES and the classical Shannon-only approaches. For instance, we found that when switching from Shannon to the refined TES approach, the PCA method detects small to medium sized anomalies up to 20% more accurately. Classification accuracy is improved by up to 19% when switching from Shannon-only to TES and by another 8% when switching from TES to the refined TES approach. To complement our evaluation, we run the entropy telescope on one month of backbone traffic finding that most prevalent anomalies are different types of scanning (69–84%) and reflector DDoS attacks (15–29%).

1. Introduction

Entropy-based anomaly detection (AD) has enjoyed substantial attention of the research community in recent years [1–6]. The attractiveness of entropy metrics stems from their capability of condensing an entire feature distribution into a single number and at the same time retaining important information about the overall state of the distribution. Thus, it is possible to scalably detect concentration and dispersion of feature distributions that are typical for certain types of attacks, e.g., DDoS attacks or worm outbreaks.

Compared to merely detecting an anomalous state, it is significantly harder to *classify* an ongoing anomaly and identify its root cause. Attempts of combining changes in multiple features to establish anomaly patterns are very promising (e.g., [2]), but the accurate automatic classification of anomalies is still a major challenge, especially if anomaly sizes and affected host populations vary. Therefore, we have proposed to use generalized entropy metrics, such as the Tsallis entropy, which allow to focus on specific areas of distributions [6], for instance the area of heavy-hitters or rare elements. By doing this, we retain the

* Corresponding author.
E-mail addresses: betellen@ethz.ch (B. Tellenbach), martibur@ethz.ch (M. Burkhart), schadomi@ethz.ch (D. Schatzmann), gugdavid@ethz.ch (D. Gugelmann), dsornette@ethz.ch (D. Sornette).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.07.008

advantages of entropy metrics in general but get additional information about the nature of changes that helps distinguishing anomalies. Specifically, we proposed to use the traffic entropy spectrum (TES), which for a single feature evaluates the Tsallis entropy for different values of its characteristic parameter q. Together, these values form the entropy spectrum. We found that the TES is useful for visually matching occurring patterns against known patterns learned from existing anomalies. However, the suitability of TES for large-scale automatic detection and classification has not been evaluated.

In this paper, we build and extensively evaluate a complete anomaly detection and classification system we call the *entropy telescope*. The entropy telescope integrates several components, such as the TES, SVM based pattern-matching, and several detection approaches such as the Kalman filter [7], PCA [1], and KLE [3] (see Fig. 1). Furthermore, we develop TES_p, an improved version of TES that removes internal correlation by pruning feature distributions. As part of the initial traffic feature selection process, we revisited recent results regarding feature correlation in entropy-based AD [8]. We performed a detailed correlation analysis of a broad set of traffic features and found no persistent strong correlations. On the contrary, we show that extending the classical feature set with additional features, such as AS numbers, country code, and flow sizes helps both detection and classification.

We rigorously evaluated the entropy telescope with a combination of simulation and real background traffic. We share the concerns regarding AD evaluation practice expressed in [9] and avoid ground truth identification by manual labeling. Instead, we developed a rich set of diverse flow-level anomaly models inspired by real anomalies. These models allow to vary parameters and to abstract from a specific instance of an anomaly to a broader class, e.g., DDoS attacks of a certain type. Using FLAME [10], it is possible to inject our anomalies to arbitrary trace files. Reproducibility and fair comparison of methods is crucial for scientific progress. For these reasons and to foster further research in this direction, we make the set of anomaly models designed for this study publicly available [11]. Furthermore, we provide access (on request) to the labeled timeseries data along with a MATLAB toolset to process them. Some of the most important findings related to the evaluation of the entropy telescope are that when switching from Shannon to the refined TES approach, the PCA method detects small to medium sized anomalies up to 20% more accurately. The classification accuracy is improved by up to 19% when switching from Shannon-only to TES and by another 8% when switching from TES to the refined TES approach. Finally, to complement the evaluation with injected anomalies, we ran the entropy telescope on a 34 days trace from a backbone network and report on the prevalence of traffic anomalies. In summary, the most prevalent anomalies found in this trace were different types of scanning (69–84%) and reflector DDoS attacks (15–29%).

The remainder of the paper is organized as follows. In Section 2, we describe our data set, the traffic features we use, and the anomaly models we designed. In Section 3, we describe the different components of the entropy telescope in detail before we evaluate the detection and classification accuracy of several techniques in Section 4. Related work is discussed in Section 5 and the paper is concluded in Section 6. Finally, in Appendix A we present the detailed feature correlation analysis that refutes findings of previous work and guided our feature selection process.

2. Methodology

In this section we introduce the traffic traces and basic concepts, such as the Tsallis entropy. Furthermore, we detail our anomaly models and the anomaly injection procedure.

2.1. Data set

For our evaluation, we use Netflow data captured from SWITCH [12], a medium-sized backbone operator that connects several universities, research labs, and governmental institutions to the Internet. For analyzing the prevalence of real-world anomalies, we use a period of 34 days from

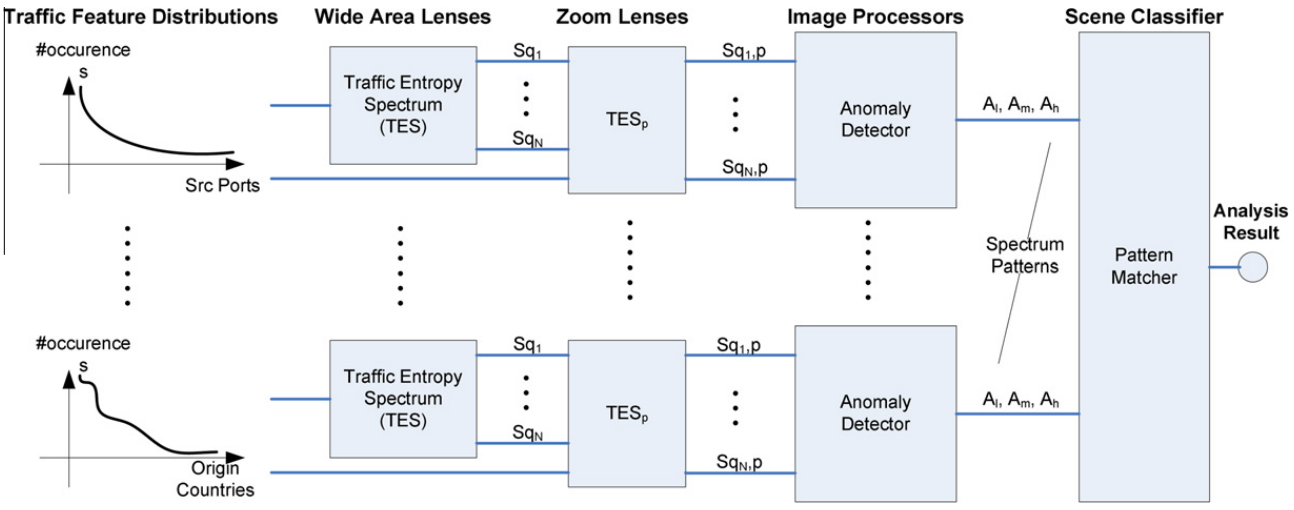


Fig. 1. Entropy telescope building blocks.

07/31/2008 until 09/02/2008 (see Section 4.3). For evaluating the entropy telescope with injected anomalies, we use one week of the month-long trace from 08/09/2008 0:00 am to 08/15/2008 11:59 pm.

The SWITCH network is a stub AS with an IP address range containing about 2.4 million addresses which we refer to as *internal* address space. External addresses are all addresses not assigned to the network's range. Accordingly, we use the term *incoming* traffic to denote flows from external source to internal destinations and *outgoing* traffic for the reverse direction. The flows are collected from four different border routers which do not apply sampling or anonymization. Note that sampling and anonymization can skew certain parts of feature distributions. For instance, deletion of least significant 11 IP address bits, as applied to Abilene traces [2], corresponds to an aggregation of IP addresses at the /21 subnet level. As a consequence, a large number of hosts with just a few flows per host aggregated under the same IP address is indistinguishable from a single host with many flows.

2.2. Tsallis entropy

The Tsallis entropy is a parameterized form of entropy that allows to focus on different *regions* of a distribution. It has recently been shown to have advantages over the Shannon entropy for the detection of network anomalies [6,5].

Let X be a random variable over the range of values $x_1,\ldots,x_n$ and $p(x_i) = p(X = x_i)$. Then, the Tsallis entropy is defined as follows:

$$S_q(X) = \frac{1}{q-1}\left(1 - \sum_{i=1}^{n} p(x_i)^q\right), \quad (1)$$

$$p(x_i) = \frac{a_i}{\sum_{j=1}^{n} a_j}, \quad (2)$$

where q is a parameter specific to the Tsallis entropy and a_i is the number of occurrences or *activity* of x_i in a time window of length T. In our context, the x_i are the feature elements, e.g., specific IP addresses or port numbers.

For $q \to 1$, S_q recovers the Shannon entropy (up to a multiplicative constant). Note that only elements occurring at least once contribute to the entropy S_q of a specific time bin. In the literature, q is referred to as a measure for the non-extensitivity of the system of interest. However, we do not use Tsallis entropy in an information-theoretic sense but rather in an operational sense as a metric measuring whether a distribution is concentrated or dispersed. The main difference to approaches using Shannon entropy in the same manner is that Tsallis entropy allows to concentrate on different regions of the distribution. We discuss this aspect of Tsallis entropy in more detail in Section 3.1.

2.3. Entropy features

In addition to packet, flow, and byte count, we compute the entropy of different traffic feature distributions. We define the following basic set of traffic features:

- ***Shannon classic*** (SHN_C): The Shannon entropy of the source/destination port and the source/destination IP address distribution.
- ***Shannon+*** (SHN_+): The same traffic features as in SHN_C but extended with the Shannon entropy of the following additional feature distributions:
  - autonomous system (AS) distribution,
  - country code distribution,
  - average packet size per flow distribution,
  - flow size distribution.
- ***Tsallis sets*** (TES_p): Based on the same feature distributions as SHN_+.

For AS numbers and country codes, the distribution is always computed from external addresses only, as we have data from a single stub AS.

To justify the selection of these features, we did a detailed analysis of whether it is necessary and/or useful to use all of the 7 (11) features in SHN_C(SHN_+). All the more because Nychis et al. recently raised a concern regarding the correlation of different feature entropies [8]. They study the pairwise correlation of different feature entropies over time, such as the entropy of node degree, flow sizes, IP addresses and port distributions. They found that port and address entropy are highly correlated with Pearson correlation scores greater than 0.95. To investigate this question further, we performed our own comprehensive correlation analysis. Our findings suggest that different feature entropies *do indeed provide useful information*.

We believe the differences between our results and the findings of Nychis et al. can largely be explained by the way the a_i (number of occurrences of element i) are calculated in (2). Nychis et al. compute a_i by counting the number of *packets* containing element i whereas we count the number of *flows* in accordance with other studies [6,2,1,13,3]. Clearly, the number of packets is highly correlated with overall traffic volume, whereas a high volume file transfer is usually summarized in a single flow. Thus, by computing the a_i using packet counts, one introduces a high correlation with traffic volume, and, in turn, also a pairwise correlation between different feature entropies. Our detailed analysis of entropy feature correlation is found in Appendix A.

