Increasing Reliability in Network Traffic Anomaly Detection

Romain Thibault Fontugne

DOCTOR OF PHILOSOPHY

Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)

2011 (School Year)

September 2011

Abstract

Network traffic anomalies stand for a large fraction of the Internet traffic and compromise the performance of the network resources. Detecting and diagnosing these threats is a laborious and time consuming task that network operators face daily. During the last decade researchers have concentrated their efforts on this problem and proposed several tools to automate this task. Thereby, recent advances in anomaly detection have permitted to detect new or unknown anomalies by taking advantage of statistical analysis of the traffic. In spite of the advantages of these detection methods, researchers have reported several common drawbacks discrediting their use in practice. Indeed, the challenge of understanding the relation between the theory underlying these methods and the actual Internet traffic raises several issues. For example, the difficulty of selecting the optimal parameter set for these methods mitigates their performance and prevent network operators from using them. Moreover, due to the lack of ground truth data, approximate evaluations of these detection methods prevent to provide accurate feedback on them and increase their reliability. We address these issues, first, by proposing a pattern-recognition-based detection method that overcomes the common drawbacks of anomaly detectors based on statistical analysis, second, by providing both a benchmark tool that compares the results from diverse detectors and ground truth data obtained by combining several anomaly detectors.

The proposed pattern-recognition-based detector takes advantage of image processing techniques to provide intuitive outputs and parameter set. An adaptive mechanism automatically tuning its parameter set according to traffic fluctuations is also proposed. The resulting adaptive anomaly detector is easily usable in practice, performs a high detection rate, and provides intuitive description of the anomalies allowing to identify their root causes.

A benchmark methodology is also developed in order to compare several detectors based on different theoretical background. This methodology allows researchers to accurately identify the differences between the results of diverse detectors. We employ this methodology along with an unsupervised combination strategy to combine the output of four anomaly detectors. Thereby, the combination strategy increases the overall reliability of the combined detectors and it detects two times more anomalies than the best detector. We provide the results of this combination of detectors in the form of ground truth data containing various anomalies during 10 years of traffic.

Acknowledgements

I owe my deepest gratitude to my supervisor, Kensuke Fukuda, for welcoming me in his laboratory and providing me the opportunity to work in a unique research environment during the last three years. I will always be grateful for the numerous and constructive advice he taught me to succeed in my research, and, the precious time he spent with me for discussions or the multiple times he read and corrected my articles to make them acceptable for submission. Kensuke also introduced me to respectful researchers with whom I had the opportunity to discuss and cooperate on successful projects.

I thank the members of my PhD committee, Motonori Nakamura, Yusheng Ji, Shigeki Yamada, Michihiro Koibuchi, and Toshiharu Sugawara, for their constructive comments and invaluable advice. I am indebted to them for investing time and effort discussing anomaly detection and suggesting directions to guide me through my PhD.

I am particularly grateful to the researchers from the CNRS and ENS Lyon I have been working with during my PhD, namely, Patrice Abry, Pierre Borgnat, Guillaume Dewaele, and Eric Fleury. Conversations with them have been fruitful and greatly contributed to my PhD, in particular, they introduced me to community mining algorithms that solved most of the difficulties I was facing to benchmark anomaly detectors. I am indebted to Guillaume Dewaele for writing and sharing his code of the Gamma-based detector.

I thank the members of the WIDE project for their support, especially Kenjiro Cho who provides and maintains the MAWI archive, the main data set I have been analyzing during these 3 years, and, Youki Kadobayashi for developing the admd format and associated tools. I also appreciated the diverse opportunities offered by the WIDE project to discuss with outstanding researchers from all over the world.

I thank all my lab mates for sharing their experiences, reporting their feedback on my work, and translating me the rules of the game of research; including people from the university of Tokyo, Yosuke Himura, Hirochika Asai, Hideya Ochiai, Takuya Motodate, and Yoshiki Kanda from Waseda university. I owe special thanks to Yosuke Himura, he has been particularly helpful during the early years of my stay in Japan, our intense reflections on anomaly detection have been central to initiate my PhD. I am also indebted to Yoshiki Kanda for writing and sharing his code of the PCA-based detector.

Certainly none of this work would exist without the financial and human support of the National Institute of Informatics (NII), I thank the NII for funding my PhD and the NII secretaries for their patience in helping me to decipher Japanese paperwork. My stay in Japan was also really enjoyable thanks to the nice people I have met in NII and that became invaluable friends with whom I

spent wonderful time.

I express my deepest and sincere gratitude to my brother, Raphaël, and his ever growing family for their unconditional love and encouragements.

This dissertation is dedicated to my parents, Huguette and Denis Fontugne, who have given me the opportunity of an education from the best institutions, and for their love and support throughout my life that are the keys of all my achievements.

Contents

Chapter 1

Introduction

The success of Internet services results in a constant network traffic growth along with an increasing number of anomalies such as remote attacks (e.g., DoS attack, port scan, worm spreading) and misconfigurations. These anomalies represent a large fraction of the Internet traffic that is unwanted and penalizes legitimate users from accessing optimal network resources. Therefore, detecting and diagnosing these threats are crucial tasks for network operators that are trying to maintain the Internet resources made available. Due to the important traffic volume, quickly and accurately identifying anomalies in Internet traffic requires automation. Intensive studies have been carried out in this field, but the proposed anomaly detection methods still have important drawbacks [60, 32] that discredit their practical usage in real environment.

The goal of this dissertation is to increase the reliability in anomaly detection by providing methods and directions that overcome the common drawbacks of current anomaly detectors.

1.1 Problem statement

We differentiate two categories of method identifying anomalous traffic, those based on signature matching and those based on statistical analysis. The signature-based detectors are conceived to analyze enterprise network traffic, they are reliable detectors but their design and computation complexity prevent their use in the core of the Internet. Thereby, the identification of anomalies in Internet traffic is usually addressed by conducting a statistical analysis of the traffic. These methods follow a common approach; first, the traffic is modeled and a reference representing normal traffic is computed, second, the traffic that is significantly distant from the computed reference is reported as anomalous. The main advantage of these methods is their ability to identify emerging and unknown anomalies, contrarily to signature-based detectors that are relying on a signature database requiring updates when a new anomaly is discovered. Since our main interest is in Internet traffic, the work conducted in this dissertation is focusing on statistical-based methods and their common practical drawbacks.

1.1.1 Anomaly detectors in practice

Statistical analysis is an appealing approach to solve the anomaly detection problem, however, resulting anomaly detectors suffer from high error rate as investigating the output of statistical tools and tuning their parameter set in accordance to the analyzed network traffic is challenging.

Detection performance

Contrarily to signature-based detectors that implement pattern matching techniques, anomaly detectors based on statistical analysis are discriminating anomalous traffic according to its singular characteristics. Thereby, these two kinds of detectors are fundamentally different; signature-based ones are inspecting the content of the traffic looking for well-known patterns, whereas, statistical-based ones profile the traffic and discriminate traffic that is distant from a computed reference representing normal traffic. Consequently, the statistical-based anomaly detectors have the advantage of detecting new and unknown anomalies that are missed by the signature-based ones; symmetrically, their drawback is to frequently report benign traffic as anomalous whereas misreports are rare using the signature-based detectors.

Moreover, statistical-based anomaly detectors are inherently misreporting traffic during important anomaly outbreaks that alter the majority of the traffic. Indeed, when anomalous traffic is dominant the computed reference is contaminated and anomalies are confused with benign traffic.

Because estimating relevant statistics from small (mice) flows is difficult, several statistical-based detectors omit these flows, however, detecting low-intensity anomalous traffic is essential since sophisticated or large-scale attacks tend to be distributed processes involving numerous hosts with small amount of traffic each.

The main challenge in designing and using an anomaly detector is to maximize the number of detected anomalies while keeping the misreport rate low. In practice this trade off is usually controlled by adjusting the parameter set of the anomaly detector according to the analyzed traffic and the theory underlying the detector.

Network traffic abstraction

In order to formalize the anomaly detection problem researchers translate the traffic to statistical observations that are easily manipulated using statistical tools. This abstract representation of the traffic allows researchers to apply appropriate statistical tools (e.g., outlier detection methods), however, understanding the relations between the analyzed traffic and the mechanisms of the statistical tools is challenging. For example, outlier detection methods monitor certain properties of the traffic using different metrics (e.g., the entropy of the traffic) and discriminate anomalous traffic by looking at abnormal values. Thereby, outlier detection methods report statistical values characterizing anomalies, however, inferring from this output the root causes of the anomalies is usually difficult. Similarly, the parameter set of the outlier detection methods usually consists of thresholds that are difficult to interpret in term of network traffic. This dissertation addresses these two significant drawbacks of

current anomaly detectors, namely clarifying the output and selecting optimal parameter set in anomaly detection.

- **Detector output:** Understanding the output of an anomaly detector is the first step towards feedback and improvement. Accurately pinpointing the anomalous flows corresponding to an alarm reported by an anomaly detector requires extra efforts because of the traffic aggregation and the level of abstraction of statistical methods. Furthermore, contrarily to signature-based detectors that inherently match anomalous traffic to its root cause, anomaly detectors based on statistical analysis report anomalous traffic regarding its differences with the majority of the traffic but are unable to indicate its root cause (e.g., DoS attack, scan, or worm).

  Researchers proposed several methods precisely extracting the anomalous traffic corresponding to reported alarms [48, 16, 13], however, the root cause analysis is a laborious and time consuming task left to the network operators. Namely, operators investigate the traffic features extracted by the detection methods (e.g., list of IP addresses [48, 16]) and infer the root cause of the anomalies based on their knowledge and intuition.

  Recently a few works have been focusing on the automation of this manual process, for example Silveira and Diot [66] proposed a tool classifying anomalous flows reported by a detector and infer the root cause of the anomalies. Interestingly this tool is fundamentally independent from the mechanisms of the anomaly detectors and help in understanding the output of any detectors, however, as it does not take into account the principles of the detectors this tool prevent from reporting feedback on the detectors. For example, it is impractical for adjusting the parameter settings of an anomaly detector as it ignores the theory underlying the detectors and its relation with analyzed traffic.

- **Parameter tuning:** Wrong parameter setting dramatically alters the performance of anomaly detectors, thus, the tuning of the parameter set requires particular attention to ensure detectors reliability.

  The parameter set of an anomaly detector is mainly the one of the statistical analysis underlying, and usually consists of parameters for modeling the traffic and thresholds discriminating abnormal traffic. Setting these parameters require a strong understanding of the statistical tools and their impact on the detection of anomalous traffic is rarely well understood. Therefore, in practice network operators arbitrarily adjust the parameter set of the anomaly detectors and select the optimal parameter set through a laborious task involving many time consuming trials and errors.

  Only a few works have investigated this important drawback currently discrediting anomaly detectors. A careful study of the detectors based on principal component analysis (PCA) was carried out by Ringberg et al. [60]; although they identified four main challenges, including the sensitivity of the parameter set to analyzed traffic, these challenges are left unsolved. In addition, an attempt to automatically tune a method based on gamma modeling and sketches was conducted by Himura et al. [32]. They designed a learning process for predicting the optimal parameters

regarding the best parameters for past data. However, this method suffers from a high error rate as unexpected events do appear.

Understanding the relations between the real Internet traffic and the theory underlying the anomaly detectors is the main challenge that prevents to quickly and accurately tune their parameter set.

1.1.2 Benchmarking anomaly detectors

Benchmarking an anomaly detector is the crucial task that enables to identify its weaknesses and strengths, consequently, to provide feedback that is necessary to improve its detection performance. Ideally the performance of an anomaly detection method is evaluated using a benchmark data set in which anomalous traffic is precisely located beforehand using a trustworthy method. This annotated traffic is hereafter referred as ground truth data.

In order to perform reliable and rigorous evaluation the ground truth data is expected to have the three following properties:

- It contains real Internet traffic, thus, the evaluation environment is similar to the real world conditions and network operators can rely on the detection rate measured during the evaluation.
- Similarly, the anomalous traffic located in the ground truth data is representative of all classes of anomalies existing in the Internet. Thereby, the evaluation emphasize, both, the classes of anomaly the detector is reporting and the classes of anomaly the detector is missing. Moreover, as new Internet anomalies are constantly emerging ground truth data needs constant updates.
- The data is publicly available so the the research community is able to reproduce the experiments and the results from several detectors are comparable.

Nevertheless, the lack of trustworthy anomaly detection method prevents the existence of such ground truth data and constrains researchers to evaluate their anomaly detectors using less reliable evaluation methodologies.

For example, two methodologies involving manual inspection are; (1) the result of the anomaly detector is manually inspected and validated by the researchers, (2) the anomalies manually located in the analyzed traffic are used as a ground truth data. These two methodologies are time consuming and error prone as they rely on the knowledge and intuition of the researchers, furthermore, these evaluations are not reproducible by the research community.

Researchers also employ another evaluation methodology that consists in building a ground truth data by injecting anomalous traffic into simulated or real Internet traffic. In spite of enabling a thorough evaluation on the injected anomalies, these evaluations are constrained to a specific kind of anomalies and they usually come with additional evaluations using real traffic [59] as simulating the diversity of the Internet traffic is impractical [19].

Another usual methodology to evaluate an anomaly detector is to compare its results to the state of the art. The result of the proposed detector is compared to the one of a well-known detector, usually the dissimilarities in their output are inspected to argue the benefits or disadvantages of the proposed detector.

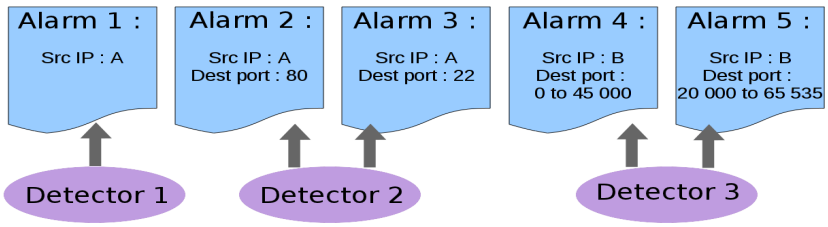


Figure 1.1: Five alarms reported by three distinct detectors. Alarm 1, Alarm 2 and Alarm 3 report different traffics from the same host. A same port scan is reported by two alarms; Alarm 4 identifies only a part of it (beginning of the port range), whereas Alarm 5 identifies another part (the end of the port range).

Although comparing diverse anomaly detectors based on different statistical tools is particularly appealing, it raises several difficulties that have been rarely addressed in the literature. In order to efficiently compare and evaluate anomaly detectors this dissertation thoroughly investigates these difficulties.

Comparing detectors output

Comparing outputs from diverse detectors seems at first glance to be trivial, but in practice, it is a baffling problem. The main issue is that detectors report different features of the traffic that are representing distinct traffic granularities and are difficult to systematically compare. The different traffic granularities of the reported traffic results from the diverse traffic abstractions, dimensionality reductions and theoretical tools employed by the detectors. For example:

- hash based (sketch) anomaly detectors [16, 48] usually report only IP addresses and corresponding time bin, no other information (e.g. port number) describes identified anomalies.
- The pattern-recognition-based anomaly detector proposed in Chapter 4 reports anomalies as sets of IP addresses, port numbers and timestamps corresponding to groups of packets identified in analyzed pictures.
- Several detection methods take advantage of clustering techniques to identify anomalous traffic [63]. These methods classify flows in several groups and report clusters with abnormal properties. Thereby, traffic reported by these methods consists of sets of flows.

These different kinds of alarm provide distinct details of the traffic that are difficult to systematically compare. A usual way is to digest all of them to a less restrictive form; namely, by examining only the source or destination IP addresses (assuming that anomaly detectors report at least one IP address), however, this simplification is error prone.

For example, Figure 1.1 illustrates five alarms reported by three distinct detectors. In this case comparing only IP addresses permits to determine that *Alarm 1*, *Alarm 2* and *Alarm 3* are similar. However, the port numbers provided by *Alarm 2* and *Alarm 3* indicate that these two alarms represent distinct traffics. Consequently, an accurate comparison of these two alarms requires to

also take into account port numbers, but it raises other issues. First, a heuristic is needed to make a decision when port number is not reported (for example in comparing *Alarm 1* and *Alarm 2*). Second, fuzzy equality is required to compare *Alarm 4* and *Alarm 5* of Fig.1.1. So forth, inspecting various traffic features reported by alarms makes the task harder although the accuracy of the comparison increases.

1.2 Dissertation contributions

The work conducted in this dissertation aims at increasing the reliability in anomaly detection. Our contributions are apparent at two scopes of the anomaly detection domain; (1) we propose a reliable anomaly detector that overcomes specific drawbacks identified in current detectors, (2) within a broader scope, we propose a benchmark methodology that helps researchers in increasing the reliability of their detectors.

1.2.1 Pattern-recognition-based anomaly detector

We address limitations of current anomaly detectors by developing an anomaly detection method based on a technique from image processing and pattern recognition. This detection method monitors the traffic in four picture categories standing for four traffic features, namely source IP address, destination IP address, source port, and destination port. These pictures are designed to highlight the anomalous traffic as linear patterns easily identifiable using a pattern recognition technique called the Hough transform. These linear patterns are illustrating a specific characteristic of anomalous traffic, which is, its abnormal distribution in the traffic feature space (i.e., IP address or port number spaces). The effectiveness of the proposed anomaly detector is validated using real Internet traffic, and its results are compared to anomaly detectors based on different theoretical backgrounds. The comparison indicates that the only anomalies detected by the pattern-recognition-based method are mainly malicious traffic with a few packets, and the proposed method has the advantage of reporting precisely the anomalous packets or flows.

Advantages of pattern recognition

By applying a pattern recognition technique the proposed anomaly detector is able to detect new and unknown anomalies whereas it does not suffer from the usual drawbacks of the outlier-based detectors. Indeed the proposed detection method detects traffic that is statistically similar to a general pattern characterizing unspecified anomalies, thus, it reports traffic that is featuring anomalous behavior. Contrarily, outlier-based detectors assume that the anomalous traffic stands for a minority of the traffic and reports traffic having singularities. Therefore, the proposed detector has the advantage of performing better in the specific case where the majority of the traffic is altered by an anomaly (e.g., during worm outbreak) whereas the outlier-based detectors tends to misreport normal traffic in this case [60, 62]. The proposed detection method also has the advantage of featuring an intuitive output and parameter set.

Detector output

The simple image processing underlying the proposed method permits network operators to intuitively investigate the output of this anomaly detector. Indeed the traffic is monitored in pictures that are familiar to the network operators, thus, the output of the proposed detection method is easily understood by network operators. In addition the proposed anomaly detector helps network operators in investigating anomalous traffic as it inherently provides the root cause of the anomalies. For example the identification of traffic abnormally dispersed in a picture representing the destination port space is intuitively translated as a port scan by network operators.

Parameter tuning

We also propose to ease the use of the pattern-recognition-based detector by investigating the relationship between its parameter set and the traffic characteristics. Therefore, we analyze the evolution of the optimal parameter set required to analyze a fluctuating traffic, and uncover the relations between the two. This analysis highlights that constantly achieving a high detection rate requires continuous adjustments to the parameters according to the traffic fluctuations. Therefore, an adaptive time interval mechanism is proposed to enhance the robustness of the detection method to traffic variations. This adaptive mechanism is tracking the characteristic of the traffic and constantly adjust the time interval of the detection algorithm to ensure its optimal performance. We validate the effectiveness of this adaptive anomaly detection method by comparison with three other anomaly detectors using four years of real backbone traffic. The evaluation reveals that the proposed adaptive detection method outperforms the other methods in terms of the true positive and false positive rate, thus, the adaptive mechanism enables the pattern-recognition-based detector to be more reliable and easier to deploy in practice.

1.2.2 Benchmarking anomaly detectors

In order to assist researchers in increasing the reliability of their anomaly detectors we provide them with tools that efficiently evaluate the detection performance of their detectors. We address two issues faced by researchers when evaluating their anomaly detectors; comparing diverse anomaly detectors and ground truth data.

First, we propose a benchmark methodology based on graph theory that allows to compare the results of diverse anomaly detectors. Second, we employ this methodology to study four combination strategies using diverse detectors and we provide the results of the best combination strategy in the form of ground truth data.

Comparing diverse detectors

One of the main contribution of this dissertation is to provide a reliable methodology that compares the output of any kinds of anomaly detector. This methodology is based on graph-theory and is independent from the mechanisms underlying the detectors.

The proposed benchmark method takes as input the traffic and corresponding alarms reported by several detectors, and it constructs a graph where a node represents an alarm and an edge stands for the common traffic between two alarms. This graph emphasizes the similarities between reported alarms and identical alarms are uncovered using a community mining algorithm. Thereby, this proposed benchmark method outputs groups of alarms that are identifying the same anomalous event.

This benchmark methodology enables fundamental advances in anomaly detection. Currently anomaly detectors are usually compared by the mean of receiver operating characteristics (ROC) curves that are taking into account the ratio of anomalies correctly reported (true positive rate) and benign traffic misreported (false positive rate). The ROC curves illustrate the accuracy of detectors, however, they do not provide any information on the characteristics of the traffic that is reported by the detectors.

The benefit of the proposed method is to help researchers in investigating the results obtained from their algorithms. For instance, while developing an anomaly detector, researchers commonly face a problem in tuning the parameter set. Therefore, researchers usually run their application with numerous parameter settings, and the best parameter set is selected by looking at the highest detection rate. Although this process is commonly accepted by the community a crucial issue still remains. For instance, a parameter set A may give a similar detection rate to that obtained with a parameter set B, but a deeper analysis of reported alarms may show that B is more effective for a certain kind of anomalies not detectable with the parameter set A (and vice versa). Deciding if A or B is the best parameter is then not straightforward. This interesting case is not solved by simply comparing detection rates. The overlap of both outputs as exhibited by our method would help, first, to compare in which conditions a parameter set is more effective, second, to make methods collaborate.

Synergy between diverse anomaly detectors

We apply the proposed benchmark method to implement a reliable anomaly detection system that combines diverse detectors. Using this combination of detectors we aim at automatically locating anomalies in a large traffic database (i.e., the MAWI archive) and providing the research community with pseudo ground truth data.

The basic approach underlying the proposed system is as follow; first, the benchmark methodology aggregates similar alarms from diverse anomaly detectors, second, each group of similar alarms is investigated by a combination strategy that determines if the group stands for anomalous traffic or not. We select and implement an unsupervised combination strategy based on dimensionality reduction that is simple to use in practice as it requires no training phase.

This system is evaluated using four independent detectors and 10 years of backbone traffic. According to the evaluation the combination strategy permits to detect twice more anomalies than the most accurate detector, and to reject the numerous false positive alarms reported by the detectors. Significant anomalous traffic features are extracted from the numerous alarms reported by the detectors, thus, the labels assigned to the MAWI archive are concise. Our results are publicly available and updated daily with new traffic in order to

provide updated ground truth data to the research community. Furthermore, our approach permits to include the results of upcoming anomaly detectors, to increase the quality and variety of labels over time.

This work highlights the reliability increase achieved by cross-validating diverse anomaly detectors from the state of the art. Consequently, we emphasize that combining detectors is a promising approach to solve the anomaly detection problem, thus, it deserves more attention in future work.

1.3 Dissertation outline

The remaining of this dissertation consists of seven chapters:

- Chapter 2 summarizes the state of the art in anomaly detection.
- Chapter 3 describes the data set analyzed in our work, some terminologies that are specific to the domain of anomaly detection and three anomaly detectors from the literature with their results on the considered data set.
- Chapter 4 is the detailed description of the pattern-recognition-based detector we proposed. We inspect its results using six years of real Internet traffic and compare its performance to two other anomaly detectors.
- Chapter 5 describes substantial improvements of the proposed pattern-recognition-based detector including a mechanism that helps in automatically tuning its parameter set. The efficiency of these improvements are evaluated by comparison with three other anomaly detectors using four years of Internet traffic.
- Chapter 6 proposes a graph-based methodology that allows to systematically compare results from diverse anomaly detectors. This methodology is applied to combine anomaly detectors and provide ground truth data.
- Chapter 7 summarizes the contributions, shortcomings and consequences of this work.
- Chapter 8 concludes this dissertation.

Chapter 2

Related Work

2.1 Anomaly Detection

Detecting anomalous traffic is a research topic that had recently received a lot of attention. We categorize this topic into two domains; network intrusion detection and Internet traffic anomaly detection. The goal of intrusion detection is to protect a network from remote threats, thus, the detection method is monitoring the traffic at the edge of the protected network where complete flows and packet payload are usually accessible. In contrast, Internet traffic anomaly detection aims at identifying anomalous traffic that is transiting in the core of the Internet where the monitored traffic is asymmetric due to routing policies, thus, flows are incomplete. Our work is dedicated exclusively to Internet traffic anomaly detection, thus, in this dissertation anomaly detection refers only to this specific domain.

For the last decade researchers have taken a strong interest in anomaly detection and proposed different detection methods that are basically monitoring traffic characteristics and discriminating outliers. We differentiate different categories of anomaly detection method; the methods monitoring the traffic volume and those monitoring the distribution of traffic features.

2.1.1 Volume based anomaly detectors

Volume based approaches are monitoring the number of bytes, packets or flows transmitted over time and aims at detecting abnormal variances that represent abusive usages of network resources or resource failures. Several methods have been proposed to effectively identify local and global traffic volume variances that stand for respectively short and long lasting anomalies.

For example, Barford et al. [9] proposed a method based on wavelet[6] that inspects the traffic volume at different frequencies. Their approach makes use of the wavelet analysis to dissect the traffic into three distinct signals representing local, normal and global variances of the traffic. The decomposed signals are analyzed by a detection procedure that finds the irregularities and reports the period of time they occur. Since the three signals represent the traffic at different time scales this approach is able to report short and long lasting anomalies. Nevertheless, as the whole traffic is aggregated into a single signal diagnosing

the detected anomalies is challenging and anomalous flows or IP addresses are left unknown.

Lakhina et al. [44] proposed a detection method that detects and diagnoses anomalies in large scale networks. First, their approach monitors the traffic using a matrix in which each cell represents the traffic volume of a link of the network at a certain time interval. Second, the main behavior of the traffic is extracted from the matrix with the principal component analysis (PCA) and anomalies are detected in residual traffic. Finally, the origin and destination nodes of the network that are affected by the anomalous traffic are identified and reported. Later PCA has received a lot of attention in this research domain; its main drawbacks have been identified [60] and several improvements have been proposed [62, 37, 45, 48].

Soule et al. [69] proposed another detection method that also monitors the traffic volume in matrices. The main idea underlying their approach is to represent in a matrix the traffic between nodes of a large network and remove the normal traffic using a Kalman filter. The residual traffic is analyzed with a statistical method that detects anomalous traffic and reports the pair of nodes affected by the anomalous traffic.

These volume-based anomaly detectors effectively report volume anomalies while their false positive rate is low. Their design, however, restrict them to report only a few classes of anomaly, thus, network operators need additional detectors to identify threats that are invisible in the traffic volume (e.g., network scan or port scan).

2.1.2 Traffic features based anomaly detectors

In order to overcome the drawbacks of volume-based anomaly detectors researchers proposed to refine the traffic features that are inspected by the anomaly detectors. For example, as numerous anomalies cause abnormal utilization of ports or addresses, inspecting the distribution of the traffic into the port and address spaces permits to identify anomalous traffic that is not reported by volume-based detectors (e.g., port scan). Nevertheless, due to the size of analyzed traffic inspecting detailed traffic features is costly and impose researchers to elaborate effective traffic aggregation schemes. The main challenge in aggregating network traffic is the trade off between maintaining a concise representation of the traffic and preserving its interesting characteristics. We discriminate four groups of detection method in regard to their traffic aggregation scheme; namely, (1) detection methods aggregating the traffic in a single signal, (2) those aggregating the traffic in traffic matrices, (3) methods aggregating traffic in histograms, and (4) the other methods.

Signal

A signal provides an intuitive and coarse view of the traffic by representing the time evolution of a single characteristic of the traffic. Contrarily to volume based method, here the analyzed signals are obtained from fine-grained measures providing details traffic characteristics.

The measure that probably have received the most attention in this research domain is the entropy (i.e., Shannon entropy). The entropy helps to quantify how the traffic is distributed in a specific traffic feature space; e.g., it allows

to measure if the traffic is concentrated on a certain IP address or if it is well distributed among several addresses. Nychis et al. [55] have studied in details the entropy of several traffic features in order to estimate their correlation and selected the best candidate.