2.4. Anomaly models and injection

To evaluate the accuracy and sensitivity of the anomaly detector and the anomaly classifier component, we injected artificial anomalies into one week of real background traffic using FLAME [10]. This approach has two main advantages. First, it provides well-defined ground truth independent of an expert labeling the events. Second, it allows to inject the same type of anomaly in different scales, with different parameters, and at different offsets. Thus, the evaluation is not biased by the very set of anomalies accidentally present in a collected trace [9]. However, for background traffic, we chose to use real instead of simulated traffic to get more realistic results. The main problem with real background traffic is that it potentially contains anomalies for which we do not know the ground truth. Therefore, we first inspected the background traces

for existing anomalies by searching for heavy outliers in each traffic feature using a robust statistical outlier definition [14] based on the interquartile range. Where obvious anomalies were found, we labeled the traces accordingly and did not consider the corresponding time bins for injection and validation. To mitigate the effect of smaller anomalies still present in the trace, we injected each anomaly at different random locations.

Previous work argues that concentrated activity on few elements (e.g., the victim of a DDoS attack) leads to a decrease in entropy and dispersed activity (e.g., the spoofed source addresses of the same DDoS attack) leads to an increase in entropy [2,5,6]. However, this is not necessarily true. The precise effect on the entropy metric depends on the activity of the elements contributed by both, the normal traffic and the anomalous traffic and whether and how the two element sets overlap. Therefore, we explicity consider, for instance, set of active and inactive IP addresses.

The 20 base anomalies listed in Table 1 are variations of DDoS attacks, worm outbreaks, scans and P2P outages. Each combination of base anomaly and intensity was injected in at least 42 different (random) timeslots. For each injection, the flow parameters, such as the source/destination IP address or the source/destination port are drawn from the feature distribution defined by the models. Furthermore, depending on the base anomaly model, the feature distributions for some of the flow parameters were modified according to the schemes described below. As a consequence, each injected anomaly is uniquely parameterized. For more details, we refer to the model description files for FLAME which we make available on [11]. In total, we injected 8064 anomalies into our baseline trace. Or more precisely, we injected 42 anomalies in each of the 192 copies of our baseline trace.

Anomaly intensity. Each base anomaly is injected with various intensities, defined by the number of injected flows per 5 min. Chosen intensities are 50 K, 75 K, 100 K, 200 K, 500 K, and 1 M. The motivation for this choice is that the intensities should be (1) realistic and (2) small enough that for most of them the anomaly is invisible when using simple metrics, such as flow count only. We verified these criteria by analyzing the intensities of a set of well-known anomalies and checked that most intensity values are hard to spot when considering the variability and the average number of flows per 5 min bin contained in our traffic traces. We illustrate this with Fig. 2 showing a plot of the number of flows per 5 min bin of our baseline trace into which we injected several anomalies of intensities 75 K and 200 K. While the anomalies of intensity 75 K do not cause a significant change in the flow count signal, those of intensity 200 K start to become visible. However, most of the time they do not stand out clearly but vanish in the normal variability of the flow count signal.

IP addresses. As our traffic traces are collected from a stub AS, we distinguish addresses from the internal address space (IN) and external addresses (OUT). In our anomaly models, the victims are located inside our stub AS with the exception of the reflector DDoS I and Scan III model. We observed that the characteristics of the traffic flowing into the network show a higher variability than those of the traffic leaving our network. Hence, if we place the victims inside our AS, and if the anomalous traffic to the victim(s) is more pronounced than the response traffic, the more pronounced share would be part of the traffic with higher variability and therefore be more difficult to isolate. Previous work, as well as intuition, confirm this imbalance for most anomalies. Most victims of scans do e.g., not reply because the scan is blocked by a firewall, and victims of a DDoS attack do not

Table 1
Overview of 20 base anomaly models used. HAR/LAR means high/low activity region.

ID	Anomaly type	Description	SRC/DST, variation of IPs
	Reflector DDoS I	DDoS with few sources but medium intensity from each source	Attacker: OUT, Victim: OUT, Reflectors: IN
1		Reflector IPs in LAR	
2		Reflector IPs in HAR	
	Reflector DDoS II	DDoS with few sources but medium intensity from each source Matches other similar attacks such as coordinated password-guessing	Attacker: OUT, Victim: IN, Reflectors: OUT
3		Attacker IPs in LAR	
4		Attacker IPs in HAR, Victims in LAR	
5		Attacker IPs in HAR, Victims in HAR	
6	DDoS I	Botnet DDoS 1 (SYN flood)	Victim: IN, Attackers: OUT
7	DDoS II	Flash Crowd/ Botnet DDoS 2 (HTTP GET requests)	Victim: IN, Attacker:s OUT
8	DDoS III	DDoS with spoofed sources (SYN flood)	Victim: IN, Attackers: OUT
9	Worm I	Worm Outbreak (Blaster)	Victims: mainly IN, Attacker: mainly OUT
10	Worm II	Worm Outbreak (Witty)	Victims: mainly IN, Attacker: mainly OUT
11	P2P	P2P Supernode outage (distributed scanning event)	Mix of external/internal addresses
	Scan I	Scanning from single host outside	Victim: IN, Attacker: OUT
12		All ports on single victim	
13		All ports on subnet (hosts in LAR)	
14		Selected ports on subnet (hosts in LAR)	
	Scan II	Scanning from a botnet (2~000 hosts in LAR)	Victim: IN, Attackers: OUT
15		All ports on single victim	
16		All ports on subnet (hosts in LAR)	
17		Selected ports on subnet (hosts in LAR)	
	Scan III	Scanning from single host inside	Victim: OUT, Attacker: IN
18		All ports on single victim	
19		Selected ports on subnet and random IPs	
20	DoS	DoS (1 to 1), HTTP GET requests	Attacker: OUT, Victim: IN

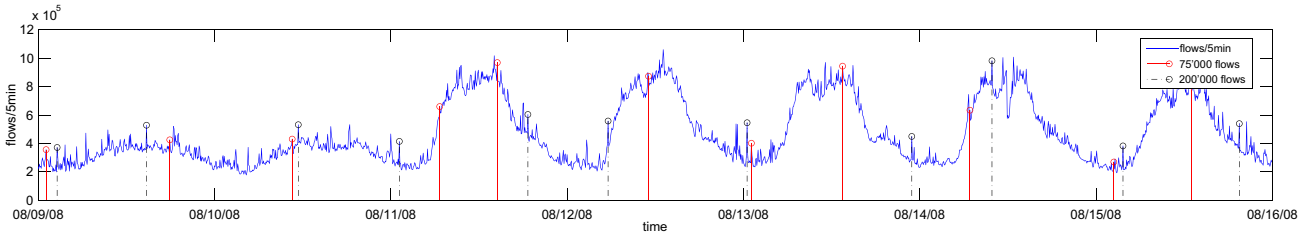


Fig. 2. The number of flows per 5 min bin of our baseline trace with injected anomalies of intensities 75 K and 200 K.

reply to (all) requests because they e.g., crashed or are simply too busy to serve all requests.

Another important aspect influencing the detection results is whether these addresses are already present in the trace or not. We label IP addresses that show persistent and significant activity as belonging to the high activity region (HAR). Those that are rarely present or show very low activity belong to the low activity region (LAR). We draw IP addresses from many combinations of activity regions and set sizes. For more details on how we chose IP addresses and which set was used to customize which anomaly, please refer to Appendix B.

Ports. For application specific attacks and worm outbreaks exploiting vulnerabilities, we selected fixed ports. For instance the HTTP GET requests used in DDoS attacks are targeted at port 80. Otherwise we assign random ports (i.i.d.) from these sets: all ports, ports above/below 1024, selected set of application ports, and a dynamic port range (1024–4999).

Packet sizes. Depending on the attack, we modeled different stages of the 3-way TCP handshake with different response probabilities from $\{0.0001, 0.02, 0.05, 0.2, 0.8\}$. For HTTP requests and Flash crowds, we modeled a percentage of delivered web pages of size 0.5 KB and 20 KB, distributed over several packets. For worm attacks we used characteristic packet sizes known form studies of the Blaster [15] and Witty [16,17] worm. For the reflector DDoS, we measured the actual flow and packet size distributions during a real attack found in our traffic traces and used these distributions for modeling.

3. Entropy telescope

In this section we describe the entropy telescope consisting of Wide Angle Lenses, zoom lenses, Image Processors and a Scene Classifier. Fig. 1 gives an overview of the different components. The Wide Angle Lenses capture the big picture in order to tell the Zoom Lenses the region they should focus on. The Image Processors then take the signals from the zoom lenses and check them for anomalies. If the composed image is considered to be anomalous, the composed image is condensed into a so-called spectrum pattern and fed to the Scene Analyzer for identification.

3.1. Wide Angle: Using Generalized Entropy

An intuitive interpretation of the Tsallis entropy given in (1) is that S_q focuses on changes of elements that show *high activity* for $q \gtrsim 1$, *medium activity* for $-1 \lesssim q \lesssim 1$ and *low activity* for $q \lesssim -1$ [6]. This is because the respective elements contribute the most to the sum (1) *compared to* the elements of other regions. Consider, for example, a high activity element h with $p_h = 0.6$ and a low activity element l with $p_l = 0.1$. If we choose $q = 2$, the contribution is $p_h^2 = 0.36$ for h and $p_l^2 = 0.01$ for l. If, on the other hand, we choose $q = -2$, the contributions are $p_h^{-2} = 2.78$ and $p_l^{-2} = 100$. Whereas the contribution of h was clearly dominant with $q = 2$, the contribution of l is dominant with $q = -2$. In other words, it is possible to focus, for instance, on IP addresses that we see often, occasionally, or rarely in a specific time interval. The main advantages of this filter-like property are (1) that changes affecting parts of the distribution only are more pronounced and (2) that there is more detailed information for the classification of different anomalies.

In [6], we propose to use a traffic entropy spectrum (TES) to characterize changes in traffic feature distributions. In contrast to other entropy based anomaly detection methods [13,1,4,3], the TES does not rely on a single (Shannon)-entropy value but uses a set of Tsallis entropy values that is calculated for subsequent time intervals of size T. The different Tsallis entropy values correspond to S_q with different choices for q, in particular [6] uses $q \in [-2, \ldots, 2]$ in steps of 0.25. However, the correlation between time series resulting from different choices of q has not been analyzed yet.

For anomaly classification, it would be most useful if the different S_q were largely independent from each other and could be used directly to infer the state of a specific region. Then, an increase or decrease of one or multiple S_q's of a specific region would imply a change of the activity pattern in the respective region. For instance, a significant change of the S_q's for $q > 1$ would imply a change in the high activity region. Unfortunately, the way element activities are normalized in (1) makes these types of direct conclusions impossible. When the probability $p(x_i)$ of an element is computed, its activity a_i is divided by the total activity $\sum_{j=1}^{n} a_j$ in a specific time interval. This has the negative side-effect that activity changes in a specific region A are also transported to regions B and C and cannot be distinguished from activity changes originating in B and C. For instance, consider a heavy-hitter being shut down. Because this host caused a lot of activity, the shutdown will reduce the overall normalization factor $\sum_{j=1}^{n} a_j$ and hence also reduce the contributions of rare hosts,[1] although the activity a_i of rare hosts might not have changed at all.

[1] A smaller normalization factor leads to increased probabilities of rare hosts, e.g., all hosts that only occur once. Increased probabilities lead to a decreased contribution in the low activity region, e.g., for $q = -2$.

In the next Section, we modify the TES to alleviate this problem. This change allows to keep the different regions in focus, independently of the overall activity. Our evaluation in Section 4 shows that this modification indeed improves detection and classification results.