A few other fine-grained measures have been studied, for example, Kim et al. [41] proposed to compute a signal representing the correlation over time of IP addresses or port numbers. Another original work has been recently proposed by Silveira et al. [67]; their approach monitors the stationarity of the traffic and reports anomalous traffic that violates the distribution of the flow.

Since signal analysis have been deeply investigated in the scope of detecting volume anomalies, the detection methods that are identifying anomalies in these fine-grained signals are similar to those of volume-based detectors. The wavelet analysis is particularly appreciated for its multi-scale capabilities [55, 41, 51].

Traffic matrix

A traffic matrix represents a time series of flows aggregated according to the ingress and egress routers they passed to transit on the network, also called, origin-destination flow (or OD flow). The effectiveness of aggregating traffic into traffic matrices have been validated in a comparative study by Soule et al. [68].

Perhaps the most famous anomaly detection method using traffic matrices is the PCA-based detector firstly proposed by Lakhina et al. [45] and deeply studied by others [48, 60, 62, 37]. Similarly to their volume-based anomaly detector they proposed an anomaly detector relying on PCA but analyzing the distribution of traffic features. The traffic distribution is observed using the entropy of four traffic features (i.e., source and destination address, and source and destination port) that allow them to identify numerous kinds of anomalous traffic. Therefore the monitored traffic is stored in four traffic matrices where each cell represents the entropy of a traffic feature for a certain OD flow and a specific time interval. Their PCA-based analysis identifies anomalies that affect the distribution of the traffic and reports OD flow. Since OD flows is an aggregation of usual traffic flows (i.e., the 5 tuple {*protocol, source IP, destination IP, source Port, destination Port*}), OD flows reported as anomalous by this PCA-based detector contains also benign flows that have to be filtered out from the output.

Li et al. [48] proposed to extend this work to obtain a more precise identification of anomalous traffic. Thereby, they infer the IP addresses responsible for the detected anomalies using random projections (also called sketches). Kanda et al. [37] also proposed to identify precisely the anomalous IP addresses and they also proposed a mechanism to adapt the PCA-based detector to analyze traffic from a single link (as opposed to Lakhina detector which monitors traffic at several links of a large network).

Histogram

In statistics the distributions of data is commonly studied in the form of histograms. Several works using histograms have been carried out in anomaly detection, for example Dewaele et al. [16] proposed to model flows in histograms and evaluate their geometry using the Gamma distribution model. The normal

behavior of the traffic is computed from the distributions of the traffic majority and outliers are reported as anomalous.

Another work taking advantage of histograms is proposed by Brauckhoff et al. [13]. In their work the distribution of several traffic features is monitored in a histogram representing a specific time interval. The histograms of consecutive time intervals are compared using the Kullback-Leibler divergence, if the difference between two histograms reaches a certain threshold the anomalous traffic is identified and reported.

Image

Similarly to our work a few image-based approaches have been proposed for anomaly detection. Kim and Reddy [40] introduced a way to represent the traffic as a movie and used a scene-change algorithm to detect significant changes in the traffic. This method uses image-processing techniques; it identifies the abrupt variances in traffic distribution and it has a short latency of detection. However, the design of frames is mainly based on packet counters and this restricts it being able to detect only those anomalies generating a large number of packets.

The anomaly detectors proposed in this dissertation (Chapter 4 and 5) also relies on image processing, however, the computed pictures monitor detailed traffic features (i.e., source IP address, destination IP address, source port, and destination port) and permit to identify abnormal distributions in these feature spaces.

2.2 Anomaly detector benchmark

Providing ground truth data to evaluate anomaly detectors is a challenge that has been addressed several times in the past. We distinguish two different approaches to evaluate an anomaly detector, namely using simulated or real traffic.

2.2.1 Simulated traffic

Evaluating an anomaly detector with simulated traffic is appealing as it allows researchers to work in a controlled environment where the characteristics of the anomalies can be customized [59]. For example, researchers can focus on a particular kind of anomalous traffic and vary the intensity of the anomalous traffic to measure the sensitivity of their anomaly detectors [67, 13, 44, 55, 62, 64].

Nevertheless, a thorough detector evaluation requires also real Internet traffic [59] as simulating the complexity and diversity of real traffic is in practice unfeasible [19].

2.2.2 Real traffic

Providing real Internet traffic for the evaluation of anomaly detectors is challenging for two main reasons; (1) labeling anomalous traffic in real Internet traffic is difficult because of the lack of trustworthy method and the traffic volume that makes manual labeling unpractical. (2) Providing Internet traffic is inherently problematic because of the privacy issues.

The DARPA Intrusion Detection Evaluation Program [50] has been a great effort to provide labeled traffic to evaluate intrusion detection systems (IDS). It has been extensively studied, mainly through the KDD Cup 1999 data (KDD'99), and has been a profitable support for researchers. The main distinctions between this work and ours are the size of the network measured and the detectors to be evaluated. The DARPA Intrusion Detection Evaluation Program focuses on the evaluation of IDS and provides labeled LAN traffic where the packet payload is available and flows are complete. Whereas our work focuses on the evaluation of backbone traffic anomaly detectors and we provide labeled backbone traffic where the packet payload is not available, and the flows are incomplete and asymmetric. Furthermore, several critical drawbacks of the KDD'99 have been reported [53]. Also, the traffic data was captured in 1998, hence it contains no traffic from recent applications or anomalies. Therefore, this data must be carefully used as it is not representative of real traffic [70] and does not contain recent anomalies.

Closer to our work, Owezarski [56] recently proposed a data set containing real backbone traffic where anomalies are precisely located. In this work the traffic is captured at different points in the RENATER network, which is supposed to be anomaly free, and the researchers generate two kinds of anomalies (i.e., flash crowd and DDoS attack). Their experiment consist of different scenarios where the intensity of the anomalies varies. Thus, the sensitivity of the detectors to DDoS and flash crowd is easily identified. However, there are only a few kinds of anomalies in their data and they are not a realistic representation of the diverse anomalies found on the Internet. Due to privacy issues, their data is not downloadable and only accessible by visiting their laboratory.

Being conscious of the shortcomings of previous works, this dissertation design a methodology aiming at providing to the research community a data set containing real Internet traffic with labeled anomalies.

2.3 Anomaly detector combination

Although the combination of classifiers is a hot topic in the clustering community [43], only a few works have been conducted in the field of network anomaly detection. The few proposals are classified into two groups; the combination strategies that requires a training phase (i.e., supervised combination), and those that make no prior assumptions (i.e., unsupervised combination).

2.3.1 Supervised combination strategies

A recent study on the combination of anomaly detectors was conducted by Ashfaq et al. [8]. They proposed a new combination strategy that takes into account the accuracy of the detectors; first, the accuracy of each detector is evaluated on a training data set, and then, the results of the detectors are combined regarding their accuracy. Their results emphasized the benefit of taking into account the detectors accuracies when combining them. Nevertheless, such methods increase the necessity of human intervention as they involve a training step.

2.3.2 Unsupervised combination strategies

Shanbhag and Wolf [65] have studied the combination of five rate-based detectors to accurately identify the real-time variance in traffic volume. They analyzed seven different combination strategies and emphasize that the best strategy improves the accuracy of the overall detectors. The goal of our work differs from theirs as they aim at detecting anomalies in real time by running several detectors in parallel. Thus, they restrict their study to a particular kind of computationally efficient anomaly detector (rate-based detector).

The approach proposed in this dissertation focuses on diverse anomaly detectors that are combined with unsupervised combination strategies.

Chapter 3

Preliminary Analysis

3.1 Dataset: the MAWI archive

The traffic we analyze through all this dissertation is captured by the WIDE Project and distributed in the form of a database called the MAWI (Measurement and Analysis on the WIDE Internet) archive [14]. This database contains daily traces representing 15 minutes of traffic captured from a trans-Pacific link between Japan and the United States. The MAWI archive started in January 2001, and thus, currently contains more than 10 years of traffic. Our study focuses on the traffic captured at the samplepoints B and F of the MAWI archive. In fact these two samplepoints are measuring at the same location a link that has been updated three times; Since 2001 the monitored interface was a 100 Mbps link with an 18 Mbps committed access rate (CAR) and is referred as the samplepoint B. It was replaced in 2006/07/01 by the samplepoint F which is a full 100 Mbps link that was updated to a 150 Mbps link in June 2007.

The MAWI archive is made publicly available on the Internet[1] since it respects users privacy by scrambling the IP addresses and omitting the packets payload. Therefore, the MAWI archive has enabled many researchers to study Internet traffic characteristics [12, 29, 38, 27], or evaluate anomaly detectors [16, 22, 20] and traffic classifiers [15, 49].

The main assets that make the MAWI archive a valuable support for studying network traffic are its size, the diversity of traffic it contains and its accessibility:

- The MAWI archive currently represents more than 10 years of traffic, therefore, it highlights the longitudinal characteristics of the network traffic and permits to study the evolution of Internet traffic [26]. In regard to anomaly detection, the MAWI archive allows us to inspect short and long lasting anomalies that appeared since 2001.

- Furthermore, the MAWI archive currently monitors daily several hundred thousand IP addresses that are using numerous applications. Thereby, it contains diverse anomalies ranging from well-know anomalies that have been identified in the previous works and other unknown anomalies that are hidden in the traffic or currently emerging. Consequently, the MAWI

[1] http://mawi.wide.ad.jp/mawi/

Table 3.1: Detection theory terminology.

		Detector result	
		Reported	Ignored
Ground truth data	Anomalous	true positive	false negative
	Benign	false positive	false negative

archive allows us to study various anomalies and enables a general analysis of anomalous traffic that is not restricted to a certain kind of anomaly.

- All traffic traces from the MAWI archive are freely downloadable on the Internet, thus, the networking research community is able to reproduce and compare results obtained using MAWI traffic.

3.2 Background

This section presents the terminology, metrics and heuristics that are used in this dissertation to evaluate the anomaly detectors.

3.2.1 Terminology

In order to evaluate the performance of an anomaly detector researchers require a data set in which anomalous traffic has been identified beforehand using a trustworthy method that is independent from the evaluated detector. This data set is commonly referred as ground truth data in the literature and this dissertation.

Using ground truth data researchers validate the effectiveness of their detectors by measuring the amount of traffic that falls into the four following categories:

- The traffic that is reported by both the evaluated detector and the trustworthy method (true positive).
- The traffic that is ignored by the evaluated detector but reported by the trustworthy method (false negative).
- The traffic that is reported by the evaluated detector but ignored by the trustworthy method (false positive).
- The traffic that is ignored by both the evaluated detector and the trustworthy method (true negative).

For clarity purpose these four different categories are referred using the concise terminology commonly used in the detection theory and illustrated in Table 3.1.

Nevertheless, in the domain of network traffic anomaly detection the lack of ground truth data makes the evaluation of anomaly detectors challenging. Therefore, researchers employ different approximate evaluation methods to evaluate the performance of the detectors. For instance, this dissertation falls back

Table 3.2: Heuristics deduced from main anomalies previously reported [12] and manual inspection of the MAWI archive.

Category	Label	Details
Attack	Sasser	Traffic on ports 1023/tcp, 5554/tcp or 9898/tcp
Attack	RPC	Traffic on port 135/tcp
Attack	Ping	High ICMP traffic
Attack	Other attacks (Flood)	Traffic with more than 50% of SYN, RST or FIN flag. And http, ftp, ssh, or dns traffic with more than 30% of flag SYN
Attack	NetBIOS	Traffic on ports 137/udp or 139/tcp
Special	Http	Traffic on ports 80/tcp and 8080/tcp with less than 30% of SYN flag
Special	dns, ftp, ssh	Traffic on ports 20/tcp, 21/tcp, 22/tcp or 53/tcp&udp with less than 30% of SYN flag
Unknown	Unknown	Traffic that does not match other heuristics

on a heuristic evaluation methodology which consists in benchmarking a detector with a pseudo ground truth data obtained using heuristics.

3.2.2 Heuristics and evaluation metric

Our evaluations of the proposed anomaly detectors rely on heuristics based on a manual inspection of the MAWI traffic and the anomalies previously reported in the literature [12].

Heuristics classifies traffic into three categories, *attack*, *special*, and *unknown* that help in investigating the results of the anomaly detectors. The attack category stands for traffic using port numbers commonly used by malicious code, ICMP traffic that is abnormally high, or TCP traffic with an aberrant proportion of SYN, RST, or FIN flag. The special category represents the traffic corresponding to well-known protocols (e.g., http and dns), while the unknown category stands for the traffic that is left unclassified.

An anomaly detector is expected to report more traffic classified as attacks than those labeled special or unknown. Thus, the *accuracy* of a detector is defined as the ratio of the alarms classified as attacks by the heuristics listed in Table 3.2:

$$Accuracy = \frac{\#attack}{\#attack + \#special + \#unknown}$$

Also, we ascertain the heuristics is fundamentally independent from the principle of the proposed detection methods, thus, it is appropriate to evaluate these anomaly detectors. Indeed, this heuristics is based on well-known port numbers and abnormal usages of TCP flags, whereas the detection methods proposed in

the following chapters uses only the port numbers as indexes and does not rely on the application information related to them nor the TCP flags.

3.3 Preliminary detection

Our understanding of the MAWI traffic and its anomalies is initiated by manually investigating several characteristics of the traffic and inspecting results from detectors that have been previously proposed in the literature.

3.3.1 Manual inspection

We manually inspect the MAWI traffic to identify the prominent anomalies that should be reported by the anomaly detectors.

We dissect the MAWI archive using the port-based classifier CoralReef [2] and inspect the traffic corresponding to each application. By monitoring the number of packets standing for each application we manually identify five anomalous events that are significantly altering the main behavior of the traffic (Figure 3.1(a)):

- From September to December 2003 we observed a substantial number of ICMP flows constituting a long-lasting ping flood. The root cause of this anomalous event is undetermined, however, it has seriously impacted the network resources as it represents 34.5% of the total number of packets transmitted from September to December 2003.

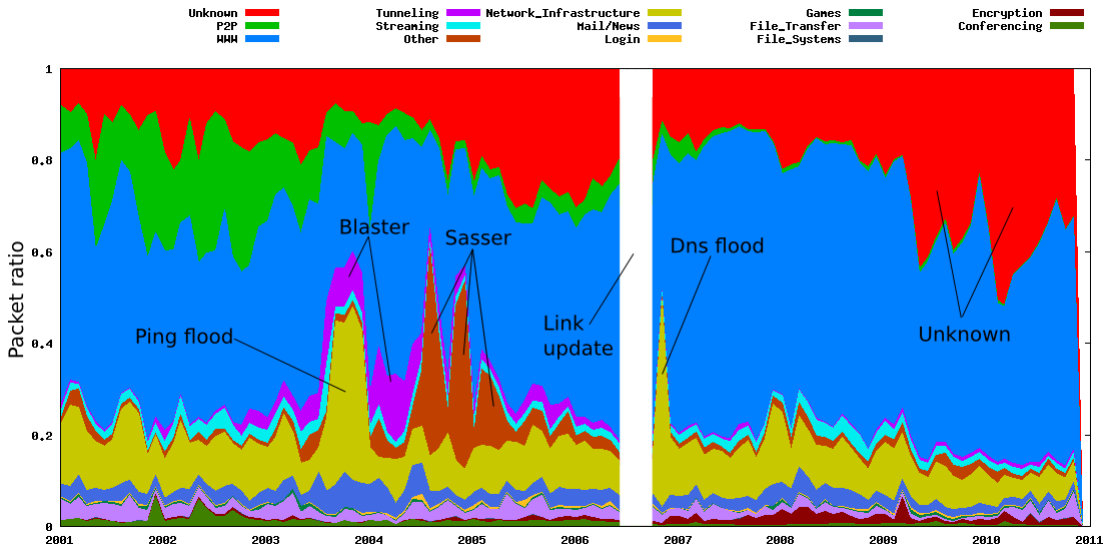
- From August 2003 the outbreak of the *Blaster* worm is observed in the MAWI traffic. This worm is spreading in the network through a Windows security hole and has been observed all over the world.

- From June 2004 to June 2005 another worm called *Sasser* is observed in the form of three peaks representing three outbreaks of different variants of this worm.

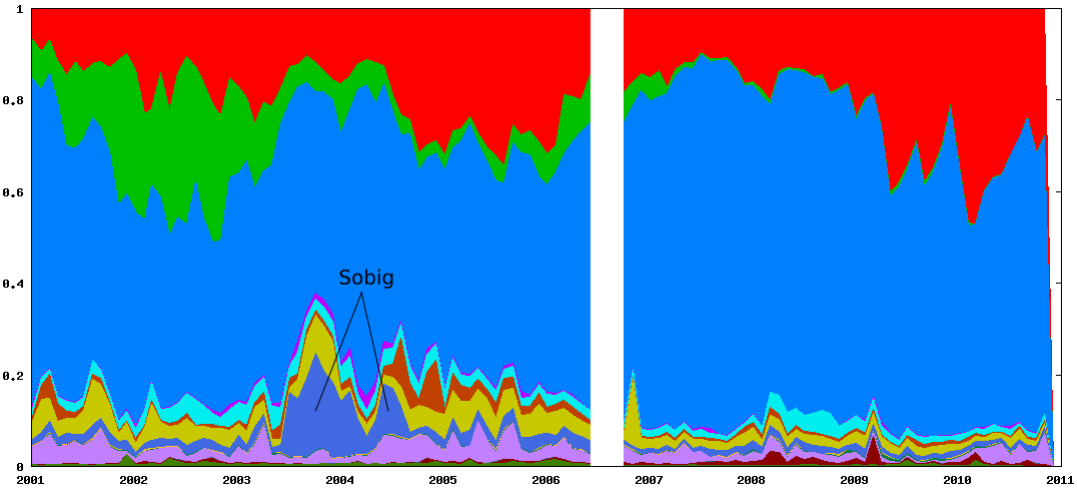
- After the update of the link in July 2006, an important traffic against DNS servers is observed. This traffic is particularly intense in the middle of November 2006, for example, the DNS traffic measured on the 2006/11/11 stands for 83% of all packets recorded this day.

- From 2009 the traffic classified as unknown has suddenly and dramatically increased. This traffic is difficult to investigate as it is observed on high port numbers for which there is no application related information, consequently, the root cause of this event is unclear. We notice, however, that a virulent worm called *Conficker* is using randomly high port numbers and emerged at a similar period of time.

Another anomalous event is observed by monitoring the traffic in regards to the number of bytes, it starts in June 2003 and stands for the outbreak of a worm spreading via emails that is called *Sobig* (Figure 3.1(b)).



(a) Packet breakdown of the MAWI archive.



(b) Byte breakdown of the MAWI archive.

Figure 3.1: Application breakdown of the MAWI archive from 2001 to 2011.

3.3.2 Current anomaly detectors

A deeper understanding of the MAWI traffic is achieved by analyzing the traffic with three anomaly detectors based on different theoretical backgrounds; one is based on the principal component analysis, one on the gamma modeling and one on the Kullback-Leibler divergence. We selected these three dissimilar detectors to inspect the results from different classes of anomaly detection method.

Principal component analysis

Principal component analysis (PCA) is a mathematical tool that projects a data set onto subspaces in which the variance of the data is maximized. Therefore, these subspaces highlight the main characteristics of the data set and help in classifying it.

The main approach underlying a PCA-based anomaly detector is, first, to monitor the traffic in a matrix, second, uncover the main characteristics of the traffic using PCA (these are considered as the profile of the normal traffic), and finally, report as anomalous the traffic that is featuring separate characteristics.

This approach based on PCA is perhaps the most studied technique for anomaly detection, it was first proposed by Lakhina et al. [44] to detect anomalous traffic transiting in a network, and it has received much attention in the last few years [48, 60, 62, 37].

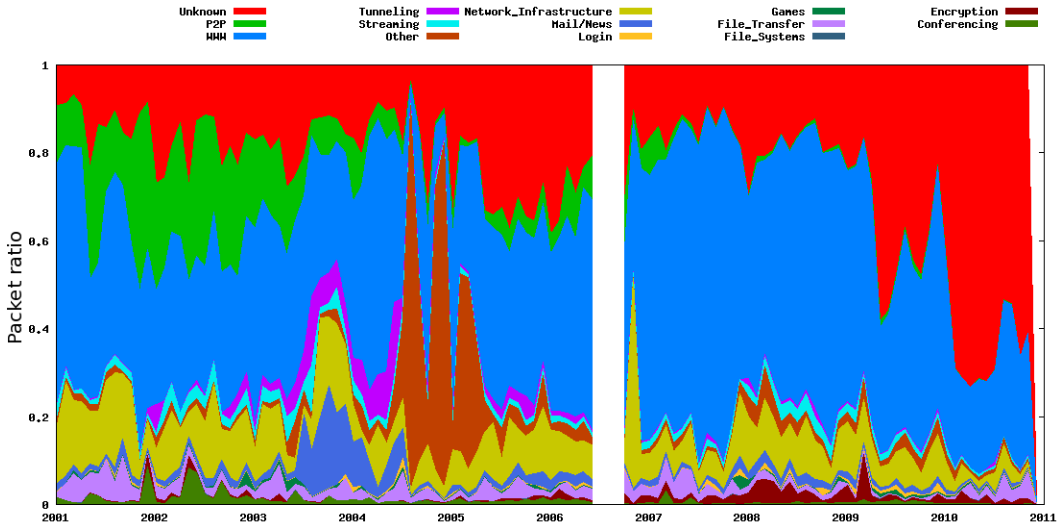
For instance, the PCA-based detector employed in our experiments is an improved version proposed by Kanda et al. [37] that overcomes two inherent problems of the original PCA-based detector proposed by Lahkina et al. [44], that are; the restriction of analyzing traffic from several links and the difficulty of precisely pinpointing the anomalous traffic due to traffic aggregation [60]. Indeed, the employed detector takes advantage of random projections (or sketches) [42, 48, 16] to analyze only traffic measured at a single link and retrieve the source IP addresses corresponding to the anomalous traffic. Hereafter we refer to this anomaly detection method as the PCA-based detector.

The result of the PCA-based detector using the MAWI traffic is illustrated by Figure 3.2(a). This detector detects the six prominent anomalies manually reported in the previous section, although, the Blaster worm and the ping flood are partially reported.

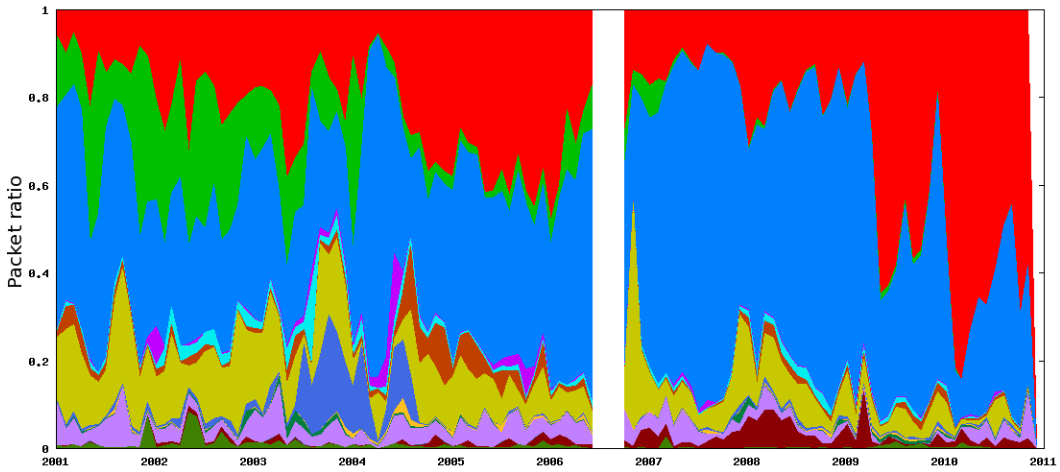
Gamma distribution model

The gamma distribution is a model that describes any probability distribution (Gaussian or not) with only two parameters. Dewaele et al. introduced an anomaly detection method based on sketches and multi-resolution gamma modeling [16]. Similarly to the PCA-based detector, this detection method uncovers the behavior of the main traffic and reports traffic with different behavior as anomalous. Nevertheless, this detection method relies on histograms and the gamma model that are fundamentally different from the PCA approach.

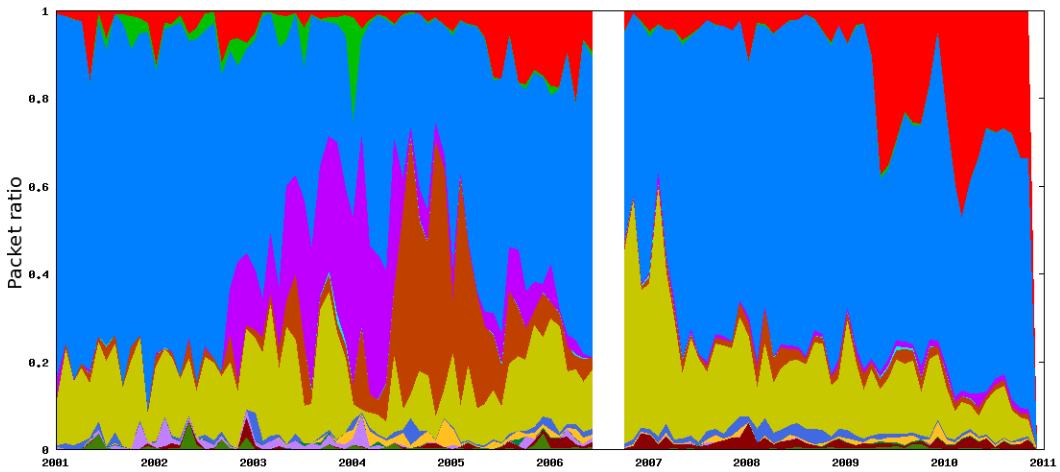
In a nutshell, this detection method splits the traffic into sketches that are monitored using histograms (contrarily to the PCA-based detector that is relying on traffic matrix). Afterwards, the sketches are modeled using the Gamma distribution and an adaptive reference standing for the normal traffic is computed. Thereby, the traffic that is distant from the computed reference is con-



(a) Traffic reported by the PCA-based detector.



(b) Traffic reported by the Gamma-based detector.



(c) Traffic reported by the KL-based detector.

Figure 3.2: Breakdown of the results of three detectors using the MAWI archive.

sidered as anomalous and the corresponding source and destination IP addresses are retrieved using the sketches.

The result of the Gamma-based anomaly detector using the MAWI traffic is illustrated by Figure 3.2(b). This detector successfully identified four prominent anomalies manually reported in Section 3.3.1, however, the Blaster and Sasser worms are not reported using this detector. A careful manual investigation revealed that the traffic corresponding to these worms consists mainly of small flows that are missed by this detector.

Kullback-Leibler divergence

The Kullback-Leibler (KL) divergence is a data differencing metric measuring the variance between two probability distributions. It has been applied to anomaly detection by Brauckhoff et al. [13] to detect the prominent changes in the traffic. Similarly to the gamma-based detection method this detector monitors in histograms the probability distribution of the traffic, however, its approach is fundamentally different as the computed reference representing normal traffic is obtained from previous observations.

The approach proposed by Brauckoff et al. [13] is to monitor the traffic in several kinds of histograms that monitor distinct traffic features and apply the Kullback-Leibler (KL) divergence to two consecutive observations. Consequently, abnormal variances in the distribution of the monitored traffic features result in high KL divergence values that are detected using an adaptive threshold. Traffic features that alter the distribution of the traffic are retrieved using sketches and allows to accurately extract anomalous traffic with an association rule mining algorithm. Thus, the alarms reported by this anomaly detector are association rules, namely 4-tuples (source and destination IP addresses, source and destination port numbers) where at most three elements can be omitted.

The result of the KL-based anomaly detector using the MAWI traffic is illustrated by Figure 3.2(c). We observe that this detector reports significantly different traffic compare to the PCA-based and the gamma-based ones. The KL-based detector successfully detected the Blaster and Sasser worms, whereas, it completely missed the Sobig worm. Other prominent anomalies are partially reported by this anomaly detector.

Chapter 4

Anomaly Detection based on the Hough Transform

4.1 Introduction

Identification of anomalies in Internet backbone traffic is an important task for securing operational networks and maintaining optimal network resources. However, analyzing traffic taken from a high speed Internet backbone — where the payload data is usually inaccessible, the traffic is asymmetric and often sampled — is a challenging issue. A significant difficulty is to accurately characterize anomalous traffic while a wide diversity of threat is constantly emerging. Researchers have mainly tried to handle anomaly detection as a statistical issue [9, 16, 45], but they have faced several common problems; normal traffic is misreported when anomalous traffic is dominant, mice flows are usually omitted, and they are in practice difficult to use as the parameter set and output requires advanced knowledge on the underlying statistical analysis.

The main idea of our work is to apply image processing and pattern recognition techniques to anomaly detection; traffic is monitored in 2-D scatter plot where each plot represents packets and anomalous traffics appear as "lines". Anomalies are easily extracted with a line detector and the original data can be retrieved from the identified plots. Thereby, the proposed approach is intuitive to network operators, it also has the advantage of quickly and precisely reporting anomalies involving mice flows, and it does not assume that the normal traffic is dominant. The method inspects only packet header information at a single point in the network, and it requires no prior information on the traffic or port numbers.

In [24] we proposed the basic idea of this new approach based on pattern recognition of network-related information. Also, the proposed method was partially validated with a single traffic trace. In this chapter, we thoroughly investigate this method; first, we estimate the dependencies of its parameter set. Next, we characterize anomalous behaviors in a large-scale publicly available traffic data set (for 6 years) taken from a trans-Pacific link. We also compare the results of our method with those of different methods based on multiresolution gamma modeling [16] and K-means [63]. Finally, we highlight the different strengths and weaknesses of each method, and emphasize the need for using

different detection approaches together.

4.2 Temporal and spatial behavior of anomalous traffic

In previous work we proposed a visualization tool that emphasize anomalies in 2-D pictures (see Appendix A), the observations made with this tool helped us in characterizing anomalous traffic.

Thereby, we identify anomalies through their unusual uses of network traffic features during a period of time. We consider four traffic features — source address, destination address, source port, and destination port — and demonstrate that anomalous traffic may be manifested by some of them having abnormal distributions. By mapping traffic into a 2-D space (one feature and time), anomalies can be intuitively identified as lines.

Figure 4.1 shows two scatter plots generated from the same traffic trace taken at a trans-Pacific link (MAWI Samplepoint-F 2007/01/09) [14]; the horizontal axes stand for time, while the vertical axes represent the source port space in the upper sub-figure and the destination port space in the lower one. The color of the plots indicates the amount of packets. The apparent "lines" represent excessive uses of traffic features; traffic is either concentrated on a specific instance of a feature (horizontal line), or dispersed on numerous instances (vertical and diagonal line). The angle of diagonal lines acquaints the propagation speed of traffic within the feature space observed.

For example the two "lines" labeled (a) in the upper panel clearly stand for malicious traffic since all source port numbers are used in only 14 minutes. Manual inspection reveals that it is only SYN packets initiated from the same source address and directed to a few destination addresses on port 443 (HTTP over SSL). This is a typical behavior of an attack against a protocol of the Microsoft SSL library. The other slanted "lines" are the same kinds of attack mounted against other services. In particular, label (b) in Fig. 1 corresponds to a DDoS attack against a few HTTP servers (SYN packets). Because the displayed traffic is bi-directional, we can see "lines" similar in the bottom scatter plot (b') representing the acknowledgments sent from the servers to the aggressors (SYN-ACK packets). Also, two kinds of "lines" are repeated several times (see labels (c) and (d) in Fig. 1); these are ACK floods from two distinct hosts against different targets. The horizontal "lines" are anomalies consuming bandwidth, such as DoS attacks, misconfigurations or heavy-hitters.

4.3 Anomaly detection method

On the basis of observations presented in Section 4.2, we devised an anomaly detection method based on pattern recognition [24]. The key idea underlying this approach is that traffic is monitored in pictures in which anomalous behaviors are displayed as "lines" that can be easily identified with a line detector.

This approach inspects only IP addresses and port numbers and requires no knowledge of the port numbers (such as application-related information). The method does not examine the packet payload, and its low computational time allows on-line detection.

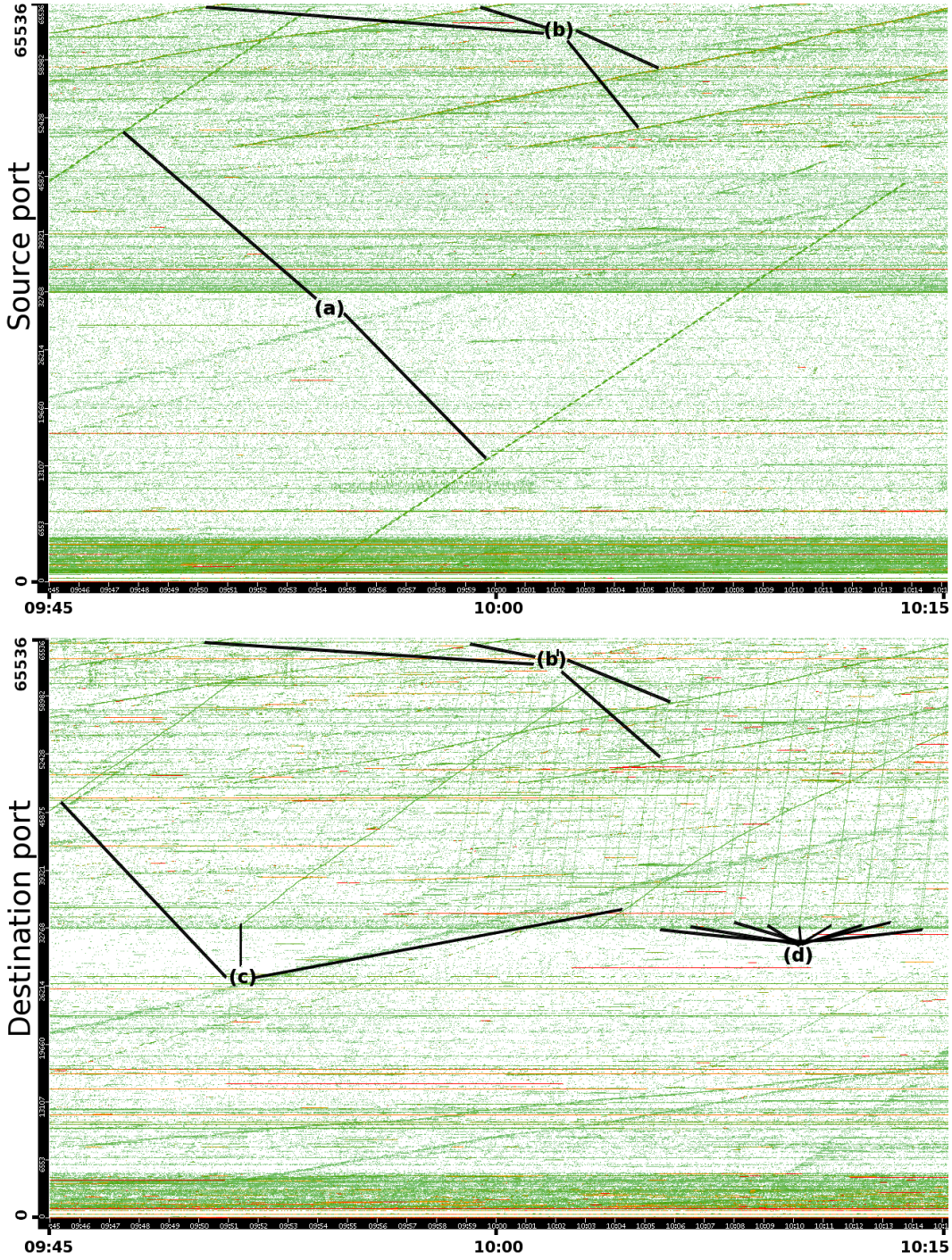


Figure 4.1: Scatter plots of trans-Pacific traffic data. Source port vs. time (top) and destination port vs. time (bottom).

Algorithm 1 Anomaly detection and classification

```
1:  f is the number of traffic features considered
2:  Set the sliding window at the beginning of the data
3:  while window != EOF do
4:    for all packets in the window do
5:      Plot packet in f pictures and store header information
6:    end for
7:    for all pictures do
8:      Compute the Hough space for the picture
9:      Extract lines from the Hough space
10:     for all lines found do
11:       New event e
12:       Retrieve all packet header from the line
13:       e ← Summarize traffic features from packet headers
14:       if ∃ anomaly a with main features = main features of e then
15:         Add e to a
16:       else
17:         Create a new anomaly
18:       end if
19:     end for
20:   end for
21:   Slide the window
22: end while
```

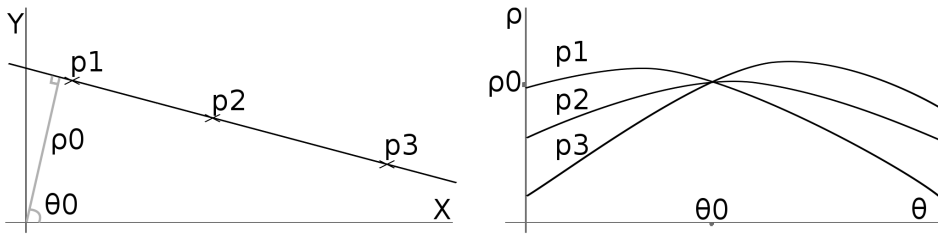
4.3.1 Algorithm

The pattern-recognition-based method is outlined as Algorithm 4.3.1. The core of the detection process consists of three steps:

Computation of pictures (lines 4-6) Four picture categories are considered to emphasize anomalous traffic ($f = 4$); all of them have time on the x axis and a different traffic feature on the y axis (source/destination address or port).

In order to reduce the IP address space and the port number space to match the size of pictures, we implemented two mechanisms. (1)Let say A is an IP address represented on 32 bits. v is the mapped value defined as $v = A \ mod \ 2^{\alpha}$ ($\alpha = 13$). Thus, A is mapped to a value in 2^{13} space. We divide this space into 16 pictures (512 pixels high) to improve the accuracy of the Hough transform. (2) Port numbers are directly aggregated into 16 pictures; in this case a pixel represents $2^{16}/16{*}512 = 8$ ports. All these values have been selected empirically and permit a low traffic aggregation not altering detection performance.

Detection: Hough transform (lines 8-9) Our method is based on a common image processing technique that extracts shapes in pictures; the so called Hough transform. The fundament of this technique was introduced by Hough in 1962 [33], its potential benefits to image processing have been highlighted by Rosenfeld 7 years later [61], and the modern form of the Hough transform as it is now employed was proposed by Duda and Hart in 1972 [17, 31]. The Hough transform is among the most popular of all existing shape extraction methods [34] and its success lies in its computationally efficient manner to achieve tem-



(a) Picture. (b) Hough space.

Figure 4.2: Principles of Hough transform.

plate matching. This technique is particularly effective to detect lines in pictures [5].

We point out two important assets of the Hough transform: (1) It allows imperfect instances of objects to be detected; in our case, it can identify lines with missing parts (e.g., dotted lines). Consequently, anomalies interrupted by network or process latencies and displayed as segmented lines are also detected. (2) It is robust against noise; it can detect anomalies surrounded by legitimate traffic that appear as noise on the analyzed pictures.

The Hough transform consists of a voting procedure, where each plotted point (x, y) of a picture elects lines that can pass through its position. It enumerates all ρ and θ solving the equation of a line in polar coordinates: $\rho = x \cdot \cos\theta + y \cdot \sin\theta$. All votes are collected in an array called a Hough space, and all candidate lines are determined as the maximum values in this array. We distinguish two ways to sum votes: all votes are equal so that the values of the Hough space increase linearly, or votes increase proportionally to the current accumulated values (exponential growth). In the former case, long and short lines are handled equally, whereas, in the latter case, the Hough transform privileges longer lines and avoids false detections.

The peaks in the Hough space are extracted with a threshold relative to the average value of accumulated votes. Naturally, in the case of a linear vote, the choice of threshold can be an involved task. We discuss the role of these parameters in section 4.3.3.

Figure 4.2 depicts an example of the Hough transform. The analyzed picture (Fig.4.2(a)) contains three plots, and the votes for each plot are represented by a curve in the Hough space (Fig.4.2(b)). The maximum number of votes in the Hough space is obviously at the intersection of the three curves $I = (\theta_0, \rho_0)$, identifying the line passing through the 3 plots, $\rho_0 = x \cdot cos(\theta_0) + y \cdot sin(\theta_0)$.

Identification (lines 10-19) For each line extracted by the Hough transform, the initial data are recovered from all plots involved. Packet information is summarized as a set of statistics called *events*. An *event* constitutes a report for a specific line in a picture. Anomalies are monitored by more than one line and cause several *events*. That is, *events* from the same address source or aimed at the same address destination are grouped together to form an *anomaly*. Since anomalies usually raise several *events*, single *events* are ignored to reduce the number of false-positive alarms. This heuristic is a trade-off between false-positive and false-negative alarms. It permits to avoid about 50% of false-

positive alarms, but decrease the number of true-positive alarms by about 20%.

4.3.2 Computational complexity

The computational complexity of our method is mainly the one of the Hough transform performed on all pictures. In our experiments, we implemented the standard Hough transform which have a computation complexity linear to the number of plots in picture. In the worst case, each plot represents a single packet, and the number of plots in a picture category is equal to the total number of packets N. Let f be the number of picture categories, p the number of pictures for each picture category, t the traffic duration divided by the time interval, and $n_{i,j,k}$ the number of plots in the picture k of category i at the time interval j. The cost of Algorithm 4.3.1 in the worst case is linear and specified as:

$$\sum_{i=1}^{f}\sum_{j=1}^{t}\sum_{k=1}^{p} O(n_{i,j,k}) = \sum_{i=1}^{f} O(N) = f \cdot O(N)$$

4.3.3 Parameter space

The performance of an anomaly detector strongly depends on the tuning of its parameters. In practice, satisfactory values are obtained by finding the best false-positive/false-negative trade-off through several tests run on well-known traffic traces. However, these values may not be suited for traffic with different properties. A relationship between parameter values and traffic characteristics is difficult to establish; thus selecting optimal parameters a priori is a challenge faced by every researchers. Automatic and dynamic tuning are still open problems.

This section pays close attention to the most significant parameters, namely the Hough transform parameters and the time interval, and evaluates their role in detecting anomalies in real Internet traffic.

We analyzed three sets of traces from the MAWI archive; two sets were collected from samplepoint-B (a 18-Mbps Committed Access Rate on a 100Mbps link) over the course of one week in 2004/08 and one week in 2005/08, and one set was collected from samplepoint-F (a full 100Mbps link) over the course of one week in 2006/08. The throughput at samplepoint-B was increasing during this period, and the data taken in 2004 and 2005 showed minor differences in volume. Moreover, samplepoint-B was replaced by samplepoint-F in July 2006, and this considerably increased the amount of data transmitted.

Simple heuristics helped us to evaluate the amount of anomalous traffic identified by our method. These heuristics were deduced from known attacks that occurred during the period of time analyzed and improper uses of TCP flags. Table 3.2 lists them in the same order as executed; the first five categorize traffic as “sure attacks”, and the last three categorize “suspected” traffic (meaning that either more inspection is needed, or it is a false-positive alarm). The quality of detection is measured as the ratio of “suspected” anomalies over the total number of anomalies reported (a lower ratio is better, see Fig. 4.3).

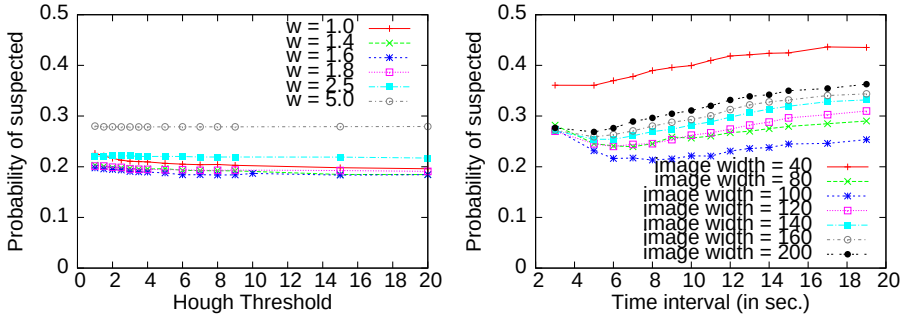


Figure 4.3: Evaluation of parameters with traces from 2004/08. For the left figure the image width is set to 100 and the time interval is set to 6 seconds. For the right figure the weight is set to 1.6 and the threshold is set to 10.

Hough parameters

During the voting procedure of the Hough transform, a vote for a line l is defined by a function of the form w^x, where x is the current number of votes for l, and w is a constant value named *weight*. A relative threshold is used to extract the detected lines in the Hough space.

The weight and threshold are the principal parameters of our method. To evaluate their impact on the anomaly detection, we executed our detection method on three data sets and changed the weights and threshold (other parameters were fixed). This analysis confirmed our expectations, that is: (1) Large weights ($w > 1$) help to highlight well-marked lines, whereas, $w = 1$ permits small lines to be elected. (2) The threshold is significant only when $w = 1$. Using the heuristics of Table 3.2 we deduced that the detection method performed better inspections on every trace analyzed with $w = 1.6$ (all thresholds tested led to similar results).

The left graph in Fig. 4.3 displays the average result for data during a week in August 2004. The two other data sets have provided similar results; hence, we concluded that this parameter is robust to throughput variances.

Image size and time interval

The manner of mapping traffic in a 2-D space is a key feature of our method; however, setting the proper resolution (pixel/second) of pictures is not intuitive.

Numerous tests on the three sets of traffic traces (the right graph of Fig. 4.3 shows the tests proceed on the traffic traces taken in 2004/08) indicated that the optimal image width for most cases is 100 pixels, whereas the ideal time interval depends on the analyzed traffic. The appropriate time interval for traces taken in 2004 is around 6-8 seconds (see right graph of Fig. 4.3). A smaller time interval (around 6 seconds) was found to be best for data recorded in 2005, whereas 3 seconds was found to be best for data from samplepoint-F. The main differences in the three sets of traces are their throughput and link bandwidth; in particular, the set collected in 2006 has more than twice the traffic volume of the one from 2005. Consequently, for the same time interval, pictures representing the traffic taken in 2006 might plot two times more points than those standing for the traffic from 2005 (depending on the traffic distribution).

The Hough transform works properly only if enough points are plotted in the pictures and the pictures are not saturated.

To maintain a certain quantity of data displayed in pictures, the time interval should be selected in accordance with the measured traffic rate of the observed traffic.

4.4 Evaluation

The evaluation of a detection method is an important step in validating its effectiveness; however, the lack of a common database with real backbone traffic and labeled anomalies raises a complicated issue. In Internet research community, the evaluation of an anomaly detection technique usually consists in one of the following processes: (1) Comparison of anomalies reported by a few different approaches [40]. (2) Analysis of real data and manual estimation of the number of false-positives reported [9, 45, 16]. (3) Injection of malicious traffic into traces supposed to be anomaly-free and computation of false-positive/false-negative rates [45].

We used the processes (1) and (2) to evaluate our detection method in realistic conditions. In section 4.4.1 we identify anomalies in a large data set and carefully inspect the results. In section 4.4.2 we compare the anomalies detected by our method with those identified by a method based on gamma modeling and a method based on K-means.

4.4.1 Anomalies of MAWI database for 6 years

We analyzed all traces of the MAWI database collected at samplepoint-B from 01/2001 to 06/2006; each trace represents 15 minutes of traffic with anonymized IP addresses. The same data set has been dissected in [12], to show the detailed evolution of the traffic as well as an application breakdown. Although [12] did not aim at labeling anomalies in MAWI systematically, it does mention several prominent anomalies that significantly altered the traffic. For example, a major ping flood occurred on 2003/08-12, and outbreaks of the Sasser worm were identified in 2004/08, 2004/12 and 2005/03.

Results

We used our method to analyze all traces collected from 2001 to 2006. The traces were processed with same parameters (weight=1.6, time interval=8 seconds, image width=100 pixels and threshold=10). Since the weight was set to 1.6 the threshold has been arbitrary chosen (see Section 4.3.3). Figure 4.4 summarizes the results and classifies them by the heuristics in Table 3.2. This graph plots the number of anomalies, whereby an anomaly is as described in Section 4.3.1 (namely as a set of grouped events with respect to their sources or destinations IP).

The large anomalies noticed in [12] can be observed in Fig. 4.4; the ping flood appears from 2003/08 to 2004/01, and the three Sasser outbreaks are represented as three peaks between 2004/05 and 2005/06. Our method also identified important activity on port 135 starting in 2003/08 and lasting several years (labeled RPC Fig. 4.4). This traffic also appears in the application breakdown of [12], and it has been attributed to MS vulnerabilities. Our manual

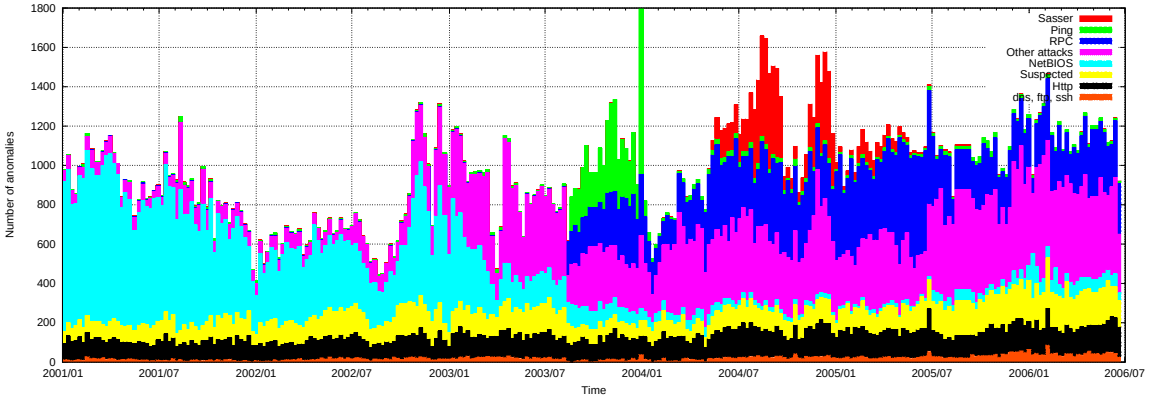


Figure 4.4: Average number of anomalies per week reported by our method on all traffic traces collected on the MAWI samplepoint B from 01/01/2001 to 30/06/2006.

inspection revealed that this anomalous traffic was initiated by a large outbreak of the Blaster worm (also known as MSBlast/Lovsan) spreading through an exploit in the Remote Procedure Call (RPC) protocol of almost all versions of Windows at this time. Security holes in RPC have been frequently reported since then, and this protocol is still a common medium for various attacks.

Mainly NetBIOS traffic was reported from January 2001 to August 2002. We deduced from our manual verification that most flows contained a tiny number of packets with both the port source and destination set to 137/udp. This traffic is a manifestation of the normal behavior of the name resolution service implemented in the Windows networking shares (even though this mechanism is designed for local networks). We concluded that the traffic was principally failed name resolution requests initiated by a large number of distinct hosts and aimed at numerous destinations. We noticed that most of the sources and destinations of the identified flows did not have other network activity, and their bandwidth consumption was really low. In addition, the average number of packets observed in the analyzed backbone link steadily increased during the six years. Our detection method identified this category of traffic in 2001 and 2002 because of the fixed parameters it employed for the analysis. This means that not enough points were displayed in the pictures to compute the Hough transform properly. Although malicious behavior is not evident, these anomalies still reflect a misuse of the NetBIOS protocol.

However, from 2002/10 onwards, the distribution of NetBIOS traffic completely changed and clearly indicated malicious behavior. Indeed, we observed that various hosts were probing entire sub-networks to take advantage of the security flaws of the Windows file sharing mechanism, and several viruses were released during the same period (e.g., Opaserv, Bugbear).

Other attacks were mainly related to the NetBIOS protocol, but the heuristics classified these as due to a high rate of SYN flags (on port 139/tcp).

This analysis of the MAWI database exposed large-scale attacks, and it demonstrates our method's ability to identify numerous anomalies. However, quantitative observations conducted over a long period (for 6 years) naturally

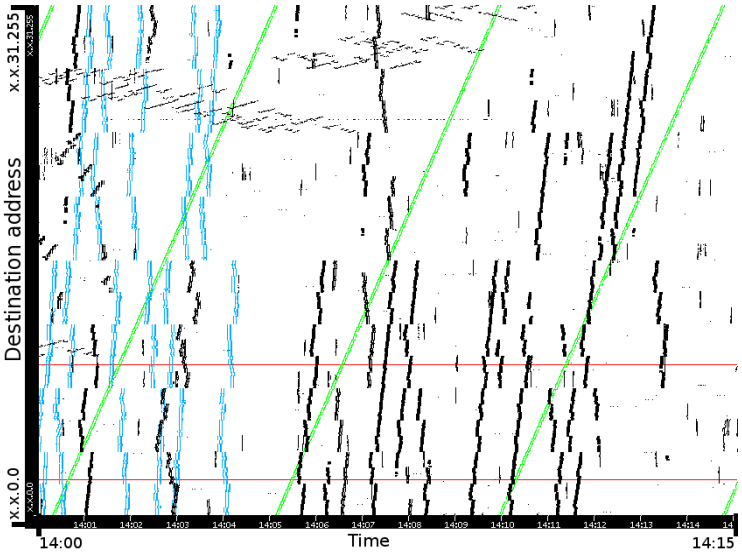


Figure 4.5: Several examples of anomalies detected in one traffic trace of 15 minutes (2004/10/14). Two horizontal lines: 8000/udp (iRDMI). On the left light-colored: 5900/tcp (VNC). Three long slanted lines: 445/tcp (MS Service). Black: 1023/tcp, 5554/tcp and 9898/tcp (Sasser).

omit occasional anomalies. Hence, the next section discusses anomalies detected in a single day.

Examples of anomalies detected in the same day

Figure 4.5 illustrates several examples of anomalies detected in the same day; legitimate traffic and other identified anomalies have been excluded for clarity.

The light-colored lines on the left side of Fig. 4.5 are generated by one host probing a large sub-network on port 5900 (VNC, a remote control application). The attack is aimed at 16^2 hosts of the same sub-network, but due to the routing policy, only half of them have been contacted via the analyzed link. Despite missing packets, the anomaly is still easily identifiable. The activity was initiated by only one source IP address, so the detection method reports it as a single anomaly.

The three long slanted lines stand for a similar behavior against a Windows service (port 445), whereas the two horizontal lines display abnormally high traffic between a couple of hosts on port 8000. These two long-lasting anomalies started before and stopped after the detection process, meaning that they could not be revealed by methods analyzing traffic volume. Our method had no difficulty in identifying them.

The traffic on this day is flanked by two significant outbreaks of the Sasser worm. Sasser activity is shown in black in Fig. 4.5, and two different propagations of the worm are shown. On the one hand, long vertical lines, depicting a large and quick spread, appear on the whole picture. On the other hand, the small slanted lines at the top of the figure show a slowly spreading worm. These two observations illustrate either two variants of the worm or the net-

work/process latency effect on the worm spread.

4.4.2 Cross-validation

We compared results of our method with those of two other methods. One consists of random projection techniques (sketches) and multiresolution gamma modeling [16]. Hereafter we call it as the gamma-based method. The traffic is split into sketches and modeled using Gamma laws, and anomalous traffic is reported by using the statistical distance from the average behavior. The other method is a distance-based outlier detection method using K-means [63]. The traffic is clustered with K-means regarding 14 traffic features and outliers are reported depending on their density and distance to other clusters.

Methodology

The three methods were tested on several trans-Pacific traces captured during August 2004. A great deal of anomalous network activity concerning the Sasser worm was reported during this time. Analysis of each data set leads to similar conclusions, so we only present the results for one traffic trace (2004/08/01). We tuned all methods until they report approximately the same number of alarms. The alarms are reported differently by these methods, so we checked whether an alarm reported by one method had also been detected by the others, and vice-versa.