3.2. Zooming in: separating activity regions

The entropy telescope mitigates the unwanted normalization effects by computing a *pruned* entropy in a two-step approach. For each time interval, we start with calculating the TES consisting of the entropy values S_q for a set of q-values. We then zoom in on the most contributing elements responsible for p percent of the value of S_q, for a given q. In the second step, we calculate the pruned entropy for the selected elements only, denoted by $S_{q,p}$. With this procedure, we make sure that changes of elements i that contribute almost nothing to the sum $\sum_{i=1}^{n} p(x_i)^q$ have no impact on the final $S_{q,p}$, neither through direct contribution nor through normalization.

More formally, let the original distribution of activities be $A = \{a_i, \ldots, a_n\}$. Then we first compute $S_q(A)$ as defined by (1). Now let $C = c_i$ be the set of element contributions, that is $c_i = \left(a_i / \sum_{j=1}^{n} a_j\right)^q$. Then we let C' be the sorted version of C such that $c'_j \geqslant c'_{j+1}$ and store the mapping of indices between C and C' in a table ϕ. Thus, if c_k is mapped to element c'_l, we have $\phi(k) = l$. Let $\sigma(x)$ be the partial entropy computed by summing up all contributions of C' up to element x, that is $\sigma(x) := \sum_{j=1}^{x} c'_j$. Further, let $\hat{x}$ be the smallest index x for which $\sigma(x) \geqslant p/100 \cdot S_q(A)$ holds. From this we construct the set of selected activities $A' = \cup_{j=1}^{\hat{x}} a_{\phi^{-1}(j)}$. Finally, the pruned entropy is computed by $S_{q,p} := S_q(A')$.

The output of the zoom lenses, the pruned TES_p, is now simply the current and past values of $S_{q,p}$ for the given set of q-values. It can therefore be plotted in the same way as the original TES. Note that the original TES corresponds to TES_{100}.

Fig. 3 illustrates the effect of TES_p. The top figure shows a destination port activity distribution with the ports on the x-axis ordered by ascending activity. That is, the leftmost port with index 1 is the rarest and the rightmost port is the top port (port 80 in this case). The activity of a port is plotted on the y-axis (i) during an anomaly and during normal activity.

For both distributions there is one plot below, showing the selected elements for different values of q and p. At a

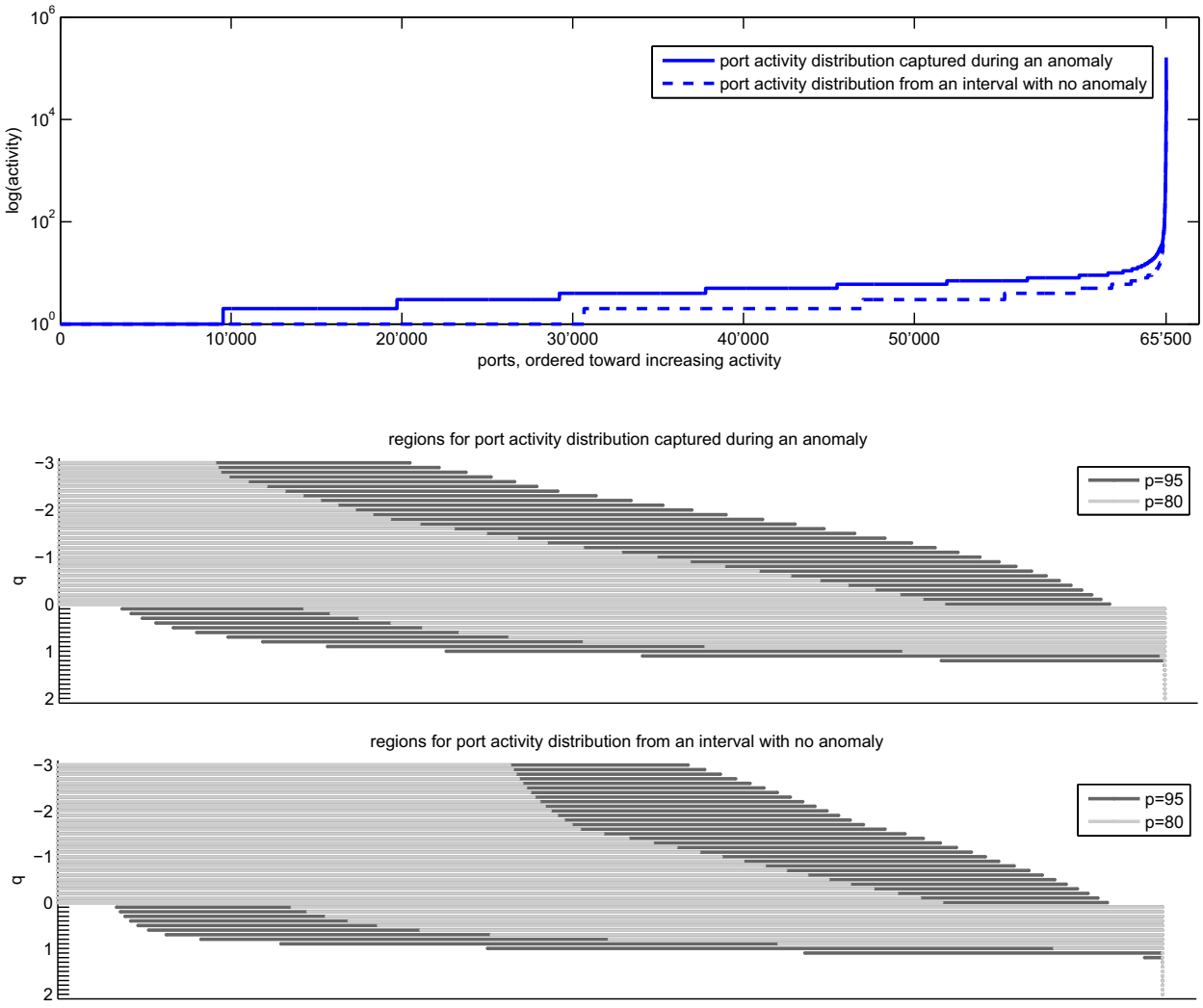


Fig. 3. Destination port activity distributions (top) and selected regions for TES_p (bottom). On the x-axis all ports are ordered by rank, i.e., with increasing activity to the right.

specific coordinate (x,q) there is a point if element x was selected for the pruned entropy $S_{q,p}$ and no point otherwise. For instance, looking at the regions for the anomalous port distribution, we see that for $q = -3$ and $p = 80$, only about 10,000 ports on the left (i.e., the low activity ports) are selected. Looking at the regions for the normal port distribution, we see that $q = -3$ kept the low activity region in focus even though there are now around 28'000 low activity ports. Similar observations can be made for other q- and p-values with smaller p values tending to capture the different activity levels more tightly at the cost of being probably too tight: $q = -3$, $p = 80$ does e.g., not select the full range of low activity ports in the normal port activity distribution.

3.3. Image processing: anomaly detection

In this section we describe how anomaly detection is performed on the various entropy signals for different metrics and q-values. Specifically, We use 20 different values for q:

$$q \in \{-3\} \cup \{-2, -1.75, \ldots, 1.75, 2\}.$$

Including bigger/smaller values is of limited use because S_2 is already very much dominated by the biggest heavy-hitter and S_{-3} by the rarest elements, respectively. With 8 feature entropies, 3 volume metrics (flow, packet, and byte count), and two directions, this yields a total number of $2 * (3 + 8 * 20) = 326$ different metrics for TES_p. For Shannon classic (SHN_C) we use $2 * (3 + 4) = 14$ metrics, and $2 * (3 + 8) = 22$ metrics for Shannon extended (SHN_+), respectively. The computational overhead is dominated by generating element distributions, in the first place. Whether we compute a single entropy value or draw multiple values from a distribution does not make a big difference in terms of running time or memory consumption.

From the list of available statistical anomaly detection methods, including wavelet transformation [18], Kalman filter [7], Principal component analysis (PCA) [2], and Karhunen–Loeve Expansion (KLE)[3], we selected the Kalman filter due to its simplicity as well as the PCA and the KLE method because they reflect the current state of the art:

- **The Kalman filter** models normal traffic as a measurement-corrected AR (1) auto-regressive process plus zero-mean Gaussian noise. The difference between this model and the actually measured value is the residual, a zero-mean signal without the diurnal patterns found in original time series. We calculate this residual for all input time series separately.
- **The principal component analysis (PCA)** condenses the information of all input time series to a single output time series reflecting how closely the current input matches the model built from some other input. PCA has a parameter k determining how many of the components are used for modeling the normal activity. We discuss the impact of k in our evaluation section.
- **The Karhunen–Loeve expansion (KLE)** is based on the Karhunen–Loeve Transform and basically an extension of the PCA method to account for temporal correlation in the data. The only but important difference is that KLE has an additional parameter m stating how many time bins should be included when accounting for temporal correlations.

We point out that our goal is not the optimization of the detection step, but rather to demonstrate that the extended set of Tsallis entropy values improves the detection accuracy using existing methods.

On the residual(s) we detect anomalies using a quantile-based approach: The first quartile Q_1 of a sample of values corresponds to the 25th percentile and is defined as the value that cuts off the lowest 25% of values. That is, one fourth of the values is smaller than Q_1. Similarly, Q_2 (the median) and Q_3 are defined as the 50th and 75th percentile, respectively. The interquartile range IQR is a measure of statistical dispersion and is defined by $IQR = Q_3 - Q_1$. The IQR can be used to detect outliers by defining a normal range of values $[Q_1 - k \cdot IQR, Q_3 + k \cdot IQR]$ for some constant k. We choose $k = 1$ and define the normalized anomaly score $A(x)$ for a residual value x by the ratio of the distance of x from the normal band and the size of the normal band, which is $3IQR$:

$$A(x) := \begin{cases} \frac{x-(Q_3+IQR)}{3IQR} & \text{if } x > Q_3 + IQR, \\ \frac{x-(Q_1-IQR)}{3IQR} & \text{if } x < Q_1 - IQR, \\ 0 & \text{else(signal is normal)}. \end{cases} \quad (3)$$

For each output time series, we compute the anomaly score and call it a *vote* if the signal is exceeding a threshold t, that is, $|A(x)| > t$.

In the case of PCA and KLE, we have only one output time series. As a consequence, one vote is enough to trigger an anomaly and the threshold t is the main parameter to tune the sensitivity of a specific detector.[2] However, in the case of the Kalman filter, we have one residual per input time series and detection is done using a two parameter approach. First, we do the same as in the case of PCA and KLE for each of the output time series: we put a threshold t on all of the anomaly scores $A(X)$ of their residuals. Next, we perform the detection by setting a minimum number v of votes required to trigger an *alarm* for the current time interval. In practice, determining good values for the threshold t and votes v is done by measuring the performance of the detector for different combination of t and v. Ideally, this is done using training data containing a representative set of anomalies. The same holds for determining k for PCA and k and m for KLE or any other anomaly detection system having one or more tuning parameters. In summary, we need to sweep the following tuning parameters to fully assess the performance of the different algorithms:

- **Kalman**: Threshold t and number of minimum votes v.
- **PCA**: Threshold t and the number k of components used for modeling the normal activity.
- **KLE**: Threshold t, the number k of components and the number m of time bins used for modeling the normal activity.

[2] Note that there are other tuning parameters such as the parameters k for PCA and k and m for KLE as described before.