Results

The gamma-based method was executed with the values of 0.8 for the alpha parameter and 500 for the threshold and it reported 1083 alarms. K-means was computed with 100 clusters and it reported 917 alarms. Our method was run with a time interval of 10 s, $w = 1.6$ and it reported 1063 alarms. For a 15-minutes trace with a mean throughput of 20.77Mbps, 6 591 957 packets, and 614 324 different IP addresses (57 862 source addresses), the execution time of our method was about 3.5 minutes on a standard desktop PC (Core2Duo 2.6 GHz, 2 GB of RAM). Table 4.1 shows these alarms classified by using the heuristics of Table 3.2.

We checked if the alarms reported by our algorithm had also been reported by the gamma-based method. We inspected all alarms not reported by either method and noticed that the 574 $(854 - 280)$ alarms labeled as *ATTACK* were true-positive alarms related to worms (Sasser and Blaster) or scan activity (mainly on NetBIOS). Our method detected twice as much anomalous traffic for this class of anomaly than the statistical one did. Several of these anomalies could not be detected with the gamma-based method because of the small number of packets involved (< 500 packets). However, the 24 $(130 - 106)$ and 27 $(79 - 52)$ alarms labeled as *SPECIAL* and *UNKNOWN* reported by our method but not by the gamma-based one were heavy traffic between two hosts using HTTP, HTTPS, or peer-to-peer protocols. Although the traffic in most of these cases seemed to be harmless elephants, their packet payloads would have to be checked to conclude if they were indeed false-positives alarms.

The gamma-based method reported 1083 alarms; 579 $(1083 - 438)$ of these were not detected by our method. Of these 579 alarms, 375 were labeled as

Table 4.1: Alarms reported by the Hough-transform-based (HT), gamma-based (G), and K-means-based (KM) methods.

	HT	G	KM	HT&G	HT&KM	G&KM
Attack	854	323	306	280	75	50
Special	130	517	488	106	23	75
Unknown	79	243	123	52	49	26
Total	1063	1083	917	438	147	151

SPECIAL, and 161 were classified as *UNKNOWN*. We deduced from a manual inspection that most of them were heavy traffic with uncommon properties using http or peer-to-peer protocols; we were not able to determine if they were false-positive alarms without payload. However, our method missed 43 $(323 - 280)$ events reported by the gamma-based method and labeled as *ATTACK*; 21 of them represents worms (mainly Sasser) and 11 stand for PING flooding.

The K-means-based method identified 917 alarms; 770 $(917 - 147)$ of these were not detected by our method. 439 of these 770 were labeled as *SPECIAL*, and 100 were classified as *UNKNOWN*. Manual inspection has shown that they were mainly harmless traffic with uncommon properties. Only 75 alarms labeled as ATTACK were reported by both the K-means-based and our method. The 231 $(306 - 75)$ alarms labeled as ATTACK only reported by the K-means-based method are mainly flows with a high percentage of TCP flags set to SYN, FIN or RST. Although these events are mainly true-positive alarms missed by our method, we had difficulty in determining the threat posed by 116 of them where the number of packets send by a suspicious host is really low (≤ 10).

In order to validate the sufficiency of the heuristics of Table 3.2, we inspected the 445 $(79 + 243 + 123)$ alarms labeled as *UNKNOWN* reported by the three methods. 411 are considered as peer-to-peer traffic because using both higher ports. The rest of them are usual traffic, RSYNC (10), NNTP (6), POP3 (5), RTP (4), etc.

Discussion

The proposed method has reported a large number of alarms labeled as *ATTACK* not detected by other methods, indicating that our method has a high probability of reporting true-positive alarms compare to others. However, our method still missed 249 $(231+43-25$(double counted)$)$ *ATTACK* alarms (false-negative) because it does not take TCP flag into account and due to the absence of port number in ICMP protocol. Considering the 116 suspicious *ATTACK* alarms reported by K-means (i.e., host sending less than 10 packets), the detection ratio (true-positive rate) of our method is about $77 \sim 87\%$.

Many alarms labeled as *UNKNOWN* and *SPECIAL* have been reported by the gamma-based and K-means-based methods. Although these alarms could be true positives misclassified by the heuristics, our manual inspection revealed that they were false-positives alarms. Also, our method reported only 209 $(130 + 79)$ false-positive alarms over the 56759 benign source IP (reported by none of the three detection methods as ATTACK). These observations show the low false-

positive rate (0.3%) of the proposed method.

Furthermore, we have manually observed 426 source addresses related to the Sasser activity, 84 (19%) have been identified by the K-means-based method, 156 (36%) by the method based on gamma modeling, and 321 (75%) by our method.

We deduced from Table 4.1 that even though our method and the gamma-based one are quite different, they had almost 50% of their results in common. Our method detected two times more traffic related to worms and scan activity than the gamma-based method did. This category of anomaly is characterized by small flows and its reflects the fundamental weakness of statistical methods. By analyzing TCP flags, the K-means-based method could detect several anomalous traffics not reported by other methods. However, this method is designed to identify outliers and since the Sasser activity has been dominating analyzed traffic, it failed in detecting such traffic and reported many false-positive alarms. Also, the K-means-based method does not scale to backbone traffic because of its computation time. The three methods have distinct weaknesses and advantages; hence, they would be a good combination.

4.5 Summary

We illustrated the characteristic shapes of anomalous traffic in time and space and presented an approach to anomaly detection based on pattern recognition. Since the proposed approach employs a pattern of anomalous traffic it identifies anomalies although they dominates the traffic or they stand for mice flows, thereby, the proposed method overcomes the shortcomings of current statistical-based detectors. Furthermore, this method takes advantage of a graphical representation to reduce the dimensions of network traffic and provide intuitive output. Only header information is required; no inspection of the packet payload and no prior information about the traffic or port numbers are needed. We conducted a detailed evaluation of our method by analyzing the principal parameters and by validating it on actual Internet traffic. The analysis of traffic from a trans-Pacific link revealed that our method can identify various anomalies (e.g., worms and network/port scans), and mice anomalous flows.

The comparison of our method with a gamma-based method and a K-means-based method indicates that the three approaches identified distinct classes of anomalies. Therefore, their use in combination would have a synergistic effect.

Despite its benefits the proposed method has two drawbacks preventing practical usages: first, maintaining the parameter set optimally tuned requires constant adjustments of the parameters according to the traffic throughput (especially the time interval). Second, the proposed anomaly detector is designed to analyze only packet headers whereas, in practice, Internet traffic is usually monitored in the form of flows. The following Chapter addresses these drawbacks and proposes an adaptive anomaly detection method that is automatically maintaining the parameter set optimally tuned.

Chapter 5

Automated Tuning of the Hough-based Anomaly Detector

5.1 Introduction

Detecting anomalies in the Internet traffic has been mainly addressed as a statistical problem. Although intensive studies have been carried out in this field, the proposed anomaly detectors still have a common drawback [60, 32] that prevent their use in real environments; selecting the optimal parameter set. This drawback is mainly due to the difficulty of understanding the relations between the Internet traffic and the statistical tool underlying the anomaly detectors.

Only a few works have investigated this drawback currently discrediting anomaly detectors. A careful study of the detectors based on principal component analysis (PCA) was carried out by Ringberg et al. [60], and they identified four main challenges including the sensitivity to analyzed traffic and parameter tuning. In addition, an attempt to automatically tune a method based on gamma modeling and sketches was conducted by Himura et al. [32]. They designed a learning process for predicting the optimal parameters regarding the best parameters for past data. However, this method suffers from a high error rate as unexpected events do appear.

In Chapter 4 we proposed a pattern recognition based anomaly detector. The main idea of this detection method is to monitor the traffic in 2D pictures where anomalies appear as "lines", which are easily identifiable using a pattern recognition technique called the Hough transform [17]. This method overcomes several shortcomings of current statistical-based anomaly detectors, however, similarly to the statistical-based detectors it requires constant attention from network operators to be optimally used. Indeed, the optimal values of the parameters fluctuate along with the traffic throughput variations and require continuous adjustments making it unpractical for real usage (Section 4.3.3).

In order to provide a detector that is easily tunable and robust to traffic variations, this chapter follows a similar approach to the one presented in Chapter 4, however, the design of the 2D pictures monitoring the traffic is fundamen-

Table 5.1: Different kinds of common anomalies and their particular traffic feature distributions.

Anomaly	Traffic feature distribution
Port scan	Traffic distributed in destination port space and concentrated on single destination host.
Network scan, Worm, Exploit	Traffic distributed in destination address space and concentrated on limited number of destination ports.
DDoS, Netbot, Flash crowd	Traffic distributed in source address space and concentrated on limited number of destination addresses.

tally different and allows to better highlight anomalies. This new design also ease the use of the proposed detector in practice as it enables it to analyze flow reports (contrarily to the detection method proposed in Chapter 4 that analyzes only packet traces). Moreover, the main contribution of this work is to obtain a complete understanding of the proposed method parameter set and provide a mechanism that automatically tunes it based on the traffic variations. The advantages of this adaptive method are demonstrated by comparing its results to those obtained using fixed parameter tunings and those of three other anomaly detectors using four years of real Internet traffic. The results highlight the superiority of the proposed method in terms of the true positive and false positive rates, emphasizing that automatically adjusting the parameter set in regards to the traffic fluctuations is crucial for continuously performing an accurate level of detection. Inspecting the false negative rate of the proposed method allows us to describe the particular class of anomaly that is inherently missed by the proposed detector. Thus, the shortcomings of the proposed detector are well-defined and complementary detectors are suggested.

5.2 Abnormal distribution of traffic features

Recent works have identified anomalous traffic as alterations in the distributions of the traffic features [45, 23, 13, 72, 16]. For example, Table 5.1 lists several kinds of anomalies commonly identified in Internet traffic. Each kind of anomaly inherently affects the distribution of two traffic features. Similarly, in this chapter an anomaly refers to a set of flows altering the distribution of at least one of the four following traffic features: the source IP address, destination IP address, source port, and destination port. However, the proposed approach for observing these alterations in the traffic feature distributions is substantially different from that in other works. Previously, anomalies have been mainly detected by identifying the outliers in the aggregated traffic using different formalisms — e.g., signals [45], histograms [16, 13], or matrices [68] — whereas, the proposed method identifies particular patterns in pictures. The analyzed pictures are two-dimensional scatter plots, where each axis represents a traffic feature, each plot stands for traffic flows, and the particular traffic feature distributions of

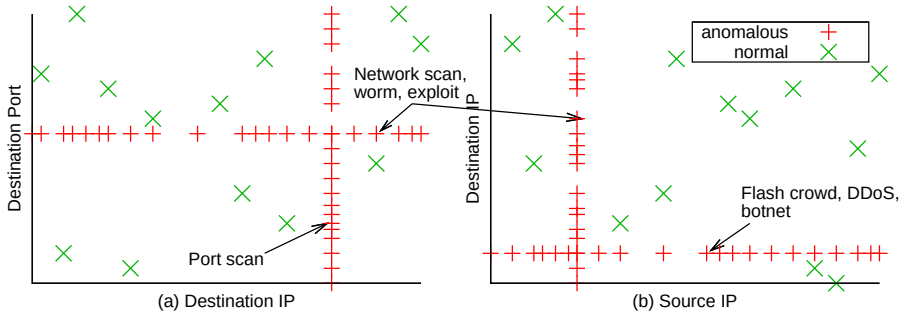


Figure 5.1: Example of two pictures highlighting anomalous traffic as lines.

the anomalies are easily identifiable as lines.

Figure 5.1 shows two examples of the pictures analyzed by the proposed method. Figure 5.1a displays traffic with regard to its destination port and destination address. This graphical representation of the traffic makes it easy to discriminate the port scan, network scan, worm, and exploit from the benign traffic as they appear as lines in the picture (Fig. 5.1a). Figure 5.1b, however, displays the traffic in regard to its destination and source addresses, and permits other kinds of anomaly to be observed. For instance, distributed denial of service (DDoS), flash crowd and botnet activities appear as horizontal lines in this scatter plot.

The three main advantages of this approach over the previous works are [60]: (1) the anomalous flows are inherently pinpointed in the scatter plots, whereas the identification of the anomalous flows detected in a signal requires additional extraction mechanisms [13, 66]. (2) The proposed approach is able to monitor the pattern of a large-scale anomaly whereas the methods detecting anomalous traffic as outliers fail if a majority of the traffic is contaminated by anomalies (e.g., outbreak of virus). (3) In regard to the traffic features monitored by the pictures and the direction of the identified line, one can easily deduce the kind of observed anomaly.

5.3 Anomaly detection method

The anomaly detection method proposed here consists of five main steps:

1. The traffic of the current time interval is mapped onto five different pictures.

2. The Hough transform is computed on each picture to uncover the plot distributions.

3. Abnormal plot distributions are detected in the Hough spaces.

4. Traffic information corresponding to the anomalous plots are retrieved and reported.

5. The time interval is shifted and step 1 is repeated.

5.3.1 Pictures computation

The proposed approach takes advantage of several kinds of pictures to monitor the different aspects of the traffic and highlight the different kinds of anomalies. The analyzed pictures are 2-D scatter plots designed from four traffic features: {source IP address, destination IP address, source port, destination port}. For the remainder of this chapter the term *traffic features* will refer to only these four traffic features. The five picture categories correspond to all the possible pairs of traffic features containing IP address. Namely, the x and y axis of the picture, respectively, correspond to the following pairs of features:

- Source IP address, destination IP address
- Source IP address, source port
- Source IP address, destination port
- Destination IP address, source port
- Destination IP address, destination port

A flow in the analyzed pictures is represented by a plot that is located using the two following mechanisms. (1) The port space is shrunk to the size of the pictures: Let assume a 1000-pixel picture ($ySize = 1000$) that has a y axis standing for the source port, then a http flow, i.e., $SrcPort = 80$, is plotted at $y = \lfloor SrcPort * ySize/2^{16} \rfloor = \lfloor 80 * 1000/65535 \rfloor = 1$, and each pixel of the picture represents approximately $\lfloor 65535/1000 \rfloor = 65$ distinct port numbers. (2) The IP address space is at first hashed by ignoring the first h bits of the addresses and then shrunk to the size of the picture. For example, supposing $h = 16$ and a 1000 pixel wide picture ($xSize = 1000$) with an x axis as the source IP, then a flow from the source IP 192.168.10.10 is plotted at $x = \lfloor (SrcIP \ mod \ 2^{32-h}) * xSize/2^{32-h} \rfloor = \lfloor (192.168.10.10 \ mod \ 2^{16}) * 1000/2^{16} \rfloor = \lfloor (0.0.10.10) * 1000/2^{16} \rfloor = 39$. Notice that we only deals with square pictures, meaning that the $xSize = ySize$.

5.3.2 Hough transform

A well-known image processing technique called the Hough transform [17, 27] helps us in extracting the relevant information from computed pictures. The Hough transform is commonly used to detect the parametric structures (e.g., line, circle, or ellipse) in pictures and has the advantage of being robust to noise and able to detect incomplete shapes.

The basic usage of the Hough transform allows for the identification of lines in a picture. It consists of a voting procedure, where each plot of the picture votes for the lines it belongs to. Formally, each plot in the picture $p = (x_p, y_p)$ votes for all the θ and ρ that satisfy $\rho = x_p \cdot cos(\theta) + y_p \cdot sin(\theta)$ (line equation in polar coordinates). All the votes are stored in a two-dimensional array, called the Hough space, in which one dimension stands for θ and one for ρ.

In order to find the local maxima in the Hough space, thus the prominent lines in the picture, a robust peak detection based on the standard deviation σ of the Hough space is implemented. Therefore, all flows corresponding to the elements of the Hough space that are higher than 3σ are reported as anomalous.

5.3.3 Complexity

The computational complexity of the proposed method is mainly one of the Hough transforms that is linear to the number of plots in picture. In a worst case scenario, each plot represents a single flow so the number of plots in the pictures is equal to the total number of flows N. Let $f = 5$ be the number of picture categories, t the traffic duration divided by the time interval, and $n_{i,j,k}$ the number of plots in the picture k of category i at the time interval j. The cost of the proposed algorithm in the worst case is linear to N:

$$\sum_{i=1}^{f}\sum_{j=1}^{t} O(n_{i,j}) = \sum_{i=1}^{5} O(N) = O(N)$$

In our experiments, the proposed method takes about one minute to analyze a 15-minute traffic trace from the MAWI archive.

5.4 Data and processing

This chapter particularly focuses on two data sets from the MAWI archive (see Section 3.1 for more details on MAWI); (1) the first week of August 2004 was particularly affected by the Sasser worm [12, 23] and provides valuable support for illustrating the benefits of the proposed method. (2) All the traffic recorded from 2003 to 2006 allowed us to evaluate the global performance of the proposed method by comparing its results to the ones of other anomaly detectors.

Due to the lack of ground truth data for backbone traffic, the evaluation of the proposed detector relies on heuristics that is fundamentally independent from the principle of the proposed method (Table 3.2). Indeed, these heuristics is based on well-known port numbers and abnormal usages of TCP flags [12, 23], whereas the proposed method uses only the port numbers as indexes and does not rely on the application information related to them nor the TCP flags. Heuristics classifies traffic into two categories, *attack* and *special*, and helps in quantifying the effectiveness of the detection method.

An anomaly detector is expected to report more traffic classified as attacks than those labeled special. Thus, the *accuracy* of a detector is defined as the ratio of the alarms classified as attacks by the heuristics listed in Table 3.2.

5.5 Parameter tuning and drawbacks

5.5.1 Experimental parameter tuning

The following experiments aim at finding the optimal parameter tuning of the proposed method using one week of traffic affected by the Sasser worm (Section 5.4). Furthermore, these experiments uncover the correlation between the two main parameters, i.e., the size of picture and the time interval, and show that the performances of the proposed method are not affected by any variance in the h value as long as the number of possible indexes is higher than the picture size, $2^{32-h} > xSize$.

Figure 5.2 depicts the average accuracy of the detection method using numerous parameter values. It highlights that the proposed method is able to

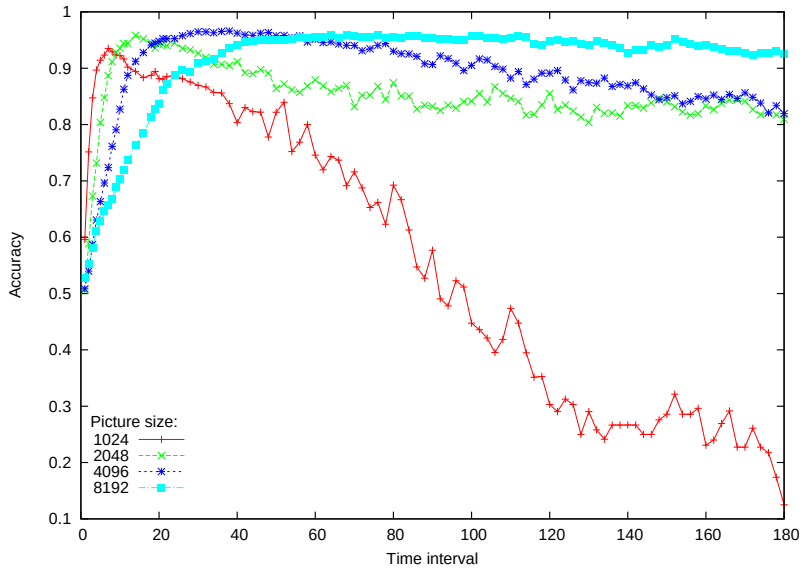


Figure 5.2: Accuracy of proposed method using four different picture sizes.

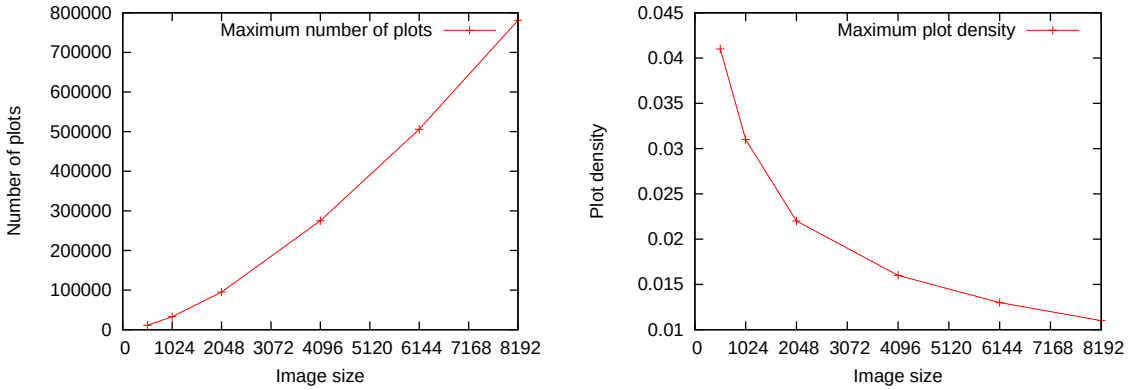
achieve an accuracy that is higher than 0.9 for any time interval $> 4s$ and a suitable picture size. Furthermore, Fig. 5.2 indicates that the optimal picture size is proportional to the size of the time interval. For instance, if the time interval is less than $8s$ the best performance is obtained with a picture size set to 1024, whereas the time interval ranges $(9, 16)$ are suitable for a picture size equal to 2048, and so forth. Intuitively, a larger time interval involves a greater number of plots in the pictures; thus, to avoid meaningless saturated pictures, the optimal size of a picture increases along with the size of the time interval.

Although the specific values given here are suitable for the analyzed traffic, different values might be more effective for traffic having different properties. Obviously, traffic with the same properties but a higher throughput displays more plots in the pictures, and thus in this case, smaller time intervals are required to maintain an acceptable number of plots in the pictures.

5.5.2 Evaluation of optimal parameter

The time interval is the parameter that controls the amount of traffic displayed in the pictures. Thus, as the proposed method inherently translates the traffic flows to the plots in the pictures, the time interval allows us to select the quantity of plots appearing in the pictures. The challenge in setting the time interval is the trade-off between displaying enough plots to have relevant pictures and limiting the surrounding noise representing the legitimate traffic and hiding anomalies.

The sensitivity of the implemented Hough transform to the number of plots in the pictures is analyzed using synthetic pictures that have a random line and various amounts of uniformly distributed noise. The algorithm was performed 100 times on different pictures with the same level of noise. If the 100 tests are successful then the noise is increased and the algorithm is again performed.



(a) Maximum acceptable number of plots

(b) Maximum acceptable plot density

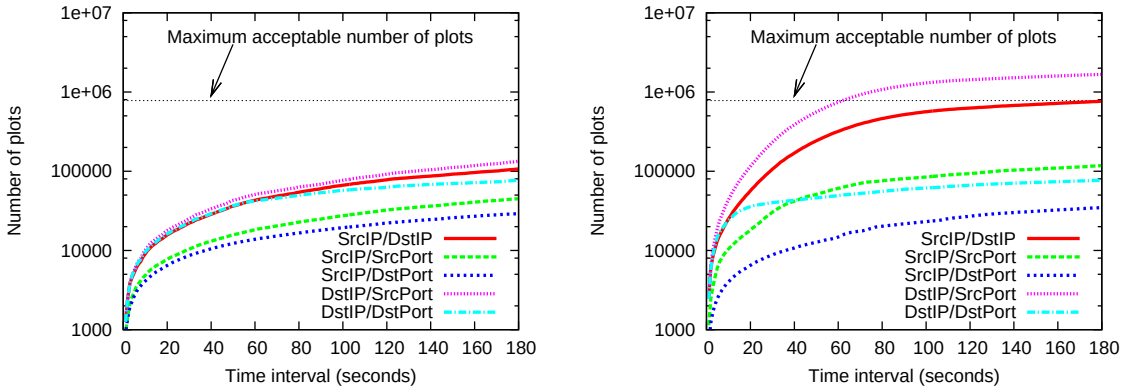
Figure 5.3: Evaluation of maximum acceptable number of plots to perform the Hough transform. The plot density is the maximum acceptable number of plots over the picture area.

The highest noise level for which all 100 executions of the algorithm succeed defines the maximum acceptable number of plots in a picture. This experiment was conducted using six different picture sizes, as indicated in Fig. 5.3(a). As expected, the maximum acceptable number of plots in the pictures increases with the picture size. Figure 5.3(a) shows that the maximum acceptable number of plots for picture sizes of 1024, 2048, 4096, and 8192 are respectively 33000, 95000, 275000, and 781000. Figure 5.3(b) shows that this increase is not linear to the area of the picture and the common upper bound for all the considered picture sizes is approximately 1% of the picture area.

5.5.3 Dispersion of plots in pictures

The previous section provided an insight on how to select the suitable time interval for a particular picture, but the proposed method analyzes five different pictures at the same time. A crucial task is to understand the divergence between the different kinds of pictures. Since the five picture categories monitor distinct feature spaces, plots corresponding to the same traffic are differently dispersed in all the pictures. Therefore, the traffic is usually depicted by using a different number of plots for two pictures from different categories. For example, Fig. 5.4(a) shows the number of plots for the five kinds of pictures for several time interval sizes. This figure highlights that the number of plots appearing in each picture category increases at different rates. A slow increase in the number of plots means that many flows share the same instance in the monitored feature spaces, whereas a rapid growth highlights the flows spreading into the observed feature spaces. The rate of increase of the plots for each picture category is strongly related to the throughput and the dispersion of the traffic in the feature space.

Since anomalies alter the traffic feature distribution, they also significantly affect the increase in the number of plots. Figure 5.4(b) is a typical example where the increase in plots for certain picture categories is rapidly increasing



(a) Traffic with few Sasser activities

(b) Traffic with many Sasser activities

Figure 5.4: Plot growth for different picture categories ($xSize = 8192$).

due to anomalous traffic. Indeed, the traffic analyzed in Fig. 5.4(b) contains an outbreak of the Sasser worm highlighting a considerable increase in the number of plots for two picture categories monitoring the destination address. This observation is in accord with the behavior of the Sasser worm manually observed in the traffic trace, that is, the worm tries to infect numerous remote hosts to spread throughout the network.

Despite their differences, the two traffic analyzed in Fig. 5.4 are taken from the same traffic trace (Fig. 5.4(b) representing the first three minutes of the traffic trace, whereas Fig. 5.4(a) is the traffic recorded three minutes later), illustrating two drawbacks of the proposed method. (1) For the same traffic, the number of plots in all the picture categories is significantly different. Thus, the suitable time interval for a picture from a certain category does not necessarily suit the pictures from the other categories. (2) The increase in plots for a certain picture category sharply varies especially when anomalous traffic appears. Thus, the suitable time interval for a single picture category fluctuates over time.

5.6 Adaptive time interval

Here, an improved version of the anomaly detection method is proposed to overcome the drawback identified in the previous section. This new version assigns different time intervals to all the picture categories and adapts these time intervals to the traffic variation. Therefore, the value of the time intervals is no longer a fixed value taken as an input, but it is automatically computed by taking into account the throughput and the traffic distribution in the traffic feature spaces.