Note that all of the three approaches require training data for two reasons: (1) for defining a conservative normal band to derive the normalized anomaly score $A(X)$ and (2) to get training data for training the models used by the Kalman, PCA and KLE methods. While the first training problem is easy to solve, the second one is more difficult. The reason for this is that our IQR based normalization is based on the first and third quartiles, which do not depend on the 25% smallest and biggest values in the data. It is therefore not affected by outliers. Unfortunately, to solve the second training problem, we need all of the data points. To ensure that the training data reflects indeed normal behavior, we selected the training data based on manual analysis of the time series using box plots and raw time series plots. While there remains an uncertainty whether our selection of training data is really clean and representative, we mitigated this by confirming our findings using different training samples. However, we can not omit this problem entirely when working with real traces containing millions of flows per hour.

In our evaluation, we focus on those configurations showing the 'best' performance for a specific method. We are aware of the fact that different sets of anomalies and/or other background traffic characteristics might result in a different choice for these values or worse, a different rating for the different methods. However, we believe that our comparison is fair for two reasons: (1) the selection of the 'best' parameters is based on a large set of different anomaly types and intensities and without potential bias because of anomalies that are more frequent than others, as typically the case with any real world traces. And (2), our traffic trace used as background traffic originates from a large stub AS with fairly complex and dynamic traffic mix characteristics.

3.4. Scene analysis: classifying anomaly patterns

The basic idea behind the scene analysis component is the notion of *spectrum patterns* introduced in [6]. The assumption underlying our anomaly classification is that each anomaly class leaves a characteristic and (to some degree) invariant footprint in different features and activity regions. As a consequence, the input to this component must be one signal per count or feature entropy. While the input signals could be the original time series signals of these features, we want to avoid this for two reasons: First, removing trend and daily patterns form the signals is difficult but has to be done for most supervised pattern recognition approaches. And second, we are not interested in the exact amplitude of the signals but rather a conservative estimate whether they are abnormal and if yes, how much.

An obvious choice for the input of the classification component is therefore the output of the Kalman detector: It outputs a conservative anomaly score per input time series. To reduce the volume of data provided by this detection component, we aggregate anomaly scores in three buckets corresponding to the low/medium/high activity regions by calculating the weighted sum of the scores for all q-values in a region. The low activity region is defined by $q \leqslant -1$, medium by $-1 < q < 1$, and high by $q \geqslant 1$. That is, we calculate three values for each metric, measuring the abnormality of the specific region, denoted by A_l (low), A_m (medium), and A_h (high). While different weights might be used to tune our classification approach in future work, we found that the simplest choice of setting all weights to one is enough for achieving a classification accuracy of around 85 percent.

In a next step, the Scene Analyzer scans the values A_l, A_m, and A_h of each traffic feature and decides whether they signal an increase, decrease or no change of entropy of the corresponding regions. This transformation can be summarized as follows:

$$C_i := \begin{cases} \text{'1'} & \text{if } A_i \geqslant \text{ upper threshold}, \\ \text{'0'} & \text{otherwise}, \\ \text{'}-1\text{'} & \text{if } A_i \leqslant \text{ lower threshold}. \end{cases}$$

For the upper and lower threshold, we use the values 0.5 and -0.5 respectively. A value of $A_i = 0.5$ is e.g. obtained, if each metric contributing to A_i exceeds its 75th percentile value by around $1.2 * IQR$.[3] Another situation resulting in $A_i = 0.5$ is when one of the metrics contributing to A_i has an anomaly score of 0.5 and all others an anomaly score of zero. From (3) it follows, that for an anomaly score of 0.5, the metric exceeds the 75th percentile value by $2.5 * IQR$. Note that a deviation of $1.5 * IQR$ is typically attributed to *mild outliers* while a deviation of at least $3 * IQR$ is attributed to *extreme outliers*.

The main reason for transforming the continuous values A_l, A_m, and A_h of each traffic feature into discrete (tri-state) values is to avoid the pitfall of overfitting our classifier to specific amplitudes. Despite the good results produced by this approach, we need to investigate the impact of this quantization in more detail. However, not using quantization should mainly improve the classification quality in cases where the input signals are not well-behaved in the sense that the IQR is not meaningful for separating normal and abnormal values. An example of such a signal is, e.g., a signal that has a more or less bi-modal distribution of its values during normal activity.

In a last step, the Scene Analyzer feeds the discretized spectrum pattern to a support vector machine (SVM) trained with different training sets discussed in the evaluation section. Our Scene Analyzer makes use of the LIBSVM [19], a popular SVM with very good performance and a wide range of available interfaces. For each of the different training sets, we followed the basic strategy outlined in [20]: First, we split the full dataset into three parts containing approximately the same amount of anomalies of each anomaly type and size. Next, we take two parts of the split for training and one part for validation. By doing this, we get three different training- and validation set combinations. On the training set, we then perform a grid search and 3-fold cross-validation to identify the best parameters for the SVM's RBF kernel. The classification result reported in the evaluation section is the average classification accuracy obtained from the three training- and validation set combinations. Note that the output of the SVM – the label

[3] With 5 metrics as in the high activity region, we get $A_h = 0.5$ if all metrics have an anomaly score of 0.1. It follows from (3) that an anomaly score of 0.1 is the same as exceeding the 75th percentile by $1.2 * IQR$.

of the anomaly – is at the same time the final result and output of our Entropy telescope.

4. Evaluation

In this Section we evaluate the entropy telescope using the feature sets SHN_C, SHN_+, TES_{100}, $TES_{99.9}$, TES_{99}, TES_{95}, and TES_{80}. We show that the biggest improvement in detection accuracy can be achieved when switching from Shannon entropy based feature sets to the TES_{100} set. The novel TES_p makes another significant step in classification accuracy and optimizes detection for some anomaly categories.

After thoroughly discussing detection and classification results, we conclude the section with an analysis of anomaly prevalence in a 34-days trace of real traffic.

4.1. Detection

In Section 3.3 we defined a metric to be anomalous, denoted by a *vote*, if its anomaly score is bigger than a threshold t, i.e., $|A(x)| > t$. Moreover, for an anomaly alarm to be raised in a time slot, a number of v votes need to be present. For the PCA and KLE method, v is equal to one since they have only one output time series. Naturally, high thresholds for t - and in the case of the Kalman filter also for v – will lead to low true/false positives while low thresholds lead to high true/false positives. The preferred operation point, however, has a high true positive (TP) and a low false positive (FP) rate. To assess detector performance, we use Receiver Operating Characteristics (ROC) curves [7] that plot the TP rate versus the FP rate for a range of threshold values. In our case, we vary t between 0 and 100. Note that for readability reasons, we plot the ROC curves using a logarithmic scale for the FP axis and display the results for FP rates of 0.4% to 10%. With our time bins of 5 min, this corresponds to roughly 1 false positive per day for an FP rate of 0.04% to 1 false positive per 50 min for (10%).

Issues with KLE. The following discussion focuses on the evaluation results for the Kalman and the PCA methods only. The reason for this is that our results for KLE are somewhat ambivalent. For intensities larger than 100 K, KLE shows a worse performance than PCA for all feature sets. The same holds for the feature sets TES_{100} or TES_p and anomalies of intensity up to 100 K. However, for SHN_C and SHN_+ and anomalies of intensity up to 100 K, we see an improvement in detection quality of up to 15%. While the improvement for SHN_C is consistent with the finding in [3], we are not quite sure about the root cause for the results with other feature sets. More research is required to better understand the performance of the KLE method with different feature sets, anomalies and network characteristics.

Shannon versus TES feature sets. Fig. 4 shows the ROC plots for the Kalman and PCA method for intensities 50 K and 75 K as well as the PCA method with 100 K and 200 K. The plots show the detection accuracy for the best configuration of different detectors for the feature sets. To find the best configurations, we performed an extensive parameter sweep for both, the Kalman and PCA detector. For PCA, the parameter is the number of components k used to build the model of normal activity. For Kalman, the parameter is the number of votes v required to trigger an alert. Doing these sweeps, we found that while the detection accuracy is changing quickly for the feature sets SHN_C and SHN_+, there is a clear peak for one specific value of k. In contrast, this is not true for TES_{100} or TES_p. After reaching the optimal detection accuracy, it remained at a comparable level for a wide range of k values. One interpretation of this is that the additional time series in the *TES* feature sets make the detectors more robust with regard to the selection of the parameter k.

From the plots in Fig. 4 we can see that a switch to TES, improves the detection accuracy for PCA by up to 20%. However, for the Kalman filter approach, the gain is rather small and lies around 5% for TES feature sets other than TES_{100} or TES_{80}. It seems that while the TES adds features carrying valuable information, it also adds noise with which the simple per-feature detection and voting scheme of the Kalman detector does not cope well. Unlike PCA, our Kalman detector does not make use of inter-feature relations. This being the main reason for the worse performance is supported by