The proposed improvement consists of controlling the amount of monitored traffic based on the quantity of plots in the picture instead of the time interval. The Hough transform is performed only if a certain number of plots p are displayed in the picture (regardless of the time interval corresponding to the traffic mapped into the picture), and other pictures keep monitoring the traffic until they display a sufficient number of plots, p. Therefore, all the pictures stand

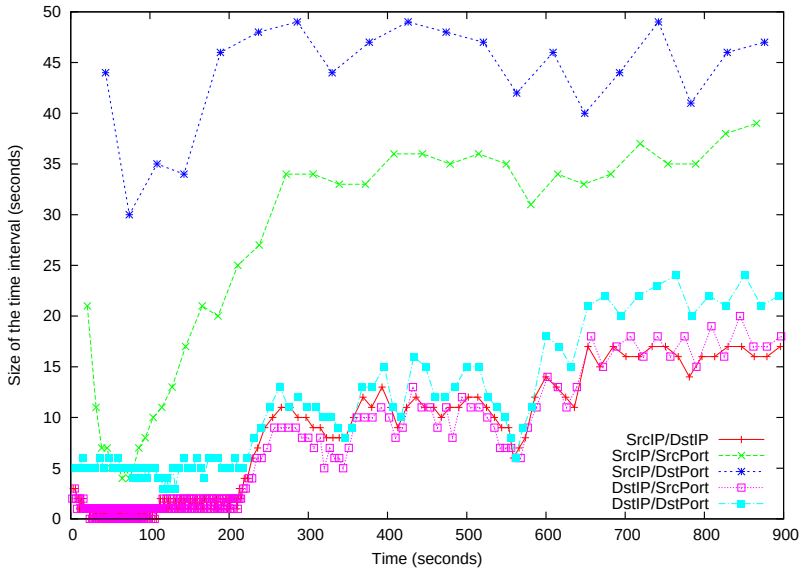


Figure 5.5: Evolution of time interval corresponding to pictures computed during 15 minutes of traffic.

for different time intervals and the Hough transform is performed at different instants of time for each picture. The first two steps of the algorithm proposed in Section 5.3 are replaced by: (1) Map traffic to pictures until a picture displays p plots. (2) Compute the Hough transform for pictures with p plots. In addition, the time interval parameter is replaced by p, which is the number of plots required to perform the Hough transform. The value of p is directly deduced from the picture size to assure the success of the Hough transform. The upper bound for p is 1% of the picture area (Section 5.5.2), and the lower values help in quickly reporting the anomalies since the Hough transform is performed earlier. However, too small p values result in irrelevant pictures as the sample traffic displayed in pictures is insignificant. In the following experiments, p is arbitrarily set to 0.5% of the picture area, $p = 0.05 \cdot xSize^2$. Hereafter, this new version of the detection method is referred to as the adaptive method.

5.6.1 Performance improvement

The benefit of the adaptive method is evaluated by using one week of traffic (Section 5.4). For clarity reasons and because all the traffic traces reach a similar conclusion, the following focuses only on the first day of the analyzed traffic.

Robustness to traffic variation

Figure 5.5 displays the time intervals corresponding to all the pictures computed during the analysis of the 15 minutes of traffic. The first four minutes of this traffic are significantly affected by the Sasser worm resulting in a higher throughput and an increase in the number of destination addresses. Nevertheless, the

method successfully handled the traffic variation, that is, the time intervals represented by the pictures monitoring the destination address remain from 1 to 5 seconds during the Sasser outbreak (Fig. 5.5). However, the same quantities range from 14 to 25 seconds during the last four minutes of traffic, where the traffic is much less polluted by the Sasser worm. This example illustrates the benefit of the adaptive method since selecting a fixed value for the time interval of the basic method is challenging.

Accuracy gain

The only parameter of the adaptive method is the picture size, and by setting it to three different values, namely 1024, 2048, and 4096, the same high accuracy score is observed, 0.99, 0.98, and 0.99, respectively. However, the number of reported alarms decreases as the picture size increases, which is 373, 173, and 117 events respectively. Thus, for the following experiments the picture size is set to 1024 in order to report as much anomalous traffic as possible.

The comparison between the two versions of the method emphasizes the better false positive and true positive rates of the adaptive method. Namely, it identifies 369 source addresses infected by Sasser (i.e. 86% of the Sasser traffic manually identified). However, the basic method, with identical parameters but a fixed time interval of 10 seconds, identifies only 258 source addresses related to Sasser (i.e. 60% of the Sasser traffic manually identified). The basic version of the method is able to identify the same amount of Sasser traffic only if the time interval is set to one second, however, in this case 229 http traffics were also reported and a manual inspection revealed that they are benign traffic regarded as false positive alarms.

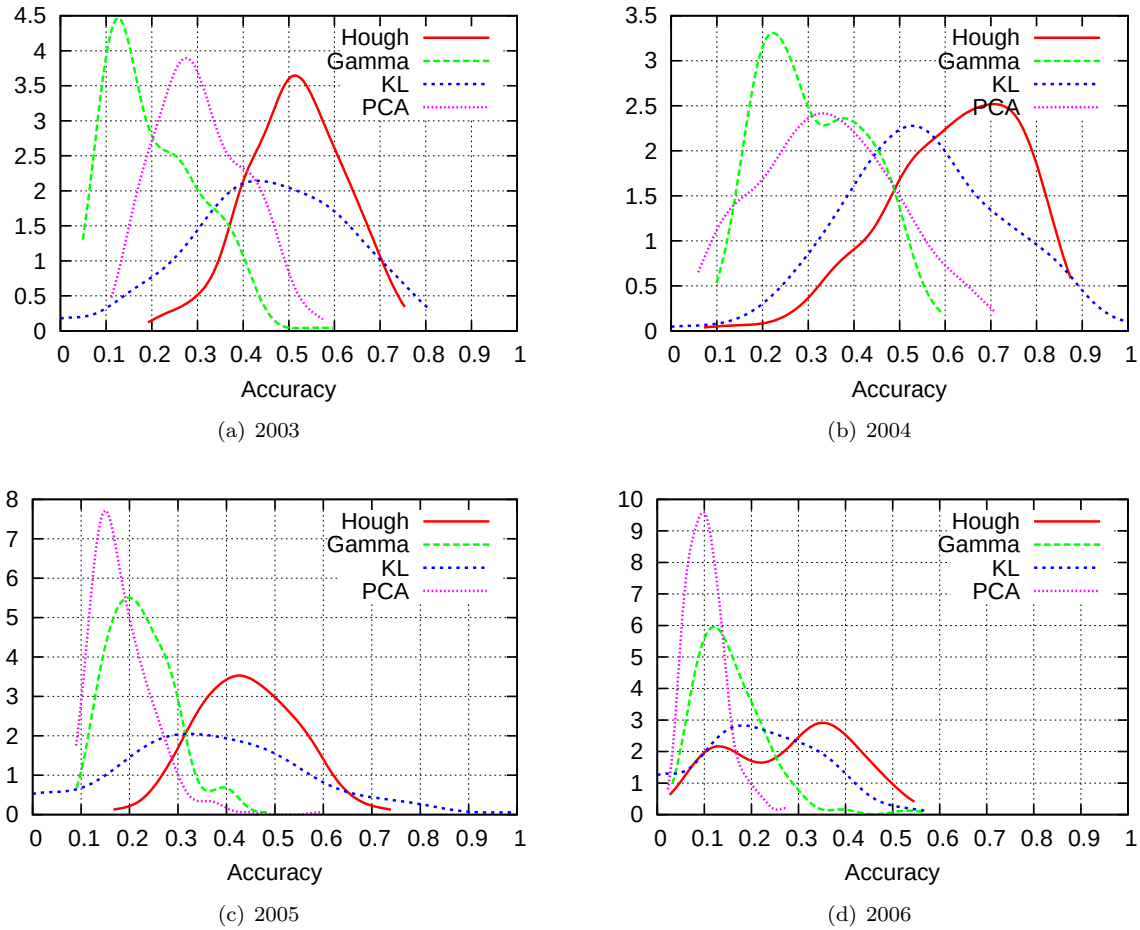
5.7 Evaluation

The adaptive detection method is evaluated by analyzing four years of MAWI traffic (i.e. 2003, 2004, 2005, and 2006) and comparing its results to the outputs of three other anomaly detectors based on different theoretical backgrounds.

5.7.1 Compared detectors

For performance comparison we select three detection methods that are, similarly to the proposed method, analyzing only packet header and aim at finding nonspecific classes of anomaly. These three compared detectors are (1) the well-known PCA-based detector [45] (in this work the implementation of this detector relies on sketches to analyze traffic taken from a single link [37]), (2) the detection method based on multi-scale gamma modeling and sketches [16], and (3) the detector based on the Kullback-Leibler (KL) divergence and association rule mining [13] (see Section 3.3 for more details on these three detectors). The picture size parameter of the adaptive method is set to 1024, whereas, the parameters of the three other methods are set with fixed and arbitrary values that are globally suitable for the analyzed MAWI traffic.

The four detectors aim at finding any kinds of traffic anomaly by inspecting only IP header. However, they aggregate traffic using different formalisms, i.e., the proposed method monitor the traffic using pictures whereas the PCA-based



(a) 2003

(b) 2004

(c) 2005

(d) 2006

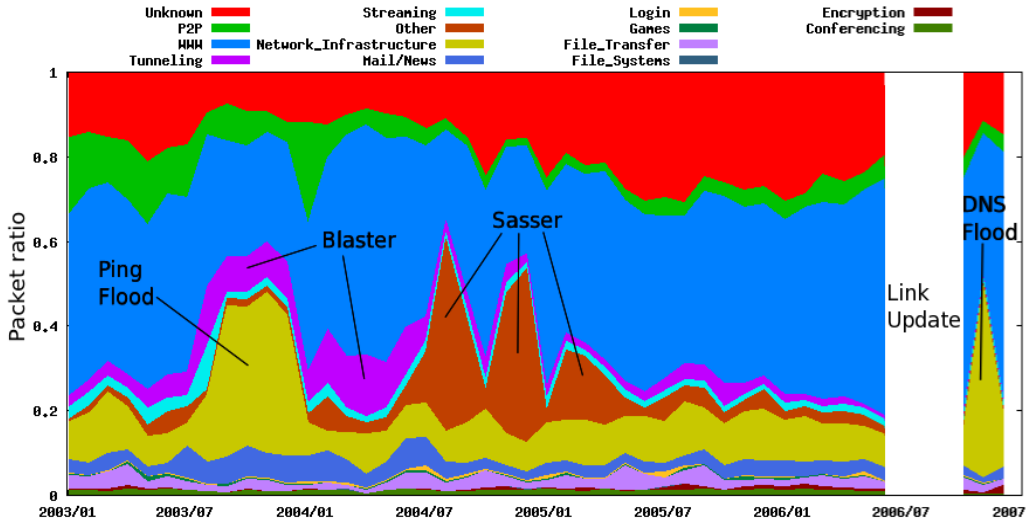
Figure 5.6: PDF of accuracy of four detectors for four years of MAWI traffic.

one analyzes time series and the gamma and KL detectors take advantage of histograms.

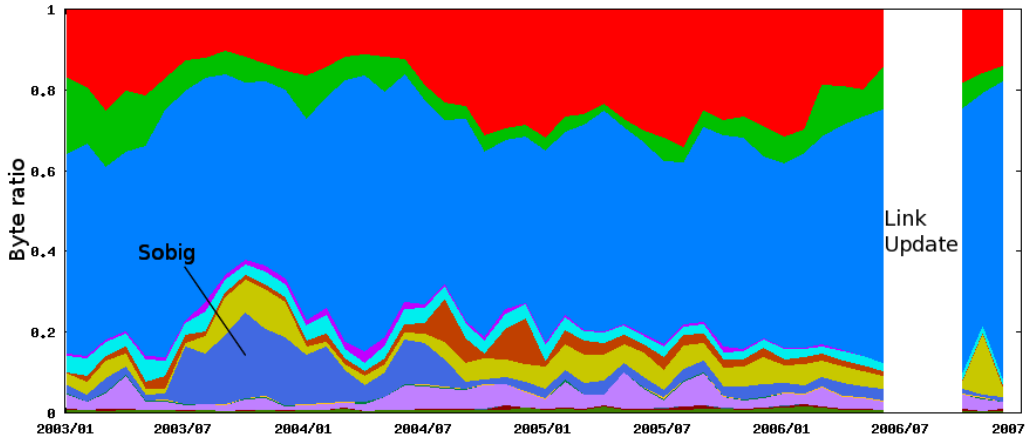
5.7.2 Reported anomalies

This section inspects the anomalies that are reported by the proposed adaptive detection method in order to evaluate its true and false positive ratio. Due to the lack of ground truth data (i.e., backbone traffic with annotated anomalies) the performance of the proposed method is evaluated using two methodologies; (1) A coarse-grained evaluation with prominent anomalies manually identified in the traffic. (2) A fine-grained comparison of the accuracy using three other detection methods and inspection of the traffic reported by the detectors and labeled as attack by the heuristics of Table 3.2.

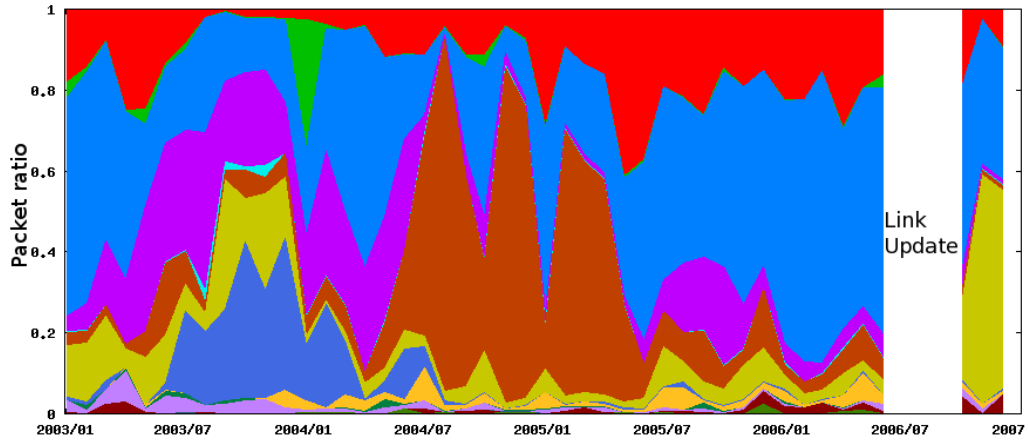
Prominent anomalies We manually inspected several characteristics of the analyzed traffic to identify the prominent anomalies that have to be reported by



(a) MAWI traffic in terms of packet



(b) MAWI traffic in terms of byte



(c) Traffic reported by the adaptive Hough-based detector

Figure 5.7: Application breakdown of the analyzed traffic and the results of the proposed method.

the detection method. Figure 5.7 displays two characteristics of the analyzed traffic, namely the percentage per application of transmitted packets and bytes. The application corresponding to each traffic is recovered using the CoralReef port-based classifier [2].

We identified five main events that have significantly affected the characteristics of the MAWI traffic from 2003 to 2006 (Fig. 5.7(a) and Fig. 5.7(b)). Four events are identified by inspecting the percentage of transmitted packets per application; from August 2003 to January 2004 we observed a substantial number of ICMP flows constituting a long-lasting ping flood. The spreading of the *Blaster* worm is also observed from August 2003 in the MAWI traffic. Another worm called *Sasser* is observed from June 2004 to June 2005 in the form of three peaks representing three outbreaks of different variants of the worm. After the update of the link in July 2007, an important traffic against DNS servers is observed. This traffic is particularly intense in the middle of November 2006 (e.g., the DNS traffic measured on the 2006/11/11 stands for 83% of all packets recorded this day). Regarding the percentage of transmitted bytes per application another event is observed from August 2003, it corresponds to the outbreak of a email-based worm, called *Sobig*.

The traffic transmitted by the three worms manually identified in the analyzed traffic (i.e., the Sobig, Blaster and Sasser worms) are successfully reported by the proposed adaptive method (Fig. 5.7(c)). Since these worms spread in the network by contacting a substantial number of peers the corresponding traffic highlights an abnormal dispersion in the destination IP address space that is easily identified by the proposed method. The adaptive method also effectively identifies the DNS flood appeared at the end of 2006 (Fig. 5.7(c)). This traffic is characterized by numerous hosts initiating several connections to a few servers. Thereby, the proposed method successfully detect this anomalous traffic because of its concentration in the destination IP address space and its distribution in the source IP address space.

Although the properties of the traffic have significantly varied over the four years (particularly after the link update), the proposed adaptive method efficiently detected anomalous traffic without any parameter adjustment from network operators.

Accuracy and attacks breakdown Based on the heuristics of Table 3.2, the proposed adaptive method is evaluated by accuracy comparison with the three other detection methods.

Figure 5.6 shows the accuracy achieved by the four detectors for each year of analyzed traffic. The average accuracy of the proposed method is higher than the one of the three other detectors during the four years of MAWI traffic. Among the three other detection methods the KL-based one is the best detector in terms of accuracy, moreover, it occasionally outperforms the proposed method (Fig. 5.6(b) and Fig. 5.6(c)).

The circumstances in which the KL-based detector remarkably outperforms the other detectors were thoroughly inspected and this highlighted the fact that this detector reports a high ratio of attacks but out of only a small number of alarms. Consequently, the KL-based detector achieves a high attack ratio along with a high false negative rate (i.e. missed anomalies). Figure 5.8 shows the quantity of attacks reported by each detector classified with the labels from

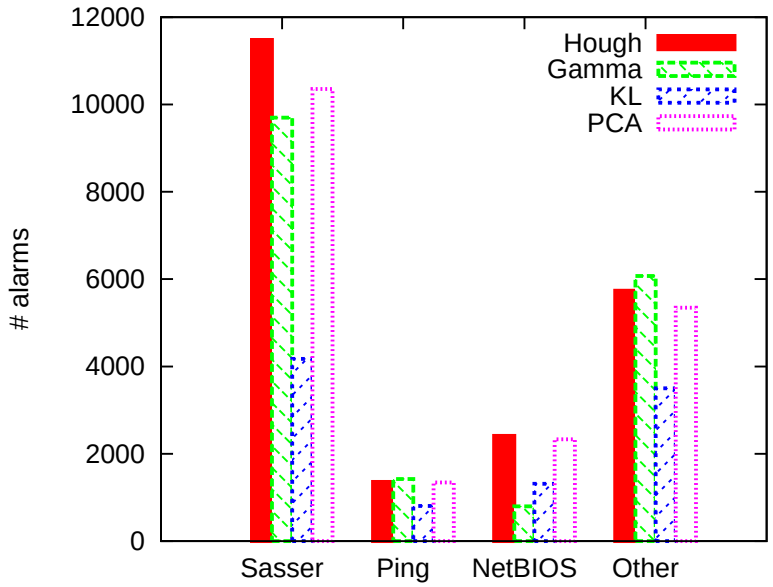


Figure 5.8: Breakdown of alarms reported by four detectors and classified as attacks during four years of MAWI traffic.

Table 3.2 (RPC is omitted as only 11 alarms of this kind were identified in the four years of traffic) and emphasizes the large amount of anomalies missed by the KL-based one.

The PCA and Gamma-based detectors, however, report the same quantity of attacks as the proposed method along with numerous alarms classified as special (Fig. 5.6). Although the proposed method is more sensitive to Sasser and attacks towards NetBIOS services, the Gamma-based method detected slightly more unusual ping traffic (66 alarms) and traffic labeled as flood (337 alarms) for the four years of analyzed traffic. Nevertheless, the PCA and Gamma-based detectors were considerably worse than the adaptive method in terms of accuracy, and this drawback is due to the quantity of traffic classified as special that was reported by these two detectors (i.e. high false positive rate).

The advantage of the adaptive method is to consistently adapt its time interval over the four years of analyzed traffic, and therefore, it constantly detects a large quantity of anomalous traffic while the number of reported benign traffic is low.

5.7.3 Missed anomalies

In order to highlight the limits of the proposed method this section inspects its false negative ratio, namely the proportion of anomalies that are missed by the proposed detection method. Nevertheless, due to the lack of ground truth data identifying the missed anomalies is a challenging task. The two following methodologies help us to pinpoint anomalous traffic that is not reported by the proposed detector; (1) A coarse-grained evaluation with prominent anomalies manually identified in the traffic. (2) A fine-grained inspection of anomalous

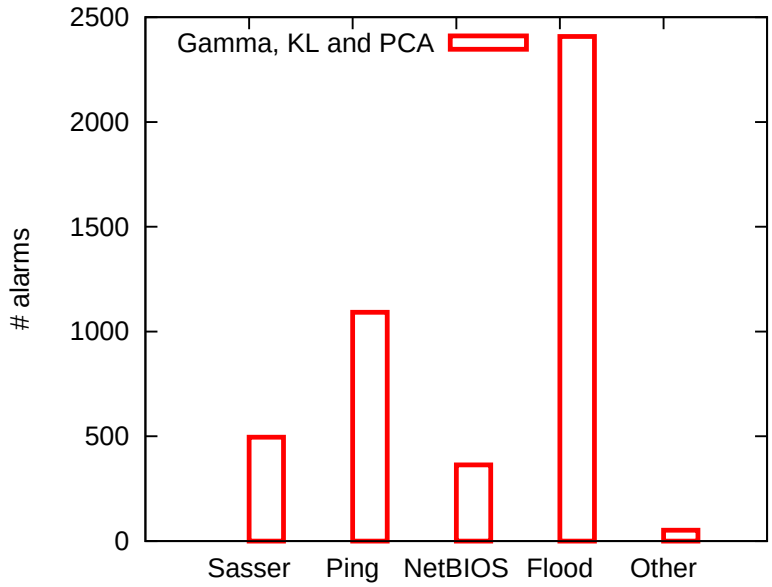


Figure 5.9: Breakdown of alarms reported by the Gamma-based, KL-based and PCA-based but not by the proposed detector during four years of MAWI traffic.

traffic reported by the three compared detection method (i.e., Gamma-based, KL-based and PCA-based) but not by the proposed adaptive method.

Prominent anomalies The manual inspection of the analyzed traffic revealed five prominent anomalies of which one is partially missed by the proposed adaptive method, that is the ping flood emerged in 2003 (Fig. 5.7(a) and 5.7(c)). This significant ping flood is characterized by numerous point to point high-rate flows (hereafter refered as alpha flows) using the ICMP protocol that are difficulty detectable by the proposed method for several reasons. First, since ICMP traffic have no port information it is only monitored in one of the five picture categories. Second, this traffic mainly consists of a set of long-lasting point to point flows without common source or destination, thus, preventing it to be shown as a line in analyzed pictures. Finally, the typical characteristic highlighting this anomalous traffic is the substantial number of transmitted packets whereas this feature is not monitored by the proposed detection method.

Attacks detected by other detectors We investigate the results of the three compared detection methods (i.e., Gamma-based, KL-based and PCA-based) to uncover the false negative rate and shortcomings of the proposed adaptive detection method. Since there is a low probability for a benign traffic to be reported as anomalous by the three compared detection methods, we consider a traffic as false negative if it is reported by all the detection methods but not the proposed one and it is categorized as attack by the heuristics of Table 3.2.

As shown in Figure 5.9, 80% of the anomalous traffic missed by the proposed detection method is labeled as ping or flood by the heuristics. This traffic

is mainly composed of alpha flows containing numerous Ping or SYN packets and representing one-to-one connections (contrarily to the successfully reported traffic from worms or DDoS attacks standing for one-to-many or many-to-one connections). These one-to-one connections appear in the analyzed pictures as single points and are difficult to identify using the proposed detection method. Furthermore, another characteristic of these flows is that they account for a large fraction of the total number of packets or bytes, however, these two traffic features are not monitored by the proposed detection method.

Since the proposed detection method is focusing on the distribution of the traffic features but not the volume of the traffic this method is insensitive to alpha flows. Also the proposed adaptive parameter tuning cannot overcome this shortcoming as it is inherent to the theoretical background of the proposed detection method. The class of anomaly misdetected by the proposed detector is however easily identifiable with a rate-based detection method that is monitoring the traffic volume. Therefore combining the proposed detection method and a rate-based detector would permit to detect a wider range of anomalies.

5.8 Discussion

In addition to propose an adaptive detector, this chapter reveals general considerations that have to be taken into account in the domain of network traffic anomaly detection.

The results presented in this chapter emphasizes the need of maintaining anomaly detectors parameter set optimally tuned. Indeed, Section 5.5.3 demonstrates that the performance of the anomaly detection method using fixed parameters is deteriorated when the characteristics of the traffic fluctuates (e.g., variations of traffic volume). Moreover, since anomalous traffic significantly alters the characteristics of the traffic anomaly detectors underperform especially during substantial anomaly outbreak. Consequently, adjusting the parameter set in regard to the fluctuations of the traffic is required to maintain the effectiveness of the detection method. These adjustments are enabled by investigating the relations between the theory underlying the detection method and the characteristics of network traffic.

The evaluation of the proposed adaptive detection method validates the efficiency of the adaptive mechanism to optimally set the parameters of the detector. Although this adaptive mechanism ensures the anomaly detector to perform optimally we observed that a certain class of anomaly is still misdetected by the proposed detector. This shortcoming is inherent to the design of the detection method thus independent from its parameter set tuning. In general, each anomaly detection method is expected to have weaknesses in detecting certain classes of anomaly, however maintaining its parameter set optimally tuned ensures that the detector is efficiently detecting the classes of anomaly it is designed for.

Section 5.7.3 highlights the shortcomings of the proposed detector and describes the class of anomaly undetectable by this anomaly detection method. This identification of the detection method shortcomings is a crucial task that allows us to understand the limits of this detector and ease the selection of a complementary detection method that would overcome the identified shortcomings. Consequently our results support the benefits of combining anomaly

detectors [67, 20, 8].

5.9 Summary

This chapter proposed a new anomaly detection method that takes advantage of image processing techniques to identify the flows with abnormal traffic feature distributions. Crucial challenges rarely addressed in the appropriate literature were uncovered by investigating the major drawbacks of this method; the sensitivity of anomaly detectors to traffic variations and the role of the time scale in anomaly detection. Addressing these two issues resulted in a significant improvement for the proposed detection method that overcomes any adverse conditions as it analyzes traffic within a time interval that is automatically adapted to the traffic throughput and the distribution of traffic features.

The evaluation of this adaptive method using real Internet traffic highlighted its ability to maintain a high detection rate while the traffic was significantly altered by anomalies. Therefore, these experiments indicated that the adaptive time interval enabled 26% more worm traffic to be detected, and decreased the false positive rate. The proposed adaptive detection method is also validated by comparing it with three other detection methods and using four years of real backbone traffic. The results highlighted that the proposed adaptive method allows for the detection of almost all the anomalies reported by the other detectors while it achieves the lowest false positive rate. We identified a class of anomaly that is disregarded by the proposed detection method and discussed the benefit of complementary detection methods to overcome these shortcomings.

While developing this anomaly detector one of the main difficulties we faced was to rigorously evaluate it. Because of the lack of ground truth data we validated the effectiveness of the proposed detector by comparing it with three other detectors. Although this methodology is commonly accepted by the research community, it involves numerous manual inspections of the traffic (based on heuristics and past experiences) that are error prone. Systematically and thoroughly evaluating an anomaly detector is indeed challenging, but, it is also the key task to identify and address detectors drawbacks and shortcomings. Therefore, the following chapter proposes a new methodology to help researchers to rigorously evaluate their anomaly detectors.

Chapter 6

Benchmarking and Combining Anomaly Detectors

6.1 Introduction

Because of the lack of ground truth data, the previous Chapter evaluated the proposed adaptive detector by comparing its accuracy to the one of four detectors and inspecting the breakdown of the traffic reported by these four detectors. In particular, the traffic commonly reported by the three compared detector was investigated to emphasize the shortcomings of the proposed detector (false negative rate), and the traffic exclusively reported by the proposed detector was investigated to highlight its benefits (true positive rate) or drawbacks (false positive rate). This investigations are essential to diagnose the drawbacks of the proposed detector and report useful feedback, however, they lack rigor as they are manual tasks that are time consuming and error prone.

Rigorously evaluating anomaly detectors is a challenging task that is commonly faced by researchers, thereby, distinct evaluation methodologies using traffic with real or simulated anomalies appear in the literature. With real anomalies, researchers evaluate anomaly detectors by manually checking the reported alarms [13, 16, 45, 48], or by comparing them to those reported by other anomaly detectors [22, 44, 45, 48]. Sometimes researchers construct ground truth data by manually inspecting the analyzed traffic [9]. However, these evaluations are hardly comparable, trustworthy, or reproducible, as they require significant human intervention and as traffic traces are usually inaccessible due to privacy issues. Also, a common shortcoming of these evaluation methodologies is the omission of the false negative rate of the detector, in spite of the fact that this metric is the good indicator of the number of missed anomalies and of the sensitivity of the detector to different kinds of anomalies.

Simulating anomalies is also a common way to evaluate an anomaly detector [44, 55, 62, 64]. In this case, the parameters of anomalies are tunable (e.g., in intensity and time duration), helping researchers to measure the sensitivity of their detectors to particular kinds of anomalies. However, simulating traffic

as diverse as it is on the Internet is notoriously difficult [19], especially for anomalous traffic. Consequently, the evaluation of a detector with simulated anomalies is restricted to certain kinds of anomaly, and thus, is insufficient for measuring the detector performance [59].

Goal

Ideally, an anomaly detector has to be evaluated using ground truth data containing real and nonspecific traffic where there is a wide range of anomalies. This ground truth data should be publicly available to allow all researchers to access the same data set and compare their results. Furthermore, the data set should follow the evolution of the Internet traffic to include traffic from emerging applications and anomalies. Currently, there is no such crucial data with ground truth; providing such data is our objective.

The goal is to find and label anomalies in the traffic from the MAWI archive [14], and to make it available to researchers so that they can refer to it while evaluating their own anomaly detection methods. The main advantages of the MAWI archive are that it is updated daily and it currently contains more than nine years of real publicly available Internet traffic data. However, manually labeling anomalies in such a large data set is certainly impractical, and therefore, the challenge we face is to accurately find anomalies in an automated and unsupervised manner. The numerous anomaly detectors that have recently been proposed in literature are the main support that will help us to reach the goal. Therefore, we are selecting diverse anomaly detectors and combining their results to accurately find anomalies in the MAWI archive. The synergy between detectors with different theoretical backgrounds allows a more accurate level of detection to be achieved. However, a key issue in combining such diverse detectors is that they report different granularities of the traffic that are difficult to rigorously compare.

Contributions

The contribution of the present chapter is twofold. Firstly, we establish a reliable methodology, which is based on graph and community mining, that compares and combines the results from any anomaly detectors, even though they operate at different traffic granularities. The proposed method outperforms the combined detectors, and enables us to precisely find twice as many anomalies as the most accurate detector from the experiments. Secondly, results are made available in the form of labeled data set, providing a benchmark for anomaly detection methods. The database currently stands for more than nine years of traffic and it is growing along with the MAWI archive. Furthermore, this approach permits the enhancement of the database over time by integrating the results from emerging anomaly detectors. Thus, the proposed database is constantly updated with new traffic and anomaly detectors, and it is a valuable tool to assist researchers designing anomaly detectors.

Proposed method

The method consists of four main steps, executed for each traffic trace:

1. Several anomaly detectors analyze the traffic and report alarms.

2. The similarities between the reported alarms are uncovered using a **similarity estimator** that groups similar alarms into communities.

3. Each community is investigated and classified by the **combiner**. Namely, the combiner decides if the community has to be reported as anomalous, or ignored, depending on the overall outputs of the detectors.

4. The anomalies are characterized using association rule mining on the combiner results so as to label anomalies in the analyzed data set.

Steps 2 and 3 are detailed in Section 6.2, and evaluated in Section 6.4. For that, the data set and anomaly detectors that are used are depicted in Section 6.3. Step 4 is described in Section 6.5. The results are further discussed in Section 6.6 and we conclude in Section 6.7.

6.2 Methodology

6.2.1 Similarity estimator

Since the benefit of combining detectors relies on the diversity among the detectors ensemble, we combine various anomaly detectors based on different theoretical backgrounds. Nevertheless, these different anomaly detectors are inherently reporting traffic at different granularities (e.g., flow, host, or packet) that are difficult to systematically compare. A detector might reports alarms at the host level, for example A_1 for IP_X, and another detector reports alarms at the flow level, for example B_1 and B_2 for $< IP_X, 80, IP_Y, 1234 >$ and $< IP_X, 80, IP_Z, 2345 >$. In that case, A_1 includes B_1 and B_2; however, telling that the three alarms are the same is hard because B_1 and B_2 are obviously reporting distinct traffic. Therefore, a rigorous method precisely measuring the similarities between alarms is required.

The similarity estimator presented in this section is an extension of a previous work [21]. Its role is to uncover the relations between the outputs of any kinds of anomaly detector. First, it reads the alarms reported by the detectors and the original traffic, and it extracts the traffic described by each alarm. Second, it constructs a graph that highlights the alarm similarities based on the traffic they have in common. Finally, similar alarms are identified by finding communities (i.e., dense connected components) in the graph.

Traffic extractor

The traffic extractor (called oracle in [21]) retrieves the traffic described by each alarm. Let an alarm be a set of traffic features that designates a particular traffic identified by a detector. The traffic extractor records the association between the alarm and this traffic. In [21], traffic associated with given alarms is always a set of packets, whereas the current work evaluates the benefits of associated traffic at different granularities: either packet, or flow (unidirectional or bidirectional). Figure 6.1 depicts the main differences in using flows and packets. The three alarms in Fig. 6.1 report three sets of packets from the same flow. By using packet as the traffic granularity, we observe that *Alarm2* and *Alarm3* have traffic in common but no packet is shared with *Alarm1*. Whereas

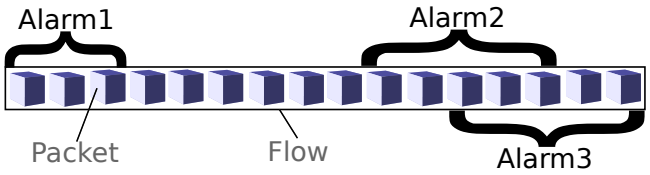


Figure 6.1: A flow composed of packets corresponding to three different alarms. *Alarm2* and *Alarm3* have common packets, whereas *Alarm1* consists of a distinct set of packets.

using flow, the three alarms are reporting the same traffic and thus will have similatities.

Graph generator

The graph generator uses the traffic retrieved by the traffic extractor to build an undirected graph called similarity graph, highlighting the similarities among all the alarms reported by the detectors. In this graph, a **node** stands for an **alarm**, and there is an **edge** between two nodes if their associated traffic intersects. In addition, an **edge** is weighted with a similarity measure that quantifies the **traffic intersection** of the two alarms it connects. Therefore, the similarity measure enables to discriminate edges connecting dissimilar alarms having an irrelevant number of packets or flows in common. We selected three similarity measures for the experiments: the Jaccard index, the Simpson index and a constant function. Since the Simpson index outperformed the two other metrics only this measure is discussed here. The Simpson index is defined as

$$S(E_1, E_2) = |E_1 \cap E_2| / \min(|E_1|, |E_2|)$$

where E_i is the traffic associated with alarm i. This metric ranges [0,1], where 0 means that the two traffic do not intersect, and that the two alarms are fully dissimilar; 1 means that they are identical or that one is included in the other.

Community mining

The similarity graph describes the alarm similarities, however alarms that are identical are not yet determined. Identical alarms are characterized in the graph as being a set of strongly connected nodes: this is called a community. Identifying the communities in a graph is a problem that has been extensively studied in the past [25]. Although numerous community mining algorithms have been proposed, the interest here focuses on those designed for sparse graph since the generated graphs have disconnected nodes (e.g., a false positive alarm reported by one detector). In the experiments, we selected a method based on the modularity: the Louvain algorithm [11]. This algorithm has the advantage of locally identifying the communities, thus allowing us to identify groups of a few alarms. Furthermore, this algorithm performs a fast and accurate analysis of the graph [25].

6.2.2 Combiner

The similarity estimator clusters similar alarms into communities: each community represents a set of distinct alarms (i.e., nodes) reported by various detectors. The role of the proposed combiner is to decide whether each community corresponds to an anomalous traffic or not. For that, the combiner classifies the communities into two categories, *accepted* and *rejected*, respectively standing for the communities reported as anomalous or those ignored. The class of a community is determined by a combination strategy, adapted from machine learning or pattern classifiers [43].

Background: combining detectors

A combination strategy is generally categorized as a detector selection or an output fusion. On the one hand, detector selection consists of selecting the detector that is the most suitable for classifying an element (i.e., a community in our case) and makes the same decisions as the single selected detector. Since each element is analyzed by only one detector, this approach is usually a good candidate for performing a quick analysis. However, selecting an appropriate detector is in practice challenging. In particular, the sensitivity of detectors in the context of network anomaly detection is misunderstood and prevents us from applying such techniques. On the other hand, output fusion makes no assumption on the detectors as it inspects the results of all the detectors. The output of a detector is assimilated to a vote for a certain class, and the combination strategy refers to a voting procedure.

In order to emphasize the advantages of combining detectors with output fusion let us review perhaps the oldest and best-known strategy, the majority vote. It is a basic, but still powerful way, where the final decision is the simple majority of the detectors outputs (i.e., more than 50 percent of the outputs). The probability of making the correct decision with the majority vote depends on the probability of each detector for providing the correct output, that is:

$$P_{maj}(L) = \sum_{m=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m}$$

where L is the number of detectors and p is their accuracy. The result, also known as the Condorcet Jury Theorem, is as follows;

- if $p > 0.5$, then $P_{maj}(L)$ is monotonically increasing in L and $P_{maj}(L) \to 1$ as $L \to \infty$.
- If $p < 0.5$, then $P_{maj}(L)$ is monotonically decreasing in L and $P_{maj}(L) \to 0$ as $L \to \infty$.
- If $p = 0.5$, then $P_{maj}(L) = 0.5$ for any L.

This theorem highlights the benefit of combining reasonable detectors (i.e., with an accuracy $p > 0.5$) over the use of a single detector.

Application to traffic anomaly detection

Each anomaly detector outputs a binary value telling if a traffic is anomalous or not. Namely, for each community in the similarity graph, a detector votes for

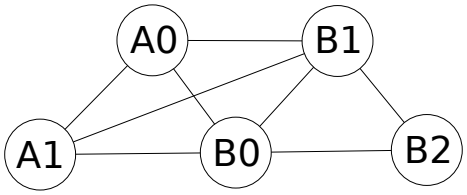


Figure 6.2: Example of community c_{ex} composed of five alarms. Assuming that the input of the similarity estimator, X_i, consists of the output of three detectors $X = A, B, C$ with three different parameter sets $i = 0, 1, 2$, then the confidence scores are: $\varphi_A(c_{ex}) = 0.66$, $\varphi_B(c_{ex}) = 1.0$ and $\varphi_C(c_{ex}) = 0.0$.

it being anomalous if at least one of its alarms is in the community. Although this is sufficient to compute a majority vote, this binary value is too coarse for a precise combination. Also, the votes of the detectors may significantly vary, depending on the tuning of their parameters.

To prevent these difficulties we propose to score the confidence of each vote of the detectors. Hereafter we refer to a certain detector with a specific parameter set as a **configuration**. Running a detector with several parameter sets and measuring the variability of its output quantifies its parameter sensitivity. The outputs of all configurations are merged through the similarity estimator, and the variability in the outputs is computed by inspecting each community. The **confidence score** φ of a detector d for a community c is defined as:

$$\varphi_d(c) = \phi_d(c)/T_d$$

where T_d is the total number of configurations with the detector d and $\phi_d(c)$ is the number of these configurations that reports at least one alarm belonging to the community c. The confidence score is a continuous value that ranges [0,1], 0 representing that a given detector ignores the community whereas 1 means that all configurations of the detector identify the community. For example, Fig. 6.2 is a community c_{ex} composed of fives alarms. Assuming that the input of the similarity estimator, X_i, consists of the output of nine configurations corresponding to three detectors $X = A, B, C$ with three different parameter sets $i = 0, 1, 2$, then the confidence scores for this community are: $\varphi_A(c_{ex}) = 0.66$, $\varphi_B(c_{ex}) = 1.0$ and $\varphi_C(c_{ex}) = 0.0$.

Combination strategies

Average, Minimum, & Maximum Let us now present three different combination strategies that aggregate the confidence scores relative to a given community c in a value $\mu(c)$, and classify a community c as *accepted* (i.e., labeled anomalous) only if $\mu(c) > 0.5$.

Aggregating the confidences score of a community by *average* allows us to rely equally on the votes of all the detectors. Formally, for a community c and using L detectors, the average is: $\mu(c) = \frac{1}{L}\sum_{i=1}^{L}\varphi_i(c)$. In the example shown in Fig. 6.2 the average of all the confidence scores equals 5/9, and thus, this combination strategy would classify the community c_{ex} as accepted.

Another strategy consists in selecting the *minimum* confidence score. This pessimistic decision classify a community as accepted only if all the detectors

support this decision. Consequently, the ratio of false positive is substantially reduced at the cost of an increase in the ratio of true negative. Formally, the decision made for the community c depends on its minimum confidence score: $\mu(c) = \min_i\{\varphi_i(c)\}$. In the example shown in Fig. 6.2, the minimum of all the confidence scores is 0, and thus, this combination strategy would classify the community c_{ex} as rejected.

On the contrary, a third strategy is to select the *maximum* confidence score. This optimistic decision classify a community as accepted only if at least one detector supports this decision. Consequently, the ratio of true positive is substantially increased, but so is the ratio of false positive. Formally, the decision made for the community c depends on its maximum confidence score: $\mu(c) = \max_i\{\varphi_i(c)\}$. In the example shown in Fig. 6.2, the maximum of all confidence scores is 1, and thus, this combination strategy would classify the community c_{ex} as accepted.

Correspondence analysis: SCANN Correspondence analysis [10] is a multivariate statistical technique for analyzing multiway tables. It represents a data set in a lower-dimensional space based on its singular value decomposition. Although its role is similar to the principal component analysis one, correspondence analysis is designed for categorical data.

Using correspondence analysis, Merz [54] proposes an unsupervised combination strategy called SCANN that is used here as an alternative combination strategy. This method stores all the decisions of the detectors in a table, such that each entry is a vector representing the votes of all detectors for a certain community. Then, using correspondence analysis, this table is reduced such that the entries are smaller vectors containing only the main features characterizing the detectors votes. The benefit is that the reduced table contains only the significant votes. For instance, an irrelevant detector is one constantly making the same vote; in the first table built by SCANN, such a detector's votes are constant values, hence they will be ignored in the reduced table because they do not help for discriminating between the communities.

The reduced table contains the characteristics of each community in a low-dimensional space. Onto this low-dimensional space, SCANN projects two reference points which are two representative communities unanimously reported by the detectors as accepted, or as rejected. The class of each community is then determined according to which representative community the closest in the low-dimensional space.

Note that, since correspondence analysis is designed for categorical data, SCANN is unable to deal with the confidence scores previously defined. In order to take advantage of different configurations, the implementation of SCANN that is used consider directly the binary outputs of different configurations as its input.

6.3 Data set and anomaly detectors

6.3.1 Data set

The traffic we are labeling is from the MAWI archive samplepoints B and F (see Section 3.1 for more details on MAWI). In the experiments, the similarity

estimator is evaluated with the traffic traces from the first week of every month from 2001 to 2009, whereas the combiner is evaluated using all the traffic traces from 2001 to 2009.

6.3.2 Anomaly detectors

Four unsupervised anomaly detectors, based on distinct statistical-analysis techniques, are employed for this experiments; one is the Hough-based adaptive detector proposed in Chapter 5, and the three other are the PCA-based detector, the gamma-based detector and the KL based detector presented in Section 3.3.

As these detectors report traffic at different granularities, the proposed similarity estimator is necessary to compare their results. The confidence score for each detector is obtained by tuning them with three different parameter sets corresponding to optimal, sensitive or conservative setting. Hence, for experiment, the input for the proposed method consists in the 12 outputs of all the configurations (4 detectors using 3 parameter tunings). The main ideas of the four detectors are as follows.

6.4 Evaluation

6.4.1 Similarity estimator

In this section the proposed similarity estimator is evaluated using the alarms reported by the twelve configurations. In particular, the sensitivity of the similarity estimator to the traffic granularity is discussed.

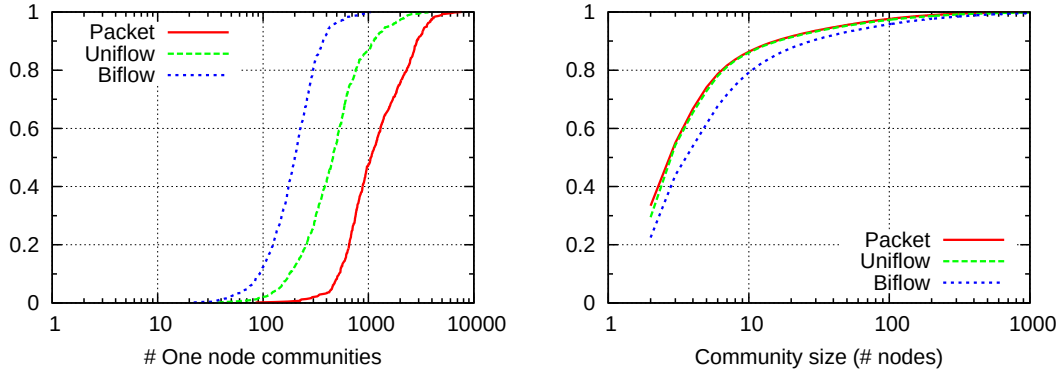
Metrics for evaluation

The following tools enable a comparison of the results given by different configurations of the similarity estimator, and a validation of its efficiency.

Size of communities The size of a community is the number of nodes that belong to that community, that is the number of similar alarms clustered in the community. We distinguish a specific class of community, called the **single communities**, that is the size 1 communities (communities with a single alarm). An alarm falls into a single community if the similarity estimator fails to find other alarms related to it. Consequently, we expect a good similarity estimator to minimize the number of single communities.

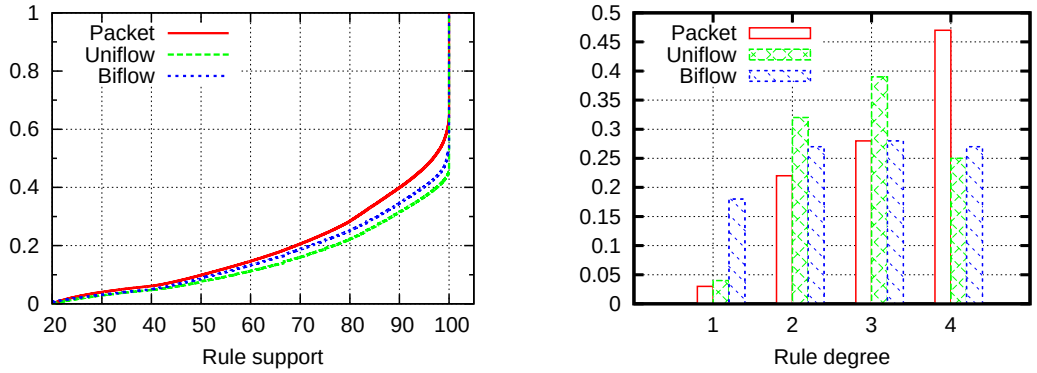
Obviously, the number of single communities is not a sufficient scale to evaluate the similarity estimator, as it reports a value 0 when all the alarms are connected regardless of their similarities. Consequently, we also score the relevance of the communities using association rule mining.

Traffic summary with association rules One key task for validating the efficiency of the proposed similarity estimator is to inspect the traffic corresponding to each community. The goal of this inspection is to assess that each community is a group of related alarms standing for the traffic with common features; this is a similar goal to the *dominant state analysis* presented in [72], or the *association rule mining* of [13].



(a) CDF of number of single communities per traffic trace (log. scale)

(b) CDF of size of communities except for single communities (log. scale)



(c) CDF of rule support except for single communities

(d) Probability distribution of rule degree except for single communities

Figure 6.3: Characteristics of communities reported by the similarity estimator with different traffic granularities.

The traffic related to each community is profiled here using an association rule mining algorithm that finds sets of features, which are called rules, describing the prominent trends in a given list of properties. We choose the Apriori [7] algorithm as it is a well-known algorithm for achieving association rule mining. The Apriori algorithm efficiently counts the candidate rules in a breadth-first search manner. It finds all the rules that describe more than s elements of the data, where s is the only parameter of this algorithm. We slightly modify the Apriori algorithm to express s in a percentage rather than a fixed number of elements. For instance, the modified version of Apriori computed with $s = 20\%$ outputs each rule that describes at least 20% of the data.

In the experiments, the modified version of Apriori is arbitrarily tuned with $s = 20\%$, and it analyzes the packets or flows corresponding to each community. Thereby, the resulting rules describe the main characteristics of the traffic corresponding to a community in the form of 4-tuples — source IP address, source port number, destination IP address, destination port number. For example, a community corresponding to the traffic from a HTTP server IPA to two hosts, IPB and IPC, is depicted by the rules $< IPA, 80, IPB, * >$ and $< IPA, 80, IPC, * >$, where $*$ means that no specific destination port was identified.

The relevance of a community as a set of alarms is quantified by two efficiency metrics based on its rules:

- The **rule degree** of a community is the average number of items in its rules. For example, if a community has the two following rules, $< IPA, *, IPB, * >$ and $< IPA, 80, IPC, 12345 >$, then its rule degree is $(2 + 4)/2 = 3$. The rule degree ranges $[0, 4]$, and values close to 4 mean that the rules are specific, and thus, correspond to a particular kind of traffic, whereas values close to 0 mean that the mining rule algorithm failed to characterize specificities of the traffic.

- The **rule support** of a community is the percentage of data covered by all the rules of this community. For instance, if the two previous rules cover, respectively, 50% and 25% of the traffic captured by the community, and because the rules are disjoint, then the rule support is $50 + 25 = 75\%$.

Traffic inspection The heuristics of Table 3.2 help to characterize traffic corresponding to communities. These heuristics are designed from previous works [12, 22] and the manual inspection of the MAWI traffic. They assign three general labels to the traffic, "Attack", "Special", or "Unknown", highlighting the type of traffic corresponding to a community. Furthermore, they inspect only the TCP flag, ICMP code, and port number related information, and allow us to conduct a fair evaluation as they are independent of the mechanisms of the chosen detectors.

Results

The similarity estimator is evaluated using three distinct traffic granularities (packet, unidirectional flow, and bidirectional flow) by looking at the size of the communities and by inspecting the traffic corresponding to it.

Size of communities The results highlight the benefit of flows to uncover similar alarms as Fig. 6.3(a) depicts a substantial decrease in the number of single communities using unidirectional or bidirectional flows. In addition, we observed a significant increase in the size of the communities when using bidirectional flows (Fig. 6.3(b)). These observations emphasize the ability of the similarity estimator to relate more alarms using flows.

Traffic summary Let us check the consistency of the communities, that is whether all the alarms of the same community are actually related. Community consistency is analyzed using the rules that are assigned to each community by the modified Apriori algorithm. Figure 6.3(c) shows that the best rule support is achieved by using unidirectional flow, and the results obtained when using bidirectional flows are slightly inferior. By using unidirectional flows more than 50% of the communities have the rule support equal to 100%. However, the results are different regarding the rule degree (Fig.6.3(d)); the most accurate rules are obtained using packets whereas the least accurate are from bidirectional flows. We observe about 18% of the communities found using bidirectional flows are described with rules having only one traffic feature.

To understand which communities are suffering from coarse rules, thus containing dissimilar alarms, we investigated the relation between the size of communities and the rules efficiency. Figure 6.4 is the rule support, rule degree, and community size obtained when using unidirectional flows. We observe that the largest communities tend to have a rule degree equal to 1 and a rule support equal to 100%. A manual inspection of these communities reveals that they have coarse rules reporting a single traffic feature, usually a well known port such as 80 or 53 (Fig. 6.5). However 90% of the communities, namely with less than 20 nodes (Fig. 6.3(b)), have a rule degree higher than 2 and a rule support higher than 75% (Fig. 6.4). Similar observations are made using bidirectional flows, whereas using packets the rule degree is higher than 2.5 and the rule support above 70%. Therefore, the consistency of the communities identified the the similarity detector is satisfactory for the three traffic granularities. Selecting a traffic granularity is a trade off between the size of the communities and their consistency.

Traffic inspection Figure 6.5 depicts the intersections of the detectors outputs and the type of corresponding traffic. The main results are: (1) The intersection of the four detectors is significantly small in comparison to the total number of identified communities, therefore, the four detectors are sensitive to distinct traffic; (2) The number of single communities containing one alarm only from the PCA-based detector is significantly high, while only a few single communities are reported by the KL-based detector. Furthermore, 6% of the single communities identified by the PCA-based detector are labeled "Attack" whereas this ratio is significantly higher for other detectors: 33% for Hough, 22% for Gamma and 56% for KL. We also observe that the PCA-based detector represents 58% of the non-single communities identified by one detector. Thus, the output of the PCA-based is separated from others and its detection ratio is low in terms of the heuristics of Table 3.2. Regarding the communities identified by more than one detector, their attack ratio increases in tandem with the number of detectors identifying them.

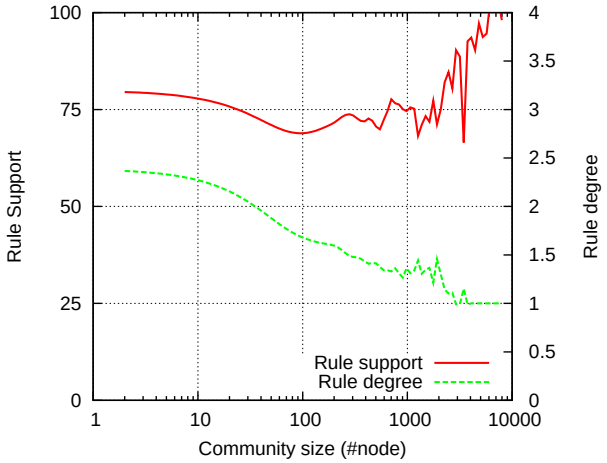


Figure 6.4: Rule degree, rule support, and size of communities identified by the similarity estimator using unidirectional flow. The curves are smoothed using weighted spline approximation.

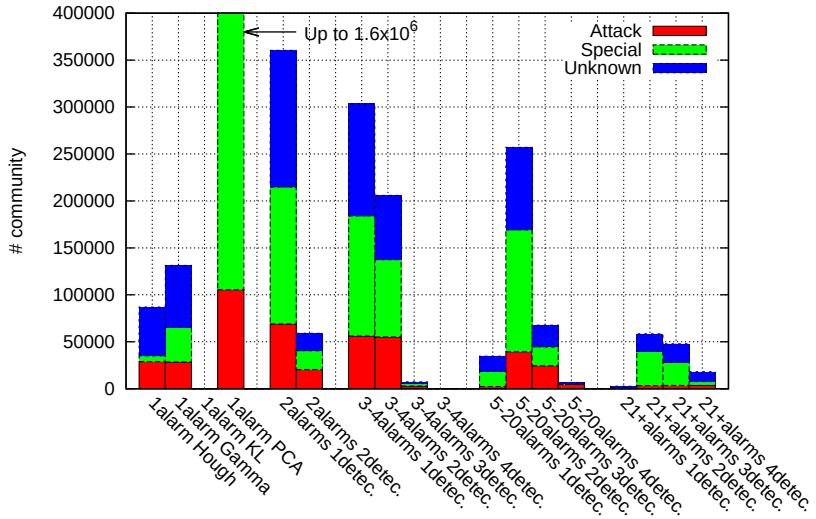
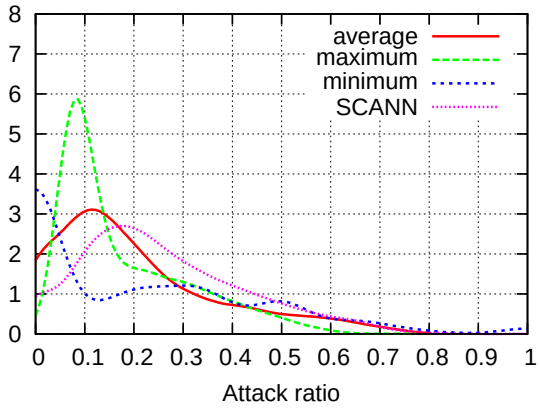
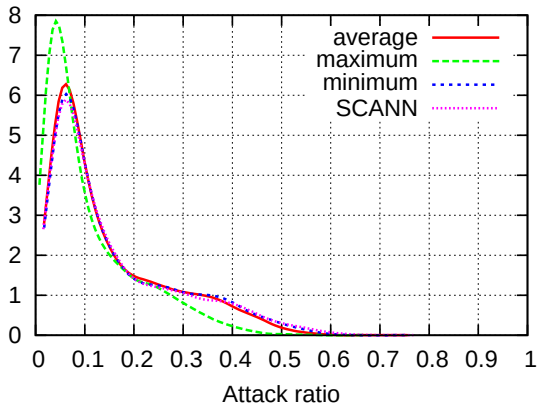


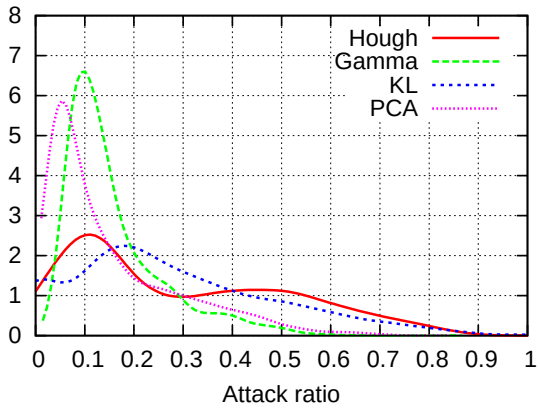
Figure 6.5: Number of communities as function of the size and the number of detectors reporting their alarms. Colors indicate the type of traffic corresponding (see Table 3.2).