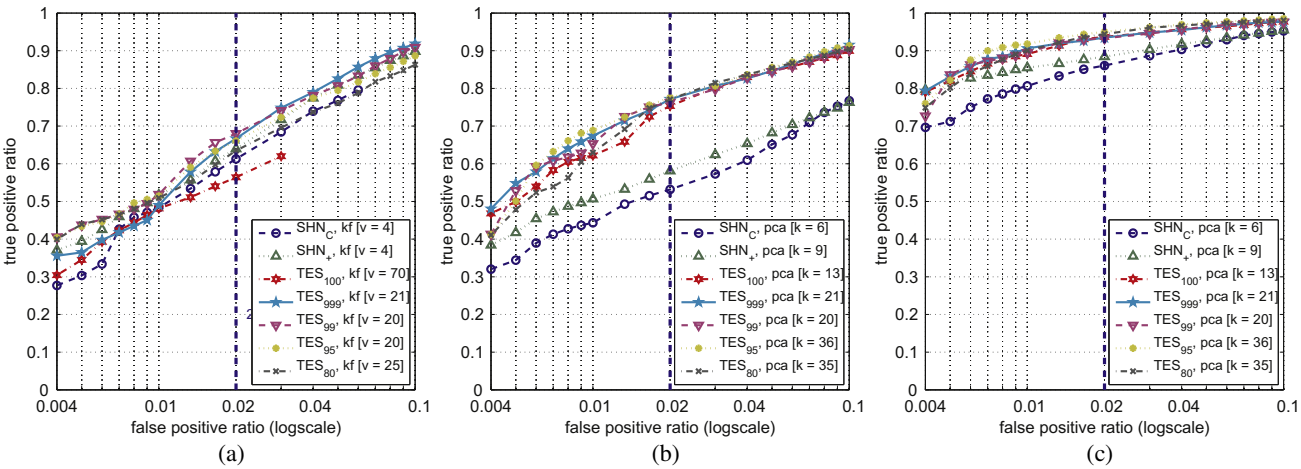


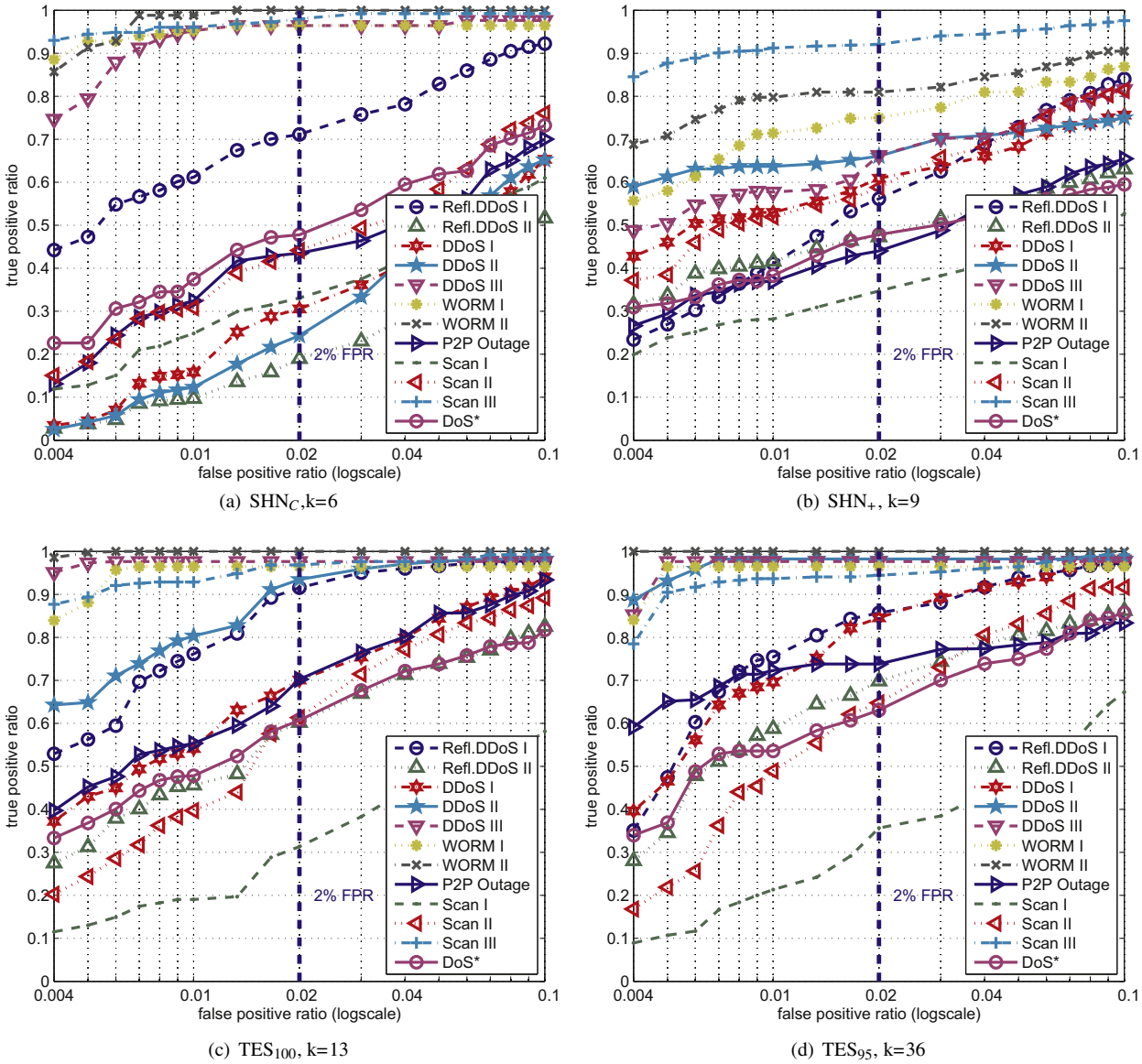
Fig. 4. ROC curves for different feature sets and detection methods: (a) Anomalies of intensity 50 K and 75 K, Kalman filter (kf) method. (b) Anomalies of intensity 50 K and 75 K, PCA method (c) Anomalies of intensity 100 K and 200 K, PCA method.

the Kalman filter's very bad performance for TES_{100} but significantly improved performance for TES_p. As explained in Section 3, the features reflecting the high and low activity area can be heavily correlated in TES_{100}, but are not correlated in TES_p. As a comparison of the different plots in Fig. 5 shows, the improvement in detection accuracy can also be confirmed when looking at the detection accuracy per anomaly type. Switching from SHN_C or SHN_+ to TES_{100} improves detection accuracy for most types for FP rates of 0.6% (=1 alert per 14 h) and above.

SHN_C. versus SHN_+ Another observation we can make based on Fig. 4 is that our extension of the traditional feature set SHN_C to SHN_+ improves detection results by up to 10%. This, as well as the increase from $k = 6$ to $k = 9$ components required to achieve the best detection accuracy with PCA, confirms that the features added to SHN_C carry relevant information. Nevertheless, as can be seen in Fig. 5, the better overall detection accuracy comes with a decrease for the anomaly types Worms I& II, DDoS III and Scan III while most of the other types show an increase in detection accuracy.

Kalman versus PCA. But the most surprising result is exposed when comparing the performance of the different detection methods for the feature set SHN_C and SHN_+ in Fig. 4(a) and (b): The Kalman filter method detects anomalies up to 10% more accurately than the PCA method. Considering that PCA has been used with the feature set SHN_C in the past, this is an interesting finding. But since this result only holds for anomalies with intensities less than 100 K, PCA might still be the best choice for SHN_C in general. The effect disappears when the feature set is extended to TES_p. There, we found that the PCA method provided consistently better results than the Kalman filter method.

Relations between parameters k. A final observation from Fig. 4 is that the optimal k value for both, Kalman and PCA increases when switching from Shannon to TES. The increase is even of comparable size. Except for TES_{100} for an afore mentioned reason: The Kalman method does not



(a) SHN_C,k=6 (b) SHN_+, k=9

(c) TES_{100}, k=13 (d) TES_{95}, k=36

Fig. 5. ROC curves for anomalies with small intensities (50 K and 75 K) and PCA detection method.

make use of inter-feature relationships, such as the correlations between high and low activity regions in TES_{100}.

In summary, the shift from Shannon-based feature sets to *TES*-based sets can improve detection accuracy up to 20%. The reason why a shift from TES_{100} to a refined version of the TES only leads to minor improvements might be the fact that the main difference between TES_{100} and TES_p is the decorrelation of the HIGH/LOW intensity parts of the distribution. Intuitively, for the detection, we do not care whether an anomaly is seen in two (correlated) metrics or just one (uncorrelated) metric. At least for PCA and KLE, which account for correlations between metrics, this makes no big difference. We believe that the minor gains are most likely due to a better signal to noise ratio for anomalies affecting the low activity region only. In TES_{100}, such anomalies could be concealed by large (but not yet anomalous) changes in the overall activity.

4.2. Classification

It is important for detection and classification to rely on models that are robust with respect to varying intensities. That is, if we train an SVM with DDoS models of a certain intensity, we do not want to miss the same attacks only because the real attack size differs slightly from the training size. Therefore, we trained the SVM with different intensities and evaluated the models on varying intensities. We always trained all of the 20 base models from Table 1. For measuring classification accuracy, we counted the percentage of anomaly instances that were assigned to the correct base model. Thus, if anomaly #16 was classified as anomaly #17, this is considered incorrect, even though both belong to the same base anomaly type (Scan II). *For assessing classification quality we assumed a perfect detector.* That is, the true anomalous intervals are considered for classification. In a real environment, classification would only be run on those instances that were detected by an anomaly detector in the first place. The consequence of this is that the difference between classification accuracy of *SHN* and *TES* feature sets would be even bigger in practice because a detector based on the *SHN* feature set would feed more false positives to the classifier.

Table 2 summarizes the classification accuracies for different anomaly intensities and feature sets. The columns labeled with arrows (⇒) show the performance difference between the feature sets on the left and right side. The use of SHN_+ over SHN_C yields a gain in classification accuracy between 7.14% and 14.21% across all intensities. Using TES_{100} gives and additional gain of 7.84% to 9.38% for small intensities in the top three rows. For training and classification with bigger anomalies, the gain is generally smaller. Although accuracy with TES_{100} is already quite high, the introduction of the pruned TES_{95} adds another 5.8% on average. While choices of $p = 99.9$ and $p = 80$ also improve over TES_{100}, $p = 95$ works best in our setting. The average aggregated gain of TES_{95} over SHN_C is 22.3%, leading to an average classification accuracy of 83.17%. The improvement is generally bigger for low-intensity anomalies. In Fig. 6 we provide a detailed view on which base anomalies were classified correctly and which not. Each point in the plots indicates the probability that the anomaly on the *y*-axis was classified as the anomaly on the x-axis. SHN_C and SHN_+ often misclassified anomalies of types 3–5 and 13–18. As expected, the classification accuracy with regard to sub-types of the broader anomaly types increases when switching from *SHN* to *TES* feature sets. This is expected since *TES* provides a more detailed view on the changes in a distribution. For a broad classification, these details are clearly less important.

To give a graphical intuition of cluster centers and boundaries for different anomaly types, we show Fisher's LDA (Linear discriminant analysis) [21] in Fig. 7. LDA is typically used in machine learning to find a linear combination of features which characterize or separate two or more classes of objects. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. The plots show that for intensity 50 K, Shannon yields no clear clusters, whereas TES_{95} is capable of separating "Ref. DDoS 1" from "DDoS + Worm" and Scans. With intensity 200 K, the situation improves for both sets of metrics, but clusters are still better distinguished for TES_{95}.

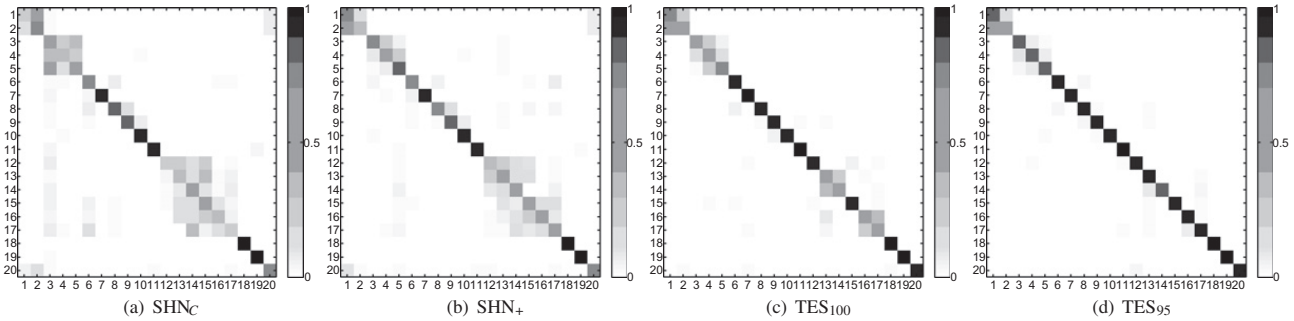
4.3. Prevalence of anomalies in real backbone traffic

As a last step in our evaluation, we report and discuss the results from applying our entropy telescope to a 34 days flow trace collected by one of the border routers of the SWITCH network in August 2008.

Fig. 8 shows four pie charts representing the detected anomalies for different detection thresholds. From subfigure (a) to (d), the detection threshold is lowered successively, resulting in alert rates of 0.5% for (a), 1% for (b), 3% for (c), and 10% for (d). An alert rate of 0.5% means that

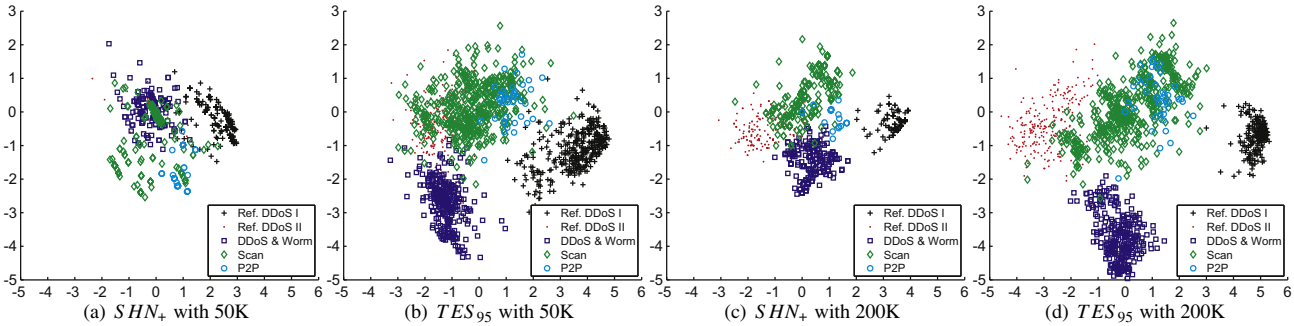
Table 2
Average classification accuracy in percent for different sets of features and for different training and validation data set constraints.

Training	Evaluation	SHN_C	⇒	SHN_+	⇒	TES_{100}	⇒	$TES_{99.9}$	TES_{95}	TES_{80}
50 K	50 K	55.13	10.42	65.55	9.38	74.93	7.14	80.58	82.07	80.95
	>50 K	54.73	8.78	63.51	7.84	71.35	7.16	74.26	78.51	77.35
200 K	<200 K	49.38	8.04	57.42	9.13	66.54	8.43	72.82	74.98	73.83
	200 K	66.07	14.06	80.13	2.16	82.29	5.21	86.53	87.50	87.72
	>200 K	64.69	14.21	78.91	1.49	80.39	4.09	80.95	84.49	84.34
ALL	<200 k	60.91	7.14	68.06	8.85	76.91	7.04	80.95	83.95	86.46
	200 K	68.30	11.68	79.99	3.65	83.63	4.17	85.27	87.80	87.80
	>200 K	67.49	13.36	80.84	1.75	82.59	3.50	83.15	86.09	82.96



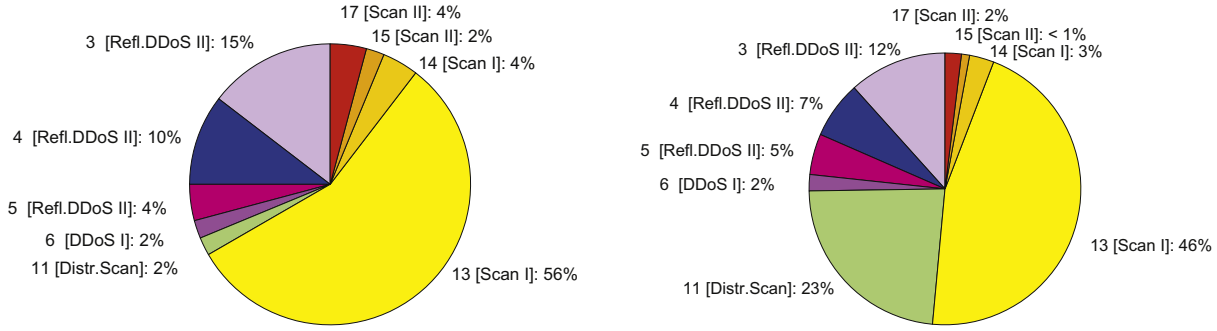
(a) SHN_C (b) SHN_+ (c) TES_{100} (d) TES_{95}

Fig. 6. Base anomaly classification matrix. The plots show which injected base anomaly types (y-axis) were classified as which types (x-axis) with what probability (color code). Models were trained using anomalies of ALL intensities. Classification is performed on anomalies with intensity < 200 K.



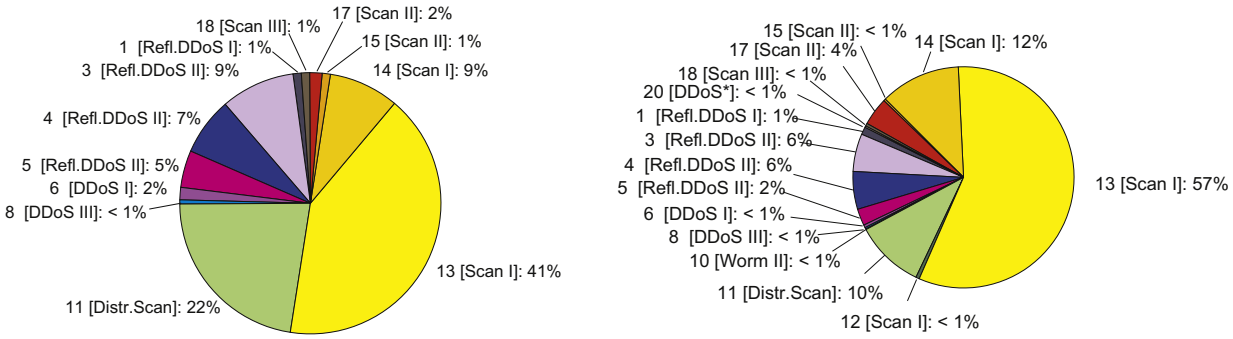
(a) SHN_+ with 50K (b) TES_{95} with 50K (c) SHN_+ with 200K (d) TES_{95} with 200K

Fig. 7. Fisher's LDA plots of SHN_+ versus TES_{95}.



(a) Alert rate: 0.5% (=1 alert every 16.7 hours)
Total alerts: 49 (out of 9792 time bins)

(b) Alert rate: 1% (=1 alert every 8.3 hours)
Total alerts: 103 (out of 9792 time bins)



(c) Alert rate: 3% (=1 alert every 2.8 hours)
Total alerts: 196 (out of 9792 time bins)

(d) Alert rate: 10% (=1 alert every 50 minutes)
Total alerts: 979 (out of 9792 time bins)

Fig. 8. Detection and Classification results for a 34 days flow trace collected by one of the border routers of the SWITCH network in August 2008. Results are for TES_{95} with a PCA [$k = 36$] detector.

1 in 200 timeslots with duration of 5 min is considered anomalous, i.e., one anomaly is reported every 16.7 h. A high alert rate of 10% as in subfigure (d) results in one alert every 50 min and is certainly not desirable for daily operations. It is only shown to give an idea of the behavior of the classifier for very low thresholds. This is interesting since we expect a larger number of false positives for this setting and were interested to see whether this leads to classifications of anomalies as events that are presumably not present in our trace: worm outbreaks.

For all thresholds, scans are predominant, accounting for roughly 2/3 to 3/4 of all anomalies. This result is consistent with the fact that scanning has become omnipresent in today's networks [22] and is often not even considered to be of special interest anymore. Among scans, type 13 (scan of a subnet from a single host) has by far the biggest share. Type 11 (distributed scanning) goes up from 2% to 23% when going from (a) to (b). The relatively high threshold in (a) was most likely not sensitive enough to detect the distributed *n*-to-*m* scanning modeled with type 11. Therefore, it is only reported with lower thresholds as in (b) to (d). Regarding worm activity, no alerts were triggered and also the network operator is not aware of any incidents. There is only one worm alert in subfigure (d), which we consider to be a false-positive.

DDoS-type anomalies have a share between 23% and 31% for (a) to (c). Translated into number of incidents, this means between 15 and 45 DDoS events for the measured period of one month. Note that these events may also contain flash crowd events, as these are generally very hard to distinguish from DDoS attacks. Or in the case of the type Refl.DDoS II, massive coordinated password guessing attacks. It is difficult to compare these figures to external numbers, primarily due to the difficulty of quantifying global DDoS activity. Furthermore, it is not clear how global numbers are broken down to an individual network for comparison. Moore et al. estimate 2,000–3,000 global DDoS attacks per week already for 2001–2004 [23]. VeriSign, drawing from different sources, estimates between 1000 and 10,000 DDoS attacks per day in 2008 [24]. The CSI computer crime and security survey 2008 [25] states that from the 522 responders, 21% were affected by DoS attacks in 2008. Of course, the reported incidents are only those that had enough impact to be recognized by operations.

Considering that our traces contain traffic from around 40 individual organizations, we think our numbers are realistic. That is, for a medium alert rate, we expect around 1 DDoS alert per day.

5. Related work

Most approaches for anomaly detection in large scale networks rely (to some extent) on traffic-feature distributions. In [26,27], the distributions are captured by histograms while [28,4,29,30] summarizes them with Sketch data structures. Sketch-based approaches rely on a set of histograms where the elements are assigned to the bins using a set of different hash-functions. Approaches that rely on entropy to expose changes in distributions using (1) Shannon-Entropy [2,1,4], (2) an approximation of (Shannon-) entropy [13] based on compression or (3) the Kullback–Leibler Distance which corresponds to the Kullback–Leibler entropy,[4] [26,32]. A different application of entropy is presented in [33] where the authors introduce an approach to detect anomalies based on Maximum Entropy estimation and relative entropy. The distribution of benign traffic is estimated with respect to a set of packet classes and is used as the baseline for detecting anomalies. In [5], Ziviani et al. propose to use Tsallis entropy for the detection of network anomalies. By injecting DoS attacks into several traffic traces they search for the optimal q-value for detecting the injected attacks. While Ziviani et al. found a q value around 0.9 is best for detecting DoS attacks, Shafiq et al. [34] could optimize the detection of portscans of malware using a q value equal to 0.5.

Tsallis entropy has also many applications in physics, medicine or in a broader context, in complex systems. In [35], the authors propose a q-parameterized Expectation Maximization (q-EM) algorithm for parameter estimation based on incomplete observations. They investigate iterative schemes for joint channel estimation and signal detection over frequency selective channels using q-EM algorithms and show that convergence speed is improved by replacing the standard expectation with q-expectation, which was first introduced in the Tsallis entropy literature. And in [36], Torres et al. exploit the ability of multi-resolution entropies to show slight changes in a parameter of the law that governs the nonlinear dynamics of a given time series signal. To do so, they capture these changes as statistical variations at each scale and calculate the corresponding principal components and feed them to a statistical change detector. There are many more applications of Tsallis entropy which are loosely related to anomaly detection such as e.g., [37]. Refer to [38] for a complete bibliography on Tsallis entropy related publications.

6. Conclusion

In this paper, we improved network anomaly classification by introducing the pruned TES (traffic entropy spectrum) feature set, which uses the non-extensive Tsallis entropy to focus on specific regions of feature distributions. We built an integrated anomaly detection and classification system called the *entropy telescope* and compared the performance of different well-known detectors, such as the Kalman filter, PCA, and KLE. We extensively evaluated the entropy telescope with a rich set of artificial anomalies and real backbone traffic. We show that using the pruned TES instead of classical Shannon-only approaches improves detection accuracy by up to 20% and classification accuracy by 22.3% on average. In particular, the pruned TES is much more sensitive for small anomalies and established anomaly patterns are very robust with respect to varying anomaly intensities. A run of the entropy telescope on one month of backbone traffic shows that most prevalent anomalies are different types of scanning (69–84%) and reflector DDoS attacks (15–29%).

[4] Rényi distance of order 1 ($\alpha = 1$) [31].

Appendix A. Analysis of feature correlation

The Shannon entropy of a random variable X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i), \tag{A.1}$$

$$p(x_i) = \frac{a_i}{\sum_{j=1}^{n} a_j}, \tag{A.2}$$

where a_i is the number of occurrences of x_i in a time window of length T and $p(x_i) = p(X = x_i)$. In our context, the x_i are the feature elements, e.g., specific IP addresses or port numbers.

Nychis et al. [8] raised a concern regarding the pairwise correlation of different feature entropies. They found that port entropy, address entropy and traffic volume (packets/s) are highly correlated. Therefore, a single feature, e.g., traffic volume, would already provide enough information for reliably detecting DDoS-like events. Consequently, the use of multiple features would not provide additional information to improve the anomaly detection rate.

Motivated by our own experience in the field, which contradicts the results reported by Nychis et al., we performed our own correlation analysis of traffic features. *We did not find any persistent strong correlation between traffic features.* To aid detection and especially classification of network anomalies, we therefore suggest to use a wide range of features to capture different aspects of traffic dynamics.