(a) PDF of attack ratio for accepted communities (large probability for high attack ratio is better)



(b) PDF of attack ratio for rejected communities (large probability for low attack ratio is better)



(c) PDF of attack ratio for detectors (large probability for high attack ratio is better)

Figure 6.6: PDF of attack ratio for four combination strategies and four detectors evaluated on 9 years.

The communities identified by several detectors certainly highlight anomalous traffic that have to be reported by the combiner. Nevertheless, the communities reported by a single detector have to be thoroughly investigated as they perhaps stand for anomalous traffic, particularly for those reported by the Hough, Gamma and KL-based detector.

6.4.2 Combiner

Attack ratio

In this work, measuring the accuracy of the four combination strategies is a contradictory task due to the lack of ground truth data. We bypass this issue by inspecting the results of the combiner with the heuristics of Table 3.2.

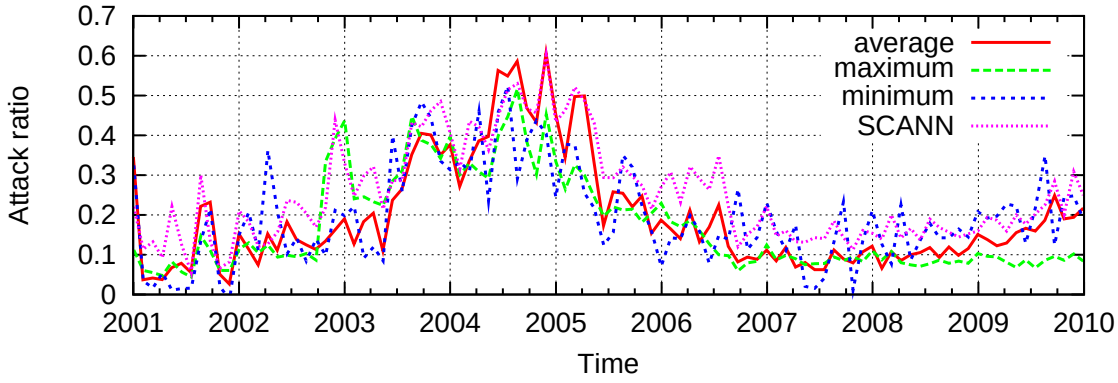
The heuristics label the communities reported by the similarity estimator into three groups: "Attack", "Special", and "Unknown". Since a relevant combination strategy is presumed to report the largest proportion of the communities labeled "Attack", we define the **attack ratio** as the amount of communities labeled "Attack" divided by the total number of identified communities. The combination strategies are expected to also report numerous communities labeled "Special" or "Unknown", thus low attack ratio, as the proposed heuristics might label incorrectly several kinds of anomalies. Nevertheless, the attack ratio is a reliable indicator that helps us to identify the best combination strategy, that is the one accepting the highest ratio of communities labeled "Attack" (Fig. 6.6(a) and 6.7(a)) and rejecting the lowest ratio of communities labeled "Attack" (Fig. 6.6(b) and 6.7(b)).

Comparison of combining strategies

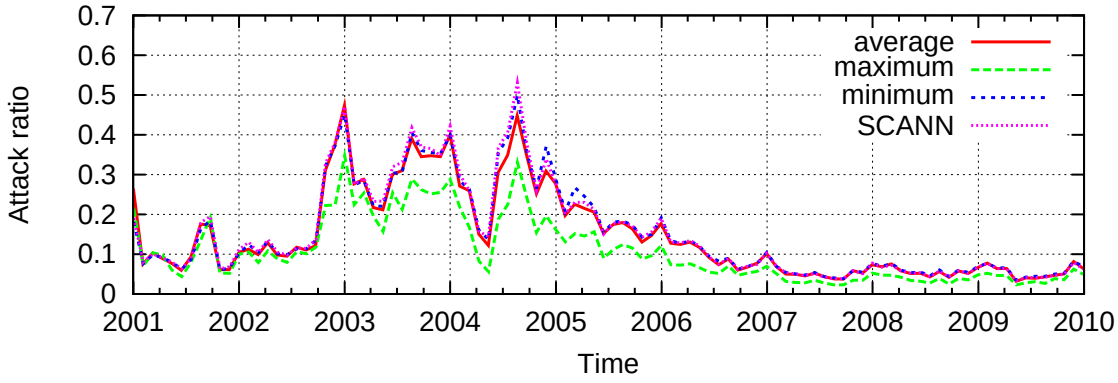
This section evaluates the ability of the four combination strategies to label communities. The analyzed communities are produced by the similarity estimator with the alarms reported by the four detectors on nine years of MAWI traffic and using unidirectional flow as traffic granularity. These communities are classified by the combination strategies into two classes (i.e., accepted and rejected) and the attack ratio of both classes are computed for each day of the analyzed traffic. Probability density functions (Fig. 6.6) and time-series of the attack ratio (Fig. 6.7) are displayed.

Regarding accepted communities, the best combination strategy is SCANN as it features the largest probability for highest attack ratio (Fig. 6.6(a)). Nevertheless, the best combination strategy regarding rejected communities is the *maximum* strategy because it has the largest probability for lowest attack ratio (Fig. 6.6(a)). Since the prominent variance between the attack ratio probability of the accepted communities and the one of the rejected communities highlights the best combination strategy, the experiments support SCANN as the best strategy for discriminating the communities representing anomalous traffic.

The probability density functions of the four anomaly detectors attack ratio emphasizes that all detectors, except the KL-based one, have an average attack ratio that is inferior to SCANN (Fig. 6.6(c)). Although the KL-based detector attack ratio is close to that of SCANN, the thorough investigation of the SCANN output in Section 6.4.2 asserts that SCANN detected twice more traffic than the KL-based detector.



(a) Accepted community attack ratio (higher value is better)



(b) Rejected community attack ratio (lower value is better)

Figure 6.7: Attack ratio of four combining strategies for nine years of MAWI traffic.

Table 6.1: Four measures quantifying benefits and losses when using SCANN.

		SCANN	
		Accepted	Rejected
Label	Attack	gain_{acc}	cost_{rej}
	Special, Unknown	cost_{acc}	gain_{rej}

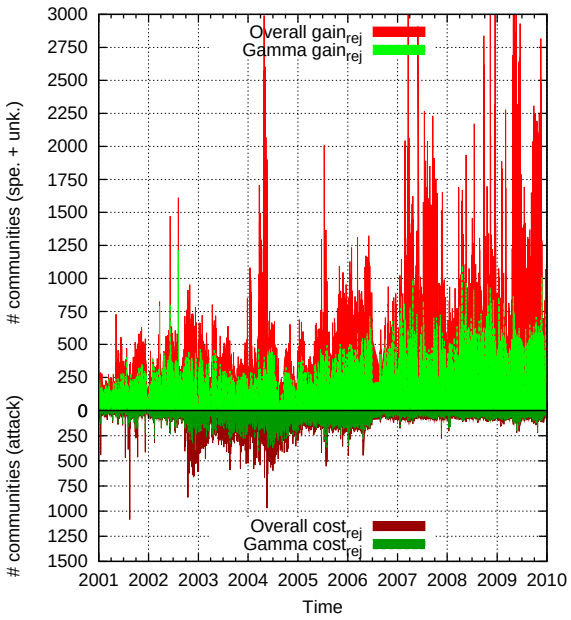
The time evolution of the attack ratio for each combination strategy is depicted in Figures 6.7(a) and 6.7(b). Although the SCANN algorithm is not constantly outperforming the other combination strategies, it never has the worst attack ratio. The low attack ratio of both the accepted and rejected communities from 2007 is due to the simple heuristics listed in Table 3.2 that mislabeled the numerous elephant flows from peer-to-peer traffic and other anomalies using random ports. Still, between 2007 and 2010, the efficiency of SCANN is noticeable as its attack ratio for accepted communities was 2 to 3 times higher than the one for rejected communities.

However, the increase in the attack ratio for rejected communities from 2003 to 2005 (Fig. 6.7(b)) highlights the particular traffic that is missed by the combination strategies. The release of the Blaster worm in August 2003 followed by the release of the Sasser worm in May 2004 were two of the main events reported during this time period [12]. These two worms have substantially affected the main characteristics of the traffic and the four detectors were differently affected by this variance in traffic. The detectors reported numerous alarms that were not related to those of the other detectors, and consequently, the combiner failed in distinguishing several anomalous traffic. Nevertheless, this shortcoming of the combiner is inherently diminished by the combination of more detectors thus increasing the intersection of their outputs. Furthermore, we observed that selecting a single detector to analyze this traffic was also challenging, as the attack ratio of each detector critically fluctuated during this time period.

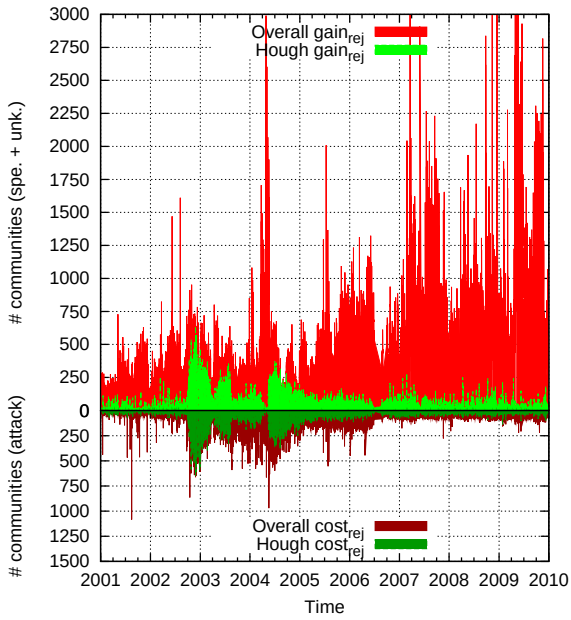
Inspecting the SCANN output

We evaluate the benefits and the losses of using SCANN based on the four quantities depicted in Table 6.1. For rejected communities the gain_{rej} is the amount of communities that are labeled "Special" or "Unknown", whereas the cost_{rej} is those labeled "Attack". Symmetrically, for the accepted communities, the gain_{acc} is the amount of communities that are labeled "Attack", whereas the cost_{acc} is those labeled "Special" or "Unknown".

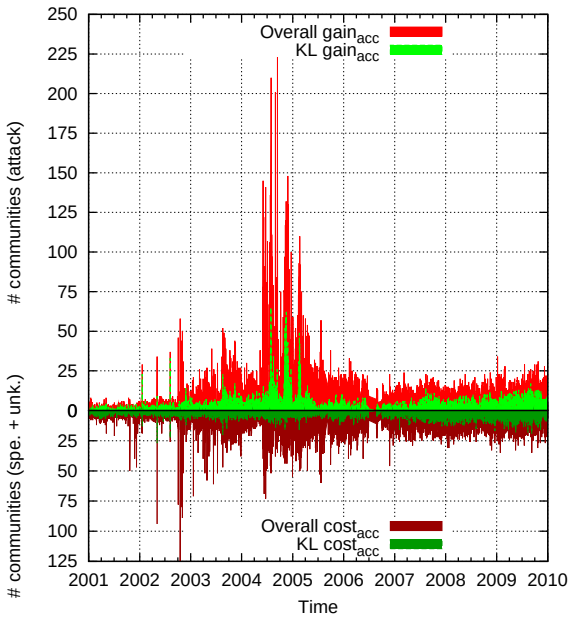
Rejected communities Figure 6.8 shows the breakout of communities classified by the SCANN algorithm. The two left-hand side plots are the communities rejected by SCANN where the alarms reported by the Hough- and the Gamma-based detectors are highlighted. The gain_{rej} for the Gamma-based detector (Fig. 6.8(a)) is substantial and stands for more than half of the overall gain_{rej} for all the detectors. Nevertheless, the high true positive rate of the Gamma-based detector is emphasized by its cost_{rej}, which represents most of the communities labeled "Attack" and rejected by SCANN. The gain_{rej} of the Hough-based detector was slightly higher than its cost_{rej} exhibiting the low false



(a) Rejected communities (Gamma)



(b) Rejected communities (Hough)



(c) Accepted communities (KL)

Figure 6.8: Communities classified by SCANN as rejected with the alarms from the Hough (a) and the Gamma-based (b) detectors highlighted, and the communities accepted by SCANN with the alarms from the KL-based detector highlighted (c).

positive rate of this detector. In addition, Figure 6.8(b) depicts the high sensitivity of this detector to worm spreading (i.e., Blaster during 2003 and Sasser during 2004). The results for the PCA and KL-based detectors are omitted, as the former one has a significant gain_{rej} that is close to the overall gain_{rej}, and the latter one has no cost_{rej} and an negligible gain_{rej}. The experiments also exhibited the contamination of the normal subspace of the PCA-based detector [62] by the first release of the Sasser worm, and thus, a considerable gain_{rej} for this detector at this time period.

The PCA-based detector reported a significant number of alarms that were mostly unrelated to the alarms of other detectors (Fig. 6.5), particularly after the MAWI link update at the end of 2006 (overall gain_{rej} in Fig 6.8(a)). Since SCANN rejected most of the communities reported only by the PCA-based detector, the number of communities rejected by SCANN was notably higher than those of the accepted one (Fig. 6.8(b) and 6.8(c)). Figures 6.8(b) and 6.8(c) suggest that the overall cost_{rej} was higher than the overall gain_{acc}. However, we emphasize that the communities accepted by SCANN are more significant, in terms of the number of alarms and the amount of corresponding traffic, than the rejected ones.

Accepted communities A manual inspection of the SCANN output reveals that several accepted communities contain only alarms from a single detector. Therefore, for the nine years of analyzed traffic, 8 accepted communities were identified by only the PCA-based detector, 325 accepted communities were identified by only the Gamma-based detector, 2467 accepted communities were identified by only the Hough-based detector, and 352 accepted communities were identified by only the KL-based detector. Meaning that 82% of the communities reported exclusively by the KL-based detector are accepted by SCANN. This highlights the advantage of SCANN over the *average* combination strategy. Whereas the *average* combination strategy inherently rejects all the communities reported by a single detector, SCANN performs a finer analysis that emphasizes the output from accurate detectors and allows for the acceptance of small communities identified exclusively by these detectors. Indeed, the SCANN algorithm factorizes the detectors decisions by disregarding the unnecessary ones, thus, SCANN ignores the output of the detectors that are making irrelevant decisions and emphasizes the other results. For example, in the experiments the PCA-based detector output was mainly separated from the outputs of the other detectors (the single communities in Fig. 6.5). Consequently, SCANN frequently disregarded the PCA-based detector and accepted only 8 of the numerous communities exclusively identified by this detector. Conversely, the Hough-based detector reports more relevant alarms as many are related to those from other detectors, and thus, SCANN selects 2467 communities reported by only this detector.

In the experiments the best detector was the KL-based one (Fig. 6.6(c)). Almost all the alarms from this detector were related to another alarm (Fig. 6.5) and are accepted by SCANN. However, about 50% of the communities accepted by SCANN and labeled "Attack" are not identified by the KL-based detector (Fig. 6.8(c) and 6.9). These communities are mainly reported by the three other detectors and they highlight the high false negative rate (i.e., anomalies missed) of the KL-based detector (Fig. 6.9).

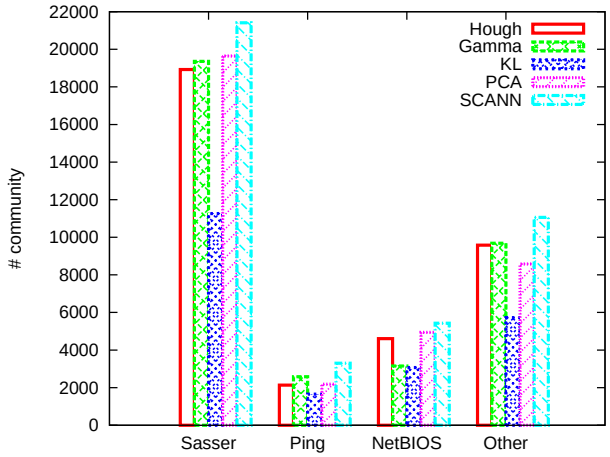


Figure 6.9: Breakdown of communities accepted by SCANN and labeled "Attack" by heuristics.

SCANN low dimensional space Combining the four detectors with SCANN allows us to improve the results of the most accurate detector and to ignore the false alarms reported by all the detectors. However, Fig. 6.7(b) suggests that it misclassified several communities. As described in Section 6.2.2, the SCANN algorithm maps the communities in a reduced space and classifies them based on their distances to two reference points. Let d_{acc} and d_{rej} be the distance from a community to the reference point standing for, respectively, accepted and rejected communities, then the relative distance of the community is defined as $(d_{rej}/d_{acc}) - 1$. This metric ranges $[0, \infty)$, where 0 means that the community is on the threshold whereas higher values highlight the communities that are distant to it. The inspection of the rejected communities exhibits that the relative distance of those labeled "Attack" is lower than the one of those labeled "Special" or "Unknown" (Fig. 6.10).

We varied the discriminating threshold of SCANN during the experiments to investigate possible improvements. Tuning the threshold to accept more communities tends to increase the fluctuations of the attack ratio of SCANN. For example, accepting all the communities within a relative distance of 0.5 achieved an attack ratio of 0.7 during the Sasser outbreak, but sometimes deteriorated the attack ratio, therefore, no global improvement was observed.

6.5 MAWI labeling

Step 4 of the proposed method consists in labeling the analyzed traffic, here the MAWI archive. In agreement with the previous evaluation, the traffic is labeled using the SCANN combination strategy, and the similarity estimator was executed using unidirectional flow. Since several communities contained a significant number of alarms, we retrieved the common traffic features corresponding to all the alarms from the same community with the association rule mining algorithm presented in Section 6.4.1, and assigned labels to the traffic described by the community rules.

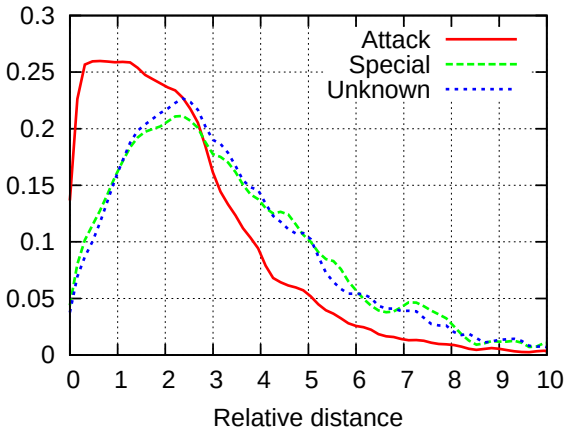


Figure 6.10: PDF of rejected communities relative distance classified with the heuristics of Table 3.2.

Using the SCANN output we define a simple traffic taxonomy with four different labels: *Anomalous*, *Suspicious*, *Notice*, and *Benign*.

- The traffic accepted by SCANN is labeled *Anomalous*, meaning that these traffic are abnormal and should be identified by any efficient anomaly detector.
- The traffic rejected by SCANN and having a relative distance lower or equal to 0.5 is labeled *Suspicious*. Most of these traffic are probably anomalous but are not clearly identified.
- The traffic also rejected by SCANN but having a relative distance greater than 0.5 is labeled *Notice*. Although these traffic are not anomalous and should not be identified by any anomaly detector, we do not label them as benign in order to trace all the alarms reported by the combined detectors.
- The other traffic is labeled *Benign* because none of the anomaly detectors identified it.

This labeling of the MAWI traffic is publicly available in the form of a database named MAWILab [3]. This database assists researchers in measuring the detection rate of their anomaly detector. The results of the emerging detectors can be accurately compared to the labels of MAWILab by using a similarity estimator like the one presented in this work.

6.6 Discussion

In addition to its accurate detection, the proposed method has several advantages that are presented in this section.

The graph-based similarity estimator proposed in Section 6.2.1 is a valuable support for systematically benchmarking a detector against other detectors that report traffic at a different granularity. Indeed, by clustering diverse detectors

alarms into communities, it allows the automated inspection of numerous detectors outputs in a rigorous manner.

Also, the community rules obtained from the rule mining algorithm consist of concise descriptions of the traffic identified by the numerous alarms being merged into the communities. Therefore, an anomalous traffic reported by numerous similar alarms is annotated with a single label. Thus, the number of labels assigned to the MAWI archive is significantly inferior to the number of alarms reported by the four detectors, and the labels are intelligible to humans.

Following the expansion of the MAWI archive, MAWILab is updated daily to track the latest trends in Internet traffic and upcoming anomalies. Furthermore, we will also take into account the results from emerging anomaly detectors, to improve the quality and variety of the labels over time. Indeed, by including new results from upcoming detectors the overlaps of the detectors outputs are emphasized and the accuracy of SCANN is improved. Therefore, MAWILab is constantly enhanced and represents a reference data set over time. In order to ease the evolution of MAWILab, we are planning to establish a collaborative system allowing researchers to easily contribute by submitting their anomaly detector or results.

We emphasize that the proposed implementation has the advantage of handling manual annotations or annotations from traffic classifiers [15]. Indeed, the similarity estimator is able to deal with any traffic annotations [21] containing at least two timestamps and one traffic feature. This significant ability of the approach allows us to label traffic with an exhaustive taxonomy. For instance, by adding in the method input the annotations from a traffic classifier, the similarity estimator aggregates similar alarms and corresponding annotations in the same community. Afterwards, the combiner classifies the communities by ignoring the annotations, but the accepted communities are still reported with the extra information provided by the annotation.

The goal of this work is to find and label traffic anomalies off-line, so we assume no constraint is placed on the execution time of the approach. Nevertheless, the experiments revealed that the current implementation requires only a few minutes to combine alarms with a 15-minute traffic trace, thus enabling for real time analysis. However, the study of concurrently running anomaly detectors in real time is left for future work.

6.7 Summary

We proposed a methodology that find network traffic anomalies in the MAWI archive by comparing and combining the results from four anomaly detectors. The approach consists of two main ingredients; first, a graph-based similarity estimator systematically uncovers the relations between the alarms reported by the detectors, second, a combiner classifies the similar alarms using a combination strategy. We evaluated the effectiveness of both using different traffic aggregations and combination strategies. The experiments emphasized the benefit of combining detectors with SCANN, a strategy based on dimensionality reduction, as it ignored irrelevant alarms and detected twice more anomalous traffic than the more accurate combined detector.

The established methodology allows us to accurately detect anomalies in the MAWI archive and precisely assign concise labels. The results are updated

daily using the MAWI archive and are publicly available [3] to assist researchers in benchmarking their detectors. We encourage researchers to contribute to the proposed system by submitting to us their results or detectors, so we can maintain a reliable labeling of the MAWI archive.

Chapter 7

Discussion

This dissertation aims at increasing the reliability of anomaly detectors by addressing different problems that raise in distinct scopes of the anomaly detection domain. Thereby, we propose a practical anomaly detector that outperforms current detectors, and, we introduce a new methodology to systematically benchmark any anomaly detectors.

This chapter summaries the contributions, shortcomings, unexpected findings and consequences of the proposed approaches.

7.1 Pattern-recognition-based anomaly detector

In order to provide network operators with a practical and reliable anomaly detector, we follow a simple approach that consists in monitoring the traffic with intuitive pictures and identifying the anomalous traffic through specific patterns. Thereby, the proposed detection method relies on image processing and pattern recognition techniques which is a novelty in anomaly detection.

Similarly to anomaly detectors that are looking for outliers, the proposed anomaly detector is able to identify unknown anomalies emerging in the network, however, the key idea underlying this method is fundamentally different from the previous detection methods. Indeed, the design of the proposed method is initiated by previous observations that enabled to profile a general pattern of the anomalous traffic, therefore, by identifying this pattern this detector reports traffic that is guaranteed to highlight the anomalous characteristics inherent to the pattern (contrarily to the outlier-based detector that reports traffic that is distinct from the majority of the analyzed data). This fundamental difference allows the proposed detector to reports reliable alarms, and, this improvement against previous anomaly detectors is supported by our experiments (Section 4.4). Contrarily to the outlier-based detectors the proposed approach does not assume the benign to be the majority, thereby, it maintains decent detection performance when anomalous traffic is dominant during important anomalies outbreaks. Our experiments also shown that the proposed detector is able to report mice flows that are contributing in distributed large scale attacks.

The proposed detector relies on a pattern standing for traffic that is abnormally dispersed in certain traffic feature spaces. Although this pattern encompass most of the current anomalous traffic (e.g. DDoS, worms, scan), the

proposed anomaly detector is inherently restricted to these classes of anomaly. For example, our experiments pinpointed the shortcoming of the proposed detector that is the misdetection of alpha flows (i.e., elephant point to point flows). Generally anomaly detectors are expected to have shortcomings in identifying certain classes of anomaly and this knowledge is essential for network operators to predict the limit of their detection systems. Furthermore, this identification of the detector main weakness also permits to take advantages of complementary detectors. For example the proposed detector is insensitive to alpha flows that are easily detected by rate-based anomaly detectors, thus, in practice the combination of the proposed pattern-recognition-based detector and a rate-based anomaly detector overcomes this shortcoming.

7.2 Parameter tuning

The other challenge we address to ensure the reliability of the proposed anomaly detector is to maintain its parameter set optimally tuned. Thereby, we analyze the fluctuations of the optimal parameter set over time and uncover its relations with the fluctuations of the traffic. This analysis enables us to automatically select the optimal parameter set according to the characteristics of the traffic. The resulting reliability increase is supported by our experiments using four years of MAWI traffic (Section 5.7).

This work provides substantial insights into the relations between the detector performance, its parameter set, and the characteristics of the traffic. Usually the parameters of an anomaly detectors are selected according to the characteristics of the analyzed link (e.g., average bandwidth) and adjusted only if the properties of the link significantly change. Nevertheless, similarly to the previous work of Himura et al. [32] our analysis highlights the importance of tuning parameters at fine-grained time scale. Indeed, our experiments reveal that certain values of the optimal parameter set are varying by one order of magnitude while analyzing only 15 minutes of traffic significantly altered by anomalies. Constantly selecting the parameter set according to traffic variations is a laborious task that cannot be defer to network operators, thus, each anomaly detector requires automated mechanisms to rapidly adjust its parameter set in regard to traffic fluctuations.

Furthermore, our experiments illustrate that the optimal parameter set is particularly varying during the outbreak of significant anomalies, consequently, a detector using a fixed parameter set is expected to fail in reporting anomalous traffic especially during the outbreak of dominant anomalies. Similar observations have been reported in the literature, in particular outlier-based anomaly detectors that misreport traffic when a significant anomaly contaminate the majority of the traffic [60, 62]. The solution we propose to this problem is practical for network operators as it automatically adjusts the parameter set according to the characteristics of the analyzed traffic (contrarily to the previous works that require past or training data [62, 32]).

7.3 Benchmarking anomaly detectors

While developing the proposed pattern-based-anomaly detector we faced difficulties in evaluating its effectiveness because of the lack of ground truth data. These difficulties are commonly faced in the domain of anomaly detection and they prevent researchers to rigorously demonstrate the reliability of their detectors and provide accurate feedback necessary for improving emerging detectors. This dissertation addresses certain challenges faced in evaluating anomaly detectors especially the difficulties of comparing several anomaly detectors outputs.

We propose a benchmarking methodology based on graph theory that compares the alarms of several anomaly detectors based on different theoretical backgrounds and reporting traffic at different granularities. Using graph allows us to uncover the similarities and differences of the alarms reported by several detectors in a systematic manner. The effectiveness of this benchmarking methodology is validated by our experiments using four independent anomaly detectors and 10 years of backbone traffic.