A.1. Methodology

In this Section we present the methodology of our correlation analysis. We performed a correlation analysis on the following entropies:

- Flow size (Fsize).
- Bytes per packet (BytesPP).
- Source and destination port (Dp, Sp).
- Source and destination IP address (Sip, Dip).
- Autonomous system (AS).
- Country code (Country).

We computed the entropy values for the various distributions over time and compared the resulting timeseries of entropy values using correlation metrics defined below.

Correlation metrics. A possible correlation metric for two timeseries X and Y consisting of n data points is the Pearson product-moment correlation r, as used by [8]. The Pearson correlation coefficient r_{xy} is defined as

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sigma_x\sigma_y}. \tag{A.3}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of X and Y, and σ_x and σ_y are the sample standard deviations of X and Y. In particular, Nychis et al. measured Pearson correlation scores bigger than 0.95 for port and address distributions, where score *1* means maximum correlation.

An alternative correlation metric is the Spearman's rank correlation defined as

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \tag{A.4}$$

where $d_i = x_i - y_i$ is the difference between the ranks of corresponding values X_i and Y_i.

Whereas Pearson only captures linear correlation, Spearman considers any correlation described by a monotone function, including linear correlation. A comparison of the two correlation metrics on our data set showed that Spearman correlation was consistently higher than Pearson correlation, hinting at considerable non-linear correlation. Therefore, we used Spearman's correlation for our analysis.

Data Set. To evaluate the feature correlation, we used 10 different traces summarized in Table A.4 from SWITCH, the Swiss educational and research network [12]. SWITCH connects several universities, research labs, and governmental institutions to the Internet. The network is a stub AS with an IP address range containing about 2.4 million addresses, which we refer to as *internal* address space. External addresses are all addresses not assigned to the network's range. Accordingly, we use the term *incoming* traffic to denote flows from external source to internal destinations and *outgoing* traffic for the reverse direction.

The flows are collected from four different border routers which do not apply sampling or anonymization. Note that sampling and anonymization can skew certain parts of feature distributions. For instance, deletion of least significant 11 IP address bits, as applied to Abilene traces [2], corresponds to an aggregation of IP addresses at the /21 subnet level and reduces the utility of entropy metrics for anomaly detection [39].

Traces 1–9 were captured on the largest exchange point (router 1) around major anomalies, such as global worm outbreaks, outages or a DDoS attack using internal hosts as reflectors. On average, roughly 50% of their duration is considered anomalous. Trace number 10 is a continuous trace over 4 months from all exchange points with no major anomaly. In total, the traces cover 247 days from 5 years.

A.2. Feature correlation

The absolute value of the Spearman coefficients in percent are presented in the Tables A.3, A.5, and A.6. A value of 100 denotes maximum correlation where on the other hand 0 means no correlation.

Table A.3 shows correlation statistics for traces 1–9, comprising several *anomalous* intervals from a range of 5 years. Strong correlations ($\geqslant 0.8$) are highlighted. For each feature pair, we compute the correlation of the respective time series for each of the nine traces. Then, the maximum, minimum, and average correlation is selected for each feature pair. Generally, correlation of the different features is low. For some feature pairs, correlation is high in some traces, but low in general. This is, for instance, the case for (Sip, Dip) with a maximum correlation of 0.9 but an average correlation of only 0.4. Only the pair (BytesPP, Fsize) has a very strong average correlation of almost 1. The next highly correlated feature pairs are (Sip, Fsize) with 0.83 and (Sip, BytePP) with 0.81 average

Table A.3
Correlation of different feature entropies for traces 1–9 (see Table A.4) in percent. The table shows maximum, minimum, average, and standard deviation for correlation of $H(X)$ for TCP traffic.

	Sp				Dp				AS				Sip				Dip				Country				BytesPP				Fsize			
	max	min	avg	std	max	min	avg	std	max	min	avg	std	max	min	avg	std	max	min	avg	std	max	min	avg	std	max	min	avg	std	max	min	avg	std
Fcnt	78	14	48	22	73	26	56	16	60	27	43	13	76	45	57	12	71	1	34	22	64	10	46	15	**90**	44	65	15	**90**	48	65	14
Sp	–	–	–	–	**89**	19	75	21	69	3	41	22	69	5	31	20	65	1	36	22	75	7	42	23	**80**	13	54	19	78	12	52	20
Dp	–	–	–	–	–	–	–	–	**90**	36	68	17	76	25	52	16	**81**	3	40	25	52	3	28	19	**85**	65	76	7	**86**	64	76	7
AS	–	–	–	–	–	–	–	–	–	–	–	–	**89**	35	65	17	73	6	38	22	41	1	25	11	**87**	38	75	14	**88**	33	76	16
Sip	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	**90**	3	40	28	41	4	22	12	**93**	53	**81**	12	**94**	54	**83**	12
Dip	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	56	18	34	12	**90**	4	43	28	**93**	5	44	28
Country	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	51	2	25	18	54	1	24	18
BytesPP	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	**100**	**99**	**100**	0
Fsize	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Table A.4
Overview of traces used. To indicate the size of traces, we list the 75-percentile (75p) of flow counts computed in 5-min windows.

ID	Description	Start	Days	75p Fcnt (K)	
				TCP	UDP
1	Blaster worm	08/01/03	22	567	146
2	DNS attack	02/04/04	6	919	793
3	Witty worm	03/16/04	6	1095	304
4	Sasser worm	04/26/04	9	1068	276
5	YouTube outage	08/07/06	13	544	468
6	Telia fiber cut	08/12/07	26	877	921
7	Gant anomaly	10/17/07	6	954	1456
8	YouTube outage II	02/01/08	25	895	1404
9	Reflector DDoS	03/31/08	14	954	1479
10a	4 months (router 1)	02/29/08	120	930	1520
10b	"(router 2)"			442	618
10c	"(router 3)"			206	82
10d	"(router 4)"			547	623

correlation. All other pairs have an average correlation of less than 0.8.

Tables A.5 and A.6 show correlation statistics for traces 10a-d, studying the correlation between different routers during a 4 months period of relatively *normal* traffic, containing no major anomaly. Table A.5 shows the correlation for TCP traffic and Table A.6 for UDP traffic respectively. For TCP, again correlation is in general very low, the only exception being (Sp,Dp) with correlations between 0.96 and 0.98. Surprisingly, the three most correlated pairs from Table A.3 are not at all correlated in traces 10a-d, although both tables show statistics for TCP traffic. This suggests that correlation can vary significantly with time and between normal or anomalous traffic conditions. For UDP, there is quite a number of pairs with high maximum correlations. However, it is usually not stable over all routers, as the minimum correlation is quite weak for most of them. The only pair with constant strong correlation is again (Sp,Dp). However, while (Sp,Dp) is strongly correlated in normal traffic (traces 10a-d), it is only moderately correlated in anomalous traffic (traces 1–9).

Our findings suggest that different feature entropies provide useful and non-correlated information for detecting and classifying anomalies.

Besides a strong correlation of (Sp,Dp) in normal traffic, our results do not confirm the very strong correlation between src/dst port and address entropies in normal and anomalous traffic found by Nychis et al. [8], even though we used the more comprehensive Spearman correlation. We think these differences can largely be explained by the way the a_i (number of occurrences of item i) are calculated in (A.2). Nychis et al. compute a_i by counting the number of *packets* containing element i whereas we count the number of *flows* in accordance with other studies [6,2,1,13,3]. Clearly, the number of packets is highly correlated with overall traffic volume, whereas a high volume file transfer is usually summarized in a single flow. Thus, by computing the a_i using packet counts, one introduces a high correlation with traffic volume, and, in turn, also a pairwise correlation between different feature entropies.

Table A.5
Correlation of different feature entropies in percent for traces 10a-d (TCP). The table shows the maximum and minimum of 4 different routers for $H(X)$. Percentages of 80 percent or more are set in bold.

	Sp		Dp		AS		Sip		Dip		Country		BytesPP		Fsize	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
Fcnt	**94**	51	**93**	38	70	46	61	26	45	7	61	19	67	36	43	5
Sp	-	-	**98**	**96**	63	28	65	38	57	36	76	43	26	4	35	7
Dp	-	-	-	-	68	19	66	32	58	32	73	34	22	3	37	7
AS	-	-	-	-	–	–	**85**	62	45	14	29	18	43	9	44	23
Sip	–	–	–	–	–	–	–	–	64	58	70	42	23	15	27	12
Dip	–	–	–	–	–	–	–	–	–	–	**93**	54	58	7	67	7
Country	–	–	–	–	–	–	–	–	–	–	–	–	54	14	56	22
BytesPP	–	–	–	–	–	–	–	–	–	–	–	–	–	–	39	5
Fsize	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Table A.6
Correlation of different feature entropies in percent for traces 10a-d (UDP). The table shows the maximum and minimum of 4 different routers for $H(X)$. Percentages of 80 percent or more are set in bold.

	Sp		Dp		AS		Sip		Dip		Country		BytesPP		Fsize	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
Fcnt	**82**	73	**80**	64	**95**	63	**86**	7	**84**	13	**83**	14	78	13	**86**	12
Sp	–	–	**96**	**93**	79	65	79	29	70	27	78	42	64	1	76	6
Dp	–	–	–	–	78	49	79	46	72	18	64	39	63	2	74	2
AS	–	–	–	–	–	–	**89**	11	**94**	16	**84**	19	79	22	**96**	8
Sip	–	–	–	–	–	–	–	–	**89**	20	79	0	**87**	21	**91**	21
Dip	–	–	–	–	–	–	–	–	–	–	**92**	27	70	14	**94**	53
Country	–	–	–	–	–	–	–	–	–	–	–	–	47	1	**88**	8
BytesPP	–	–	–	–	–	–	–	–	–	–	–	–	–	–	78	4
Fsize	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Table A.7
IP address sets used to customize anomaly models. The ID column corresponds to the anomaly ID in Table 1.

ID	Attacker IPs	Victim IPs	Reflector IPs
1	EXT-IP	EXT-IP	INT-IPS-P80-LA-{500,2000}, INT-IPS-P80–5000
2	EXT-IP	EXT-IP	INT-IPS-HA-{500,2000,5000}, INT-IPS-P25-HA-{500,2000}
3	EXT-IP	INT-IP-{LA/HA}	EXT-IPS-P25-LA-2000, EXT-IPS-LA-500
4	EXT-IP	INT-IP-LA	EXT-IPS-P25-HA-500, EXT-IPS-HA-{2000,5000}
5	EXT-IP	INT-IP-HA	EXT-IPS-P25-HA-500, EXT-IPS-HA-{2000,5000}
6	EXT-IPS-LA-{5000,10000}	INT-IP-HA	n/a
7	EXT-IPS-LA-{5000,10000}	INT-IP-HA	n/a
8	EXT-IPS-RAND-2.5MIO	INT-IP-HA	n/a
9	EXT-IPS-RAND-2.5MIO	INT-IPS-RAND-0.5MIO	n/a
10	EXT-IPS-RAND-2.5MIO	INT-IPS-RAND-0.5MIO	n/a
11	INT-IPS-1000	EXT-IPS-20	n/a
12	EXT-IP	INT-IP-LA	n/a
13	EXT-IP	INT-IP-LA-1200	n/a
14	EXT-IP	INT-IP-LA-1200	n/a
15	EXT-IPS-LA-2000	INT-IP-LA	n/a
16	EXT-IPS-LA-2000	INT-IP-LA-1200	n/a
17	EXT-IPS-LA-2000	INT-IP-LA-1200	n/a
18	INT-IP	EXT-IP	n/a
19	INT-IP	EXT-IPS-2000	n/a
20	EXT-IP	INT-IP	n/a

In summary, we found *no strong feature* correlation that is invariant over time, different routers, and normal/anomalous traffic conditions. Hence, to build a broad information basis for modeling both normal and anomalous traffic, we make use of *all* these features in our entropy telescope.

A.3. Summary

We revisited the results of Nychis et al. [8] by performing an extensive correlation analysis of traffic feature entropies on a large data set containing traffic from a diverse set of customers. In contrast to Nychis et al., we did not find persistent strong correlation between traffic feature entropies. We believe the differences between our results and the findings of Nychis et al. can largely be explained by the way the a_i (number of occurrences of element i) are calculated in (A.2). Nychis et al. compute a_i by counting the number of *packets* containing element i whereas we count the number of *flows* in accordance with other studies [6,2,1,13,3]. Clearly, the number of packets is

highly correlated with overall traffic volume, whereas a high volume file transfer is usually summarized in a single flow. Thus, by computing the a_i using packet counts, one introduces a high correlation with traffic volume, and, in turn, also a pairwise correlation between different feature entropies.

Appendix B. Variation of IP address sets

The source and destination IP addresses for one instance of an anomaly of the base anomaly types described in Table 1 are determined as follows: For each flow, the source- and destination IP address are drawn from a set of IP addresses assigned to this anomaly. If multiple sets are assigned, only one of those set is used for a specific anomaly instance. But in total, all sets are used the same number of times. We built the following sets based on an analysis of the persistence and activity of IP addresses in our baseline trace:

- **IP:** A single fixed IP measured from real attacks.
- **IP-LA/ IP-HA:** An IP with low/high activity.
- **IPS:** IPs from all activity ranges.
- **IPS-HA:** IPs with high activity.
- **IPS-LA:** IPs with low activity.
- **IPS-Pxx:** IPs with activity on port xx.
- **IPS-Pxx-HA:** IPs with high activity on port xx.
- **IPS-Pxx-LA:** IPs with low activity on port xx.
- **IPS-RAND:** Randomly chosen IPs. They might or might not be present in the base trace.

An IP address shows low activity, if it occurs on a more or less regular basis but is not the source/destination of a significant number of flows (typically less than 10 flows per protocol and 5 min). An IP address showing high activity is one that occurs on a regular basis and is the source/destination of a significant number of flows (typically more than 100 flows per protocol and 5 min). To indicate the size of the sets, we append the number of IP addresses to the set name. Also, the prefixes INT and EXT denote whether IP addresses were chosen from the internal or external address range. For instance, the set INT-IPS-HA-5000 contains 5000 IP addresses randomly chosen from highly active internal addresses. Likewise, the set EXT-IPS-RAND-2.5MIO contains 2.5 million random addresses from the external range. Table A.7 shows which sets were used for which anomaly type.

References

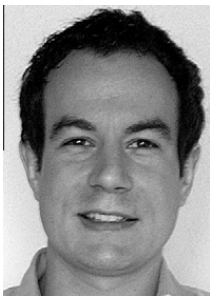
[1] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: ACM SIGCOMM, 2004.
[2] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: ACM SIGCOMM, 2005.
[3] D. Brauckhoff, K. Salamatian, M. May, Applying PCA for Traffic Anomaly Detection: Problems and Solutions, in: INFOCOM, 2009.
[4] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, A. Lakhina, Detection and identification of network anomalies using sketch subspaces, in: Internet Measurement Conference (IMC), 2006.
[5] A. Ziviani, M.L. Monsores, P.S.S. Rodrigues, A.T.A. Gomes, Network anomaly detection using nonextensive entropy, IEEE Communications Letters 11 (12) (2007) 1034–1036.
[6] B. Tellenbach, M. Burkhart, D. Sornette, T. Maillart, Beyond Shannon: Characterizing Internet Traffic with Generalized Entropy Metrics, in: Passive and Active Measurement Conference (PAM), 2009.
[7] A. Soule, K. Salamatian, N. Taft, Combining filtering and statistical methods for anomaly detection, in: Internet Measurement Conference (IMC), 2005.
[8] G. Nychis, V. Sekar, D.G. Andersen, H. Kim, H. Zhang, An empirical evaluation of entropy-based traffic anomaly detection, in: Internet Measurement Conference (IMC), 2008.
[9] H. Ringberg, M. Roughan, J. Rexford, The need for simulation in evaluating anomaly detectors, SIGCOMM Comput. Commun. Rev. 38 (1) (2008) 55–59.
[10] D. Brauckhoff, A. Wagner, M. May, FLAME: a flow-level anomaly modeling engine, in: Workshop on Cyber Security Experimentation and Test (CSET), 2008.
[11] B. Tellenbach, Collection of FLAME anomaly models, http://people.ee.ethz.ch/~betellen/AnomalyModels, visited on July 29, 2010.
[12] SWITCH, The Swiss education and research network, http://www.switch.ch, visited on July 29, 2010.
[13] A. Wagner, B. Plattner, Entropy based worm and anomaly detection in fast IP networks, in: IEEE WET ICE, 2005.
[14] H. Frank, S. Althoen, Statistics: Concepts and Applications, Cambridge University Press, 1994.
[15] T. Dübendorfer, A. Wagner, T. Hossmann, B. Plattner, Flow-level traffic analysis of the blaster and sobig worm outbreaks in an internet backbone, in: SIG SIDAR DIMVA, 2005.
[16] C. Shannon, D. Moore, The spread of the witty worm, IEEE Security and Privacy 2 (4) (2004) 46–50. doi: http://dx.doi.org/10.1109/MSP.2004.59.
[17] A. Dainotti, A. Pescap, G. Ventre, Worm traffic analysis and characterization, in: IEEE ICC, 2007.
[18] P. Barford, J. Kline, D. Plonka, A. Ron, A signal analysis of network traffic anomalies, in: Internet Measurement Workshop (IMW), 2002.
[19] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. Visited on July 29, 2010.
[20] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, Tech. rep., Department of Computer Science, National Taiwan University, 2009.
[21] R. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.
[22] M. Allman, V. Paxson, J. Terrell, A brief history of scanning, in: Internet Measurement Conference (IMC), 2007.
[23] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, S. Savage, Inferring internet denial-of-service activity, ACM Transactions on Computational System 24 (2) (2006) 115–139.
[24] VeriSign, Distributed Denial of Service (DDoS) Attacks: Latest Motivations and Methods, http://www.verisign.com/static/idefense_ddos_verisign_20080908.pdf, visited on July 29, 2010.
[25] R. Richardson, CSI computer crime and security survey 2008, Computer Security Institute.
[26] M. Stoecklin, Anomaly detection by finding feature distribution outliers, in: ACM CoNEXT, 2006.
[27] M.P. Stoecklin, J.-Y. LeBoudec, A. Kind, A two-layered anomaly detection technique based on multi-modal flow behavior models, in: Passive and Active Measurement Conference (PAM), 2008.
[28] B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, Sketch-based change detection: methods, evaluation, and applications, in: Internet Measurement Conference (IMC), 2003.
[29] X. Dimitropoulos, M. Stoecklin, P. Hurley, A. Kind, The eternal sunshine of the sketch data structure, Computer Networks 52 (17) (2008) 3248–3257.
[30] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, K. Cho, Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures, in: Workshop on Large Scale AttackDefense (LSAD), 2007.
[31] M. Alencar, F. Assis, A relation between the renyi distance of order and the variational distance, in: Telecommunications Symposium, 1998, pp. 242–244 vol.1. doi:10.1109/ITS.1998.713125.
[32] X. Song, M. Wu, C. Jermaine, S. Ranka, Statistical change detection for multi-dimensional data, in: ACM SIGKDD Conference on Knowledge Discovery and Data mining (KDD).
[33] Y. Gu, A. McCallum, D. Towsley, Detecting anomalies in network traffic using maximum entropy estimation, in: Internet Measurement Conference (IMC), 2005.
[34] M.Z. Shafiq, S.A. Khayam, M. Farooq, Improving accuracy of immune-inspired malware detectors by using intelligent features, in:

Conference on Genetic and Evolutionary Computation (GECCO), 2008.
[35] W. Guo, S. Cui, Fast convergence with q-expectation in em-based blind iterative detection, in: Asilomar Conference on Signals, Systems and Computers (ACSSC), 2006.
[36] M.E. Torres, M.M. An++ino, G. Schlotthauer, Automatic detection of slight parameter changes associated to complex biomedical signals using multiresolution *q*-entropy, Medical Engineering & Physics 25 (10) (2003) 859–867.
[37] J. Gao, W.W. Tung, Y. Cao, J. Hu, Y. Qi, Power-law sensitivity to initial conditions in a time series with applications to epileptic seizure detection, Physica A: Statistical Mechanics and its Applications 353 (2005) 613–624.
[38] Nonextensive statistical mechanics and thermodynamics: Bibliography, http://tsallis.cat.cbpf.br/biblio.htm, visited on July 29, 2010.
[39] M. Burkhart, D. Brauckhoff, M. May, E. Boschi, The risk-utility tradeoff for IP address truncation, in: ACM workshop on Network Data Anonymization (NDA), 2008.

Bernhard Tellenbach received his MS in Electrical Engineering and Information Technology from ETH Zurich, Switzerland, in 2005. Since 2005 he is pursuing his PhD in the Communication Systems Group at ETH Zurich. His research interests include network anomaly detection, network- and system security and network measurement. In parallel to pursuing his PhD, he started to work as a lecturer at the applied science university Rapperswil, Switzerland, in 2006 and as a security consultant at Consecom AG in 2008. He is a member of the board of the Information Security Society Switzerland and has served as a technical reviewer for several international journals and conferences. He has an issued patent in network anomaly detection.

Martin Burkhart received his MS in Computer Science from ETH Zurich, Switzerland, in 2003. From 2003–2007 he worked as a software engineer for the banking and logistics industry. Since 2007 he is pursuing his PhD in the Communication Systems Group at ETH Zurich. His research interests include Internet measurement, network anomaly detection, collaborative network security and applied cryptography. He has served as a technical reviewer for several international journals and conferences. He has an issued patent in network anomaly detection.

Dominik Schatzmann received his MS in Electrical Engineering and Information Technology from the ETH Zurich, Switzerland, in 2007. In 2007 he joined a startup company in Switzerland where he worked in the field of VoIP. Since 2008 he is pursuing his PhD in the Communication Systems Group at ETH Zurich. His research interests lie in the area of Internet measurements and network security.



David Gugelmann received his BSc in Electrical Engineering and Information Technology from ETH Zurich, Switzerland, in 2009. He is currently pursuing his master's degree taking courses at ETH Zurich and KTH Stockholm, Sweden. In late 2010, he will start his PhD in the Communication Systems Group at ETH Zurich. His research interests are in the areas of network security, large-scale network measurement and root cause analysis. He has been an intern at Grey Worldwide AG in 2005 and at Open Systems AG in 2009. Currently, he is working part-time at University of Zurich and ETH Zurich.

Didier Sornette trained as a statistical physicist, Didier Sornette is a strong advocate and practitioner of inter- and trans-disciplinarity, being a professor in three different academic departments at ETH Zurich. Didier Sornette is professor on the Chair of Entrepreneurial Risks at ETH Zurich, Switzerland and a professor associated with both the department of Physics and the department of Earth Sciences at ETH Zurich. He is also a professor of finance at the Swiss Finance Institute. His field of research concerns the diagnostic and prediction of crises, ruptures, bifurcations, catastrophes, tipping points and extreme events in complex systems, with applications including financial economics, social and computer networks, biology and medicine, and earthquakes.