This approach is a novelty in anomaly detection enabling rigorous comparison of detectors results that is not possible using only the traditional performance metrics (e.g., true positive or false positive rate). In the literature detectors are usually compared by inspecting their ROC curves, however, ROC curves omit crucial information that is required to provide substantial feedback on the evaluated detector. For example comparing the ROC curves of two detectors permits to select the one with the best accuracy, but, it does not help to answer at the following basic questions: does the best detector missed anomalies found by the other one? Are all anomalies reported by both detectors? Addressing these questions is possible by manually inspecting the reported alarms, nevertheless, the proposed methodology enables a systematic and rigorous approach to this process.

The proposed methodology has also the advantage of reporting groups of similar alarms that uncover characteristics of the underlying anomalous events. For example, Figure 7.1 illustrates a group of similar alarms reported by two anomaly detectors using real Internet traffic. The two detectors are; the proposed pattern-recognition-based detector reporting traffic at the packet level, and the gamma-based detector reporting either source or destination IP addresses. The group consists of 29 alarms; 27 are from the gamma-based detector (i.e., red rectangles) and 2 from the other detector (i.e., green ellipses). All these alarms are reporting abnormal DNS traffic, however, the alarms on the right and left hand side of Fig.7.1 (both labeled 200.24.119.113) report a DNS server under heavy traffic whereas the rest of the alarms report clients soliciting this service. By grouping all these alarms together our method permits to report the flooded server and the blamed clients at the same time. Whereas, by analyzing individually alarms raised by clients, one may misunderstand the distributed characteristic of this network activity — similar to DDoS, flash crowd, or botnet activity — and misinterpret each alarm.

Unexpectedly we notice that the proposed benchmark method is also useful in inspecting the output of a single anomaly detector. In the example of Figure 7.1, the proposed method merges 27 alarms from the gamma-based detector, therefore, it reports the 27 alarms at once and assists network operators to find the root cause of this anomalous traffic.

Due to its flexible design the proposed method is also able to handle an-

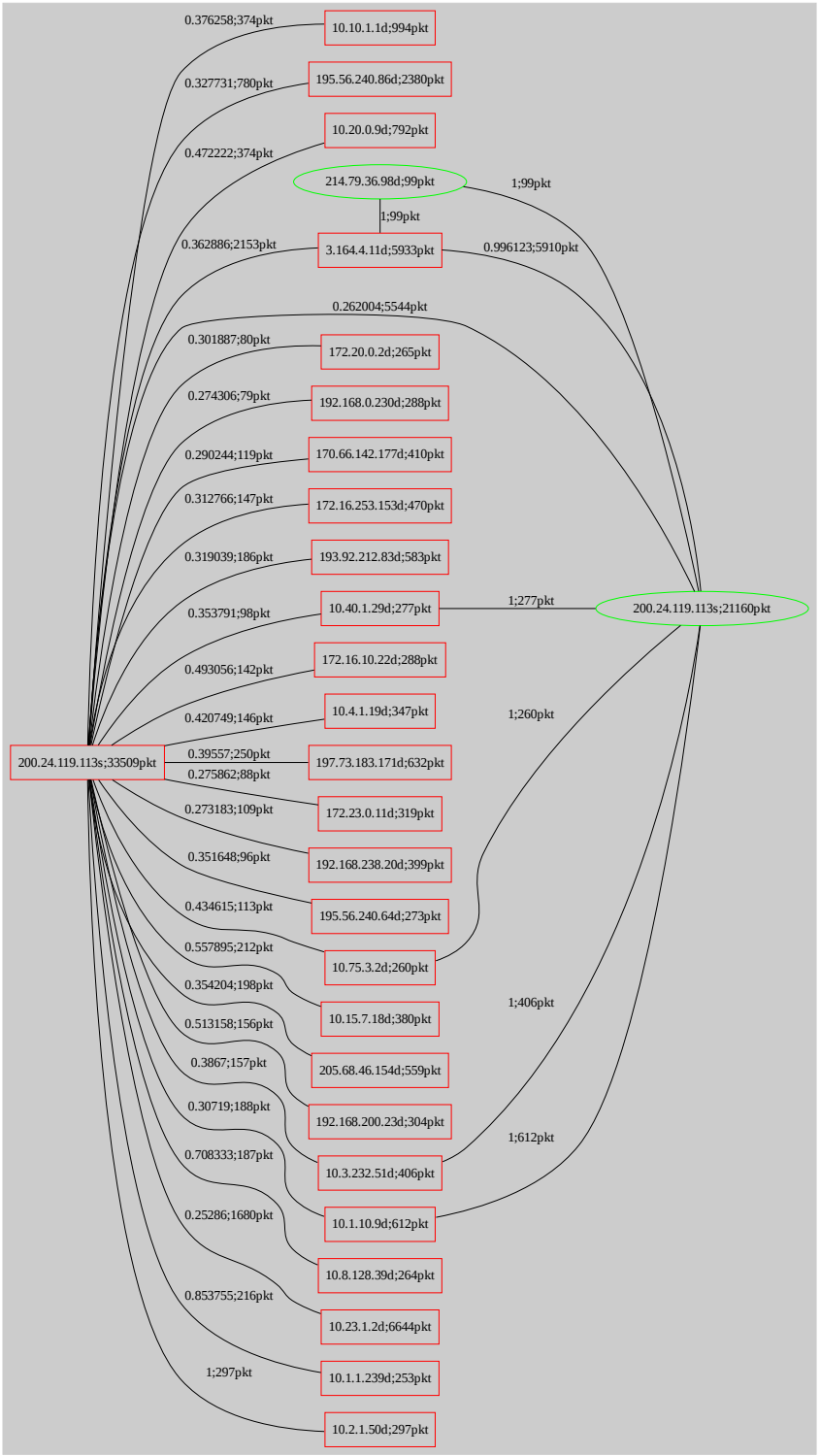


Figure 7.1: DNS traffic reported by 29 alarms from two anomaly detectors.

notations from the traffic classifiers (i.e., identification method recovering the applications corresponding to the traffic). Therefore, the proposed method allows to aggregate similar alarms with the corresponding traffic annotations, and, provides detailed information of the traffic reported by the anomaly detectors.

Relations between anomaly detectors and kinds of anomaly

Using the proposed benchmarking methodology, four diverse detectors and 10 years of backbone traffic, this dissertation highlights detectors differences that are rarely reported in the literature. Indeed, the results presented in Section 6.4.1 show that each detector is reporting a significant number of unique alarms standing for traffic not reported by other detectors. Thereby, we deduce that the detectors are reporting distinct kinds of anomalous traffic. This statement corroborates our previous observations with the proposed pattern-based detector; the proposed pattern-based anomaly detector is able to detect a specific kind of anomaly that is represented by small flows and missed by the compared detectors, and, the proposed detector misses anomalous alpha flows that are reported by the compared detectors. Our results (Section 6.4.1) also highlight the singularities of the three compared detectors; the PCA and gamma based detectors identify specific kinds of anomaly not identified by the other detectors, whereas, the KL-based detector reports mainly anomalies commonly detected by all the detectors. Since the four analyzed anomaly detectors detect different kinds of anomaly, we emphasize that network operators need to consider these differences while selecting anomaly detectors.

Because each detector detects (or misses) specific kinds of anomaly combining them is a promising approach that requires more attention from researchers. This dissertation illustrates the advantages of combining anomaly detectors by studying an unsupervised combination strategy that is indeed outperforming all combined detectors and reporting a variety of anomalies.

Combining anomaly detectors

The benefits of the proposed benchmarking methodology are demonstrated by applying it to combine diverse anomaly detectors. Thereby, using an unsupervised combination strategy and four anomaly detectors we implemented a reliable detection method that is identifying two times more anomalies than the best detector. The main contribution of this application is to combine anomaly detectors that are fundamentally different and take advantage of their synergistic effect.

The results of this combination of anomaly detectors using the MAWI archive are provided to the research community in order to evaluate emerging anomaly detectors. Therefore, researchers can use it as ground truth data to evaluate their detector which can be afterward integrated in the proposed combination strategy, thus, improving the provided results.

Despite the flurry of anomaly detectors proposed in the last decade, the combination of detectors have been rarely studied in the domain of anomaly detection. Nevertheless, the advantages of combining anomaly detectors to provide reliable detection tools to network operators are significant, thus, combining anomaly detectors deserves more attention in the future. Our main contribution in this broad topic is to propose a benchmark methodology that enables to

combine any kinds of anomaly detectors, however, the substantial tasks of selecting an optimal detector ensemble and a sophisticated combination strategy are beyond the scope of this dissertation and left for future works.

Chapter 8

Conclusion

8.1 Concluding remarks

The constant emergence of anomalous traffic in Internet is a serious problem that holds the attention of many researchers. Thereby, during the last decade numerous anomaly detection methods have been proposed to assist network operators in detecting and diagnosing traffic anomalies. Several of these proposals are particularly interesting as they have the ability to prevent the outbreak of new and unknown anomalies by applying statistical analysis of the traffic.

Nevertheless, these anomaly detectors have several common drawbacks that discredit their reliability in practice. In this dissertation we addressed these drawbacks and proposed two solutions operating a two different scopes. First, we designed, implemented, and evaluated an anomaly detector that overcomes several drawbacks of current detectors, second, we proposed a methodology to benchmark anomaly detectors and took advantage of it to emphasize the benefits of combining several anomaly detectors.

8.1.1 Pattern-recognition detector

We proposed a new approach to detect traffic anomalies using image processing and pattern recognition. This detection method was evaluated by analyzing 6 years of real Internet traffic and by comparing its results with those of two other anomaly detectors. This evaluation highlighted the strengths of the proposed detector that are its intuitive mechanisms and its good detection performance.

The proposed anomaly detector identifies anomalies by taking advantages of a pattern recognition technique and a pattern featuring common characteristics of anomalous traffic. Thereby, the proposed approach is fundamentally different from the two typical approaches found in the literature, namely the ones based on signature matching and the ones based on outlier detection. The proposed detector is indeed located between these two approaches as it is explicitly looking for malicious traffic (contrarily to outlier-based methods that are looking for traffic with singularities) and has the ability of detecting unknown anomalies (contrarily to signature-based methods).

Parameter tuning

In practice one of the main challenge with Internet traffic anomaly detectors is to select the optimal parameter set. Consequently, we inspected the relation between the parameter set of the pattern recognition technique employed by the proposed anomaly detector and the characteristics of the analyzed traffic. This analysis highlighted the significant performance deterioration caused by traffic fluctuations and the need of adjusting detectors parameters according to the variances of the traffic. Consequently, we designed an adaptive anomaly detector that automatically tunes its parameter set according to the traffic fluctuations. The effectiveness of this adaptive anomaly detector was validated by comparing its results to those of three other anomaly detectors using four years of Internet traffic.

According to these experiments we expect that most of the anomaly detectors requires parameter adjustments in regard to the traffic fluctuations, and we emphasize the need for researchers to address this issue while developing anomaly detectors.

8.1.2 Benchmarking anomaly detectors

Rigorous evaluation of the anomaly detectors is a crucial task towards enhancing the quality of current and emerging detectors. Consequently, in this dissertation we addressed two common challenges that penalize researchers to efficiently evaluate anomaly detectors; (1) rigorously comparing the results of several detectors based on different theoretical backgrounds. (2) Providing common ground truth data (i.e. traffic traces with labeled anomalies).

Benchmark methodology

We proposed a methodology to relate alarms reported by several detectors although they are expressed in different ways and represent distinct granularities of the traffic. Our approach relies on the abstraction level of graph theory, thereby, graphs are generated from alarms and the original traffic to uncover the similarities of alarms. An algorithm finding community structure permits to distinguish coherent sets of nodes in the graph standing for sets of similar alarms. An evaluation using 10 years of traffic highlighted the effectiveness of this method to cluster similar alarms reported by different anomaly detectors.

By rigorously comparing the results of several anomaly detectors this graph-based methodology enables to systematically inspect the differences between different anomaly detectors. Unexpectedly, the proposed method also has benefits when it is used with only one detector; it aggregates similar alarms reported by a single detector, thereby, it clarifies the output of the detector and helps in accurately describing anomalous traffic.

Detector combination

Due to its benefits, the proposed benchmark methodology enabled us to compare and combine the results from diverse anomaly detectors. Thereby, using the proposed method and an unsupervised combination strategy we developed a method combining four anomaly detectors. The synergy between anomaly detectors permitted to detect twice more anomalies than the most accurate

detector, and to reject the numerous false positive alarms reported by the detectors. Furthermore, using this combination of detectors we automatically located a variety of anomalies in a data base containing 10 years of traffic, and, we provided our results in the form of a common ground truth data that assists researchers in evaluating their detectors.

8.2 Futures perspectives

Combination strategies are appealing as they theoretically permit a performance increase over the combined methods. While, they have received a lot of attention in the field of data mining, the attempts to apply these techniques to anomaly detection are rare. Therefore, the combination of anomaly detectors deserves more attention in the future, especially, the three following topics.

Sensitivity of anomaly detectors

A key task to effectively combine several anomaly detectors is to understand the strengths and weaknesses of each detector. Consequently, investigating the sensitivity of several detectors to distinct traffic is an important task that needs particular attention. The objective is to systematically identify for each detector the traffic characteristics that tend to raise true or false positive alarms. Also, a formal description of anomaly detectors sensitivity is required to make a good use of it in an automated manner.

Anomaly detectors ensemble

One can easily understand that combining detectors that output identical results is hopeless. In fact the assets of combining anomaly detectors is originated by the diversity of the detectors. Therefore, selecting diverse detectors that would have a synergistic effect is also a crucial task. Past studies in the data mining community have proposed diversity measures to build effective ensemble, however, these measures has been validated only on specific cases and applying them to anomaly detection is challenging.

Combination strategy

Combining multiple detectors consists in classifying the traffic based on the set of decisions reported by the combined detectors and knowledge on the detectors. Numerous combination strategies have been proposed in the field of data mining and they require different knowledge on the combined methods. For example, combination strategies based on Bayes theory may rely on the accuracy of each method whereas unsupervised combination strategies do not requires such knowledge. Consequently, the insights of the task on the anomaly detectors sensitivity (described above) are essential to determine knowledge on the detectors and select an effective combination strategy.

Publications

List of the publications related to this PhD dissertation.

Journal papers

- Romain Fontugne, Kensuke Fukuda. "A Hough-transform-based Anomaly Detector with an Adaptive Time Interval", ACM Applied Computing Review, pp.41-51, vol.11, No.3, ACM, Summer 2011 (extended version of SAC2011).

- Romain Fontugne, Toshio Hirotsu, Kensuke Fukuda. "A Visualization Tool for Exploring Multi-scale Network Traffic Anomalies", Journal of Networks, Special Issue: Performance Evaluation of Communication Networks and Systems, pp.577-586, vol.6, No.4, Academy Publisher, April 2011 (extended version of SPECTS2009).

- Romain Fontugne, Yosuke Himura, Kensuke Fukuda. "Evaluation of Anomaly Detection Method based on Pattern Recognition", IEICE Transactions on Communications, pp.328-335, vol.E93-B, No.2, IEICE, February 2010.

International conference papers with review committee

- Romain Fontugne, Kensuke Fukuda. "A Hough-transform-based Anomaly Detector with an Adaptive Time Interval", ACM Symposium on Applied Computing (SAC 2011), pp.468-474, Taichung, Taiwan, March 21-24, 2011.

- Romain Fontugne, Pierre Borgnat, Patrice Abry, Kensuke Fukuda. "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking", ACM CoNEXT 2010, p.12, Philadelphia, PA, Nov.30-Dec.3, 2010.

- Romain Fontugne, Pierre Borgnat, Patrice Abry, Kensuke Fukuda. "Uncovering Relations Between Traffic Classifiers and Anomaly Detectors via Graph Theory", 2nd COST-TMA Workshop, LNCS (Volume 6003) pp.101-114, Zurich, Switzerland, Apr.7, 2010.

- Romain Fontugne, Toshio Hirotsu, Kensuke Fukuda. "A Visualization Tool for Exploring Multi-scale Network Traffic Anomalies", International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2009), pp.274-281, Istanbul, Turkey, Jul.13-16, 2009.
- Romain Fontugne, Toshio Hirotsu, Kensuke Fukuda. "An image processing approach to traffic anomaly detection", Proceedings of 8th Asian Internet Engineering Conference (AINTEC 2008), pp.17-26, Bangkok, Thailand, Nov.18-20, 2008.

Student workshop papers with review committee

- Romain Fontugne, Pierre Borgnat, Patrice Abry, Kensuke Fukuda. "Towards Systematic Traffic Annotation", 5th ACM CoNEXT student workshop, pp.15-16, Rome, Italy, Dec.2, 2009.
- Romain Fontugne, Yosuke Himura, Kensuke Fukuda. "Anomaly Detection Method based on Pattern Recognition", Student workshop of 10th Passive and Active Measurement Conference (PAM 2009), p.2, Seoul, South Korea, Apr.1-3, 2009.

Appendix A

Visualizing Internet Traffic and Characterizing Anomalies

A.1 Introduction

The Internet has become a common medium for communication and information exchange providing many attractive services for ordinary users. A victim of its own success, Internet traffic is still growing at a fast rate and contains an increasing amount of anomalies such as misconfigurations, failures, and attacks. These improper uses of network resources consume bandwidth and adversely affect the performances of networks. Thus, these anomalies penalize legitimate applications from using an optimal amount of network resources. Since the core of the Internet is particularly deteriorated by anomalous traffic, quick and accurate detection of anomalies in the backbone traffic has been a hot topic (e.g., [9, 45, 16, 23]). However, due to the lack of ground truth data for backbone traffic, evaluating an anomaly detector is quite challenging and tricky [30]. Therefore, researchers must validate their results from their anomaly detectors by manually investigating the dump files or flow records. This is a baffling problem as it is laborious to identify a few thousand harmful packets from millions of innocuous ones.

Nevertheless, visualizing network traffic is a valuable tool for in investigating dump files. The main advantage of graphical representations is to highlight the significant features of the traffic, thus the main properties of the traffic are understood at a mere glance. Moreover, several degrees of information are retrieved by monitoring the various representations that depict different aggregations of the traffic. For example, a time series is useful for analyzing the time evolution of a single feature for a huge amount of flows. Whereas, a graphlet [39] depicts several features of only a few flows.

In this article, we propose a tool to visualize, explore, and understand network traffic at any temporal and spatial (address and port) scale. Our main contribution is to provide a tool that assists researchers, or network operators, in understanding and validating alarms reported by their anomaly detectors. The

proposed tool provides six basic features to help researchers inspect network traffic and evaluate anomaly detectors:

- Network traffic is displayed at different resolutions, and the user is able to zoom in/out along the time axis or address/port space.
- The tool provides different types of scatter plots (corresponding to IP addresses, or port numbers) and time series (e.g., throughput and average packet size). Since these graphical representations are intuitive views, the tool simultaneously displays two views and provide an exhaustive description of the traffic.
- Understanding backbone traffic involves inspecting various sub-traffics, and therefore, the tool allows to easily move along the network traffic in time and space (i.e. address and port number space).
- The tool retrieves all the details concerning the monitored traffic in the form of accurate graphlet and textual data.
- Anomalies identified by anomaly detectors are displayed by this tool, and thus, researchers and network operators are able to easily validate the veracity of the detected anomalies.
- The current implementation runs on different platforms on a daily basis, it uses no intermediate database, and it directly reads dump files (pcap form [4]).

We evaluated the tool on several kinds of traffic; darknet traffic reveals shapes highlighting anomalous traffic, and similar patterns are also observed in the backbone traffic. Furthermore, we demonstrate the help provided by the tool in identifying recent and sophisticated attacks such as the Conficker worm. We also conduct a manual inspection of anomalous traffic reported by anomaly detector, and list several typical patterns highlighting anomalies (in accordance with those reported in [23]).

A.2 Related work

Various visualization tools assist researchers and network operators in monitoring network traffic. For example, Fischer et al. [18] and Goodall et al. [28] presented two interesting tools focusing on anomaly detection. The former [18] monitors traffic related to local hosts based on a TreeMap visualization. It is used to check alarms reported by IDS, and to identify large-scale attacks aiming at local hosts. The latter, TNV [28], highlights the connections between the hosts sorted within a matrix. The traffic between local and remote hosts is clearly displayed, and all the information about the packets is accessible. However, these two tools only display a limited number of hosts (e.g., about 100 hosts for TNV on a 1280x1024 display), and their home-centric view is not suitable for backbone traffic where the terms local and remote hosts are meaningless.

InetVis [71] is a visualization tool used to monitor the network traffic in three-dimensional scatter plots. Traffic is mapped into a cube [47] highlighting the specific patterns for particular anomalies. Although InetVis is adequate enough for monitoring small or extracted traffic (e.g., using IDS [36]), figures

generated from heavy traffic (e.g. backbone traffic) are difficult to read and omit a lot of information. Moreover, textual information concerning plotted points cannot be obtained using this tool, whereas, information like port numbers, IP addresses, or TCP flags are usually required to validate anomalies. NVisionIP [46] is another visualization tool that cannot retrieve packet headers — essential to conduct thorough inspections of network traffic — although it is able to display traffic from large networks at several levels of aggregation, and provides detailed statistics on any hosts.

Similar to our work, IDGraphs [58] only displays two-dimensional views based on time. IDGraphs maps an original TCP-flag-based feature (SYN-SYN/ACK values of complete flows) on the vertical axis and emphasizes several patterns for different kind of attacks. However, due to routing policies, the backbone traffic is usually asymmetric and contains numerous incomplete flows, and therefore, the proposed feature based on the TCP flag is irrelevant for analyzing backbone traffic.

A.3 Design and Features

Our main goal is to provide an interactive tool, to intuitively understand backbone traffic at different temporal or spatial resolutions, and to validate alarms reported by anomaly detectors. Manually validating results obtained from anomaly detectors is a challenging task because of the multi-dimensionality of network traffic and the large amount of data. Thus, we designed the proposed tool to include the following requirements: the tool has to focus on the significant traffic features to show a network traffic behavior and highlight anomalies in a way that is intelligible to users. It should enable the identification of diverse anomalies by exploring traffic at different scales and in various graphical representations, and permits a particular subset of the whole traffic to be analyzed by filtering the entire set of traffic. A precise understanding of the monitored traffic has to be gained by displaying the original header information and accurate graphs from selected plots. Since this tool is interactive, it has to display figures sufficiently fast, and provide them on different platforms. Script languages or interpreted languages have to be avoided for performance reasons. As the tool has to be quickly operational on several files, it needs to read data directly from the dump files and should not use an intermediate database.

A.3.1 Graphical representations

An asset of the proposed tool is its ability to display a large amount of data and highlight unusual behaviors in two-dimensional views that are easily readable. Obviously, three-dimensional views would provide additional information compared to those that are two-dimensional (hereafter respectively called 3-D and 2-D views). However, to observe such 3-D views we have to project them down onto a 2-D visual aid (e.g., a screen or paper). Two main issues are raised by this dimensionality reduction, namely disorientation and occlusion [52]. Disorientation means that the position of the plotted data is not clear and the values corresponding to the plots are difficult to retrieve. Occlusion occurs when plots hide one another, so information is omitted from view. These two problems are well-known in the field of computer vision, and a common solution is to display

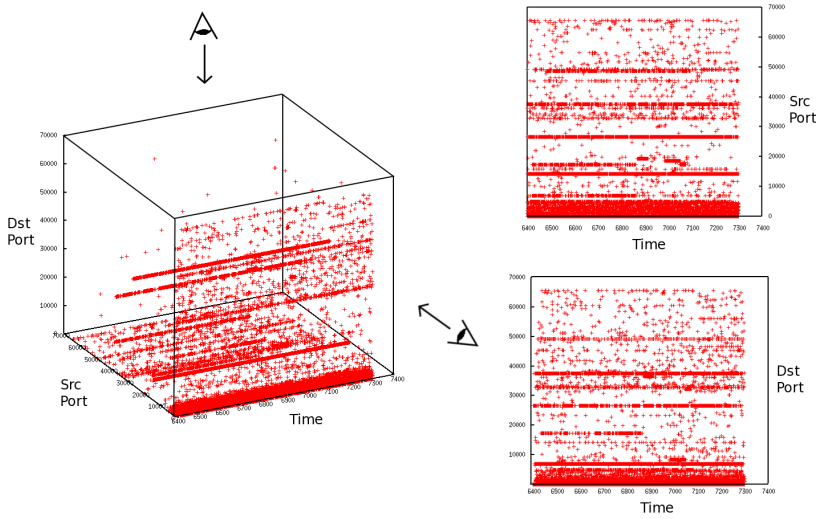


Figure A.1: Hard-to-read three-dimensional view and two projections helping to identify values.

several 2-D projections instead of a single 3-D view.

Figure A.1 shows an example of a 3-D scatter plot representing network traffic. The three dimensions correspond to the timestamp, source port, and destination port. The main advantage of this representation is to present two traffic features and the time in a single view. Nevertheless, the exact position of each point is difficult to determine and confusing. Also, we need to rotate the cube to verify that plots are not hidden in this particular view. The occlusion issue is even more important when more data are displayed. However, by projecting data onto the faces of a cube surrounding traffic, we obtain an accurate 2-D view of the traffic. For example, the two scatter plots on the right-hand side of Fig. A.1 represent the same traffic; the top one is drawn in the function of the source port and time, while the one at the bottom visualizes the traffic with regard to the destination port and time. These sub-figures are more readily understood than the 3-D representation and allow us to accurately identify the ports numbers corresponding to the plots.

The same type of 2-D scatter plot monitors traffic in the proposed tool, displaying understandable views of the traffic even though we have taken five dimensions into consideration (source port, destination port, source address, destination address, and time). In particular, the network traffic is represented in a five-dimensional space and projected onto several 2-D planes, where the horizontal axis always represents the time, but the vertical axis represents the different traffic features. The following constitutes a list of all the possible ways to represent network traffic using the tool; the first four scatter plots use a color convention where a plotted point is green when it stands for a few packets and becomes progressively redder as the number of packets it represents increases. On the other hand, the next three plots are a time series with their own color convention. Another graphical representation is discussed in Section A.3.3 for a small data set.

Destination IP address space This representation exposes anomalies through their targets. It highlights anomalies that aim at many hosts, or anomalies generating a lot of traffic to a single host/sub-network. The resulting scatter plots have vertical or oblique "lines" (consecutively aligned dots) for anomalies, such as remote exploit attacks, and horizontal "lines" for the targets of DoS attacks, or heavy hitters.

Destination port number This representation emphasizes services targeted in the observed traffic. Obviously, busy services and port scans are highlighted and respectively occur as horizontal and oblique "lines".

Source IP address space This representation highlights the origins of the traffic. Anomalies generating heavy traffic from a single host appear as a horizontal line in the resulting scatter plots. Also, this representation emphasizes various traffics initialized at the same time as DDoS, botnet, or flash crowd.

Source port number This representation reveals the port used by the hosts to communicate. Anomalies based on flooding create as many connections as possible using an increasing source port number. This is translated here as vertical or oblique "lines". This graphical representation is helpful for exposing various kinds of DoSs and remote exploit attacks.

Number of packets Here, the displayed figures are the time series of the number of packets transmitted for each protocol. A red time series is derived for TCP packets, a blue one for UDP, a green one for ICMP, and a black one for other protocols. This representation highlights the misuse of a protocol. For example, a flood generates a considerable number of packets using a particular protocol, easily identifiable as a significant variation in the time series.

Number of bytes Several anomalies cause abnormal variations in the number of bytes. These processes that consume bandwidth are highlighted in this representation as significant variations in the time series.

Average packet size As described by Bardford et al. [9], the average packet size can be taken into consideration to detect anomalies. This reveals the abuse of a particular application, as applications usually use the same packet size for all communications they carry out. This representation is a time series of the average packet size, where anomalies are emphasized by abnormal variations.

A.3.2 Tool overview

Figure A.2 is an overview of our tool, which is composed of three panels, a small one (W0) with a menu bar and an overview of the traffic, and two larger ones (W1 and W2) displaying the traffic in detail. Since our tool displays only 2-D graphical representation based on a single traffic feature, the two detailed panels (W1 and W2 in Fig. A.2) allow the monitoring of two traffic features simultaneously. Users choose which representation has to be displayed in each panel (available representations are listed in Section A.3.1). Thus, our tool avoids the confusion caused by irrelevant information and focuses on the anomalies as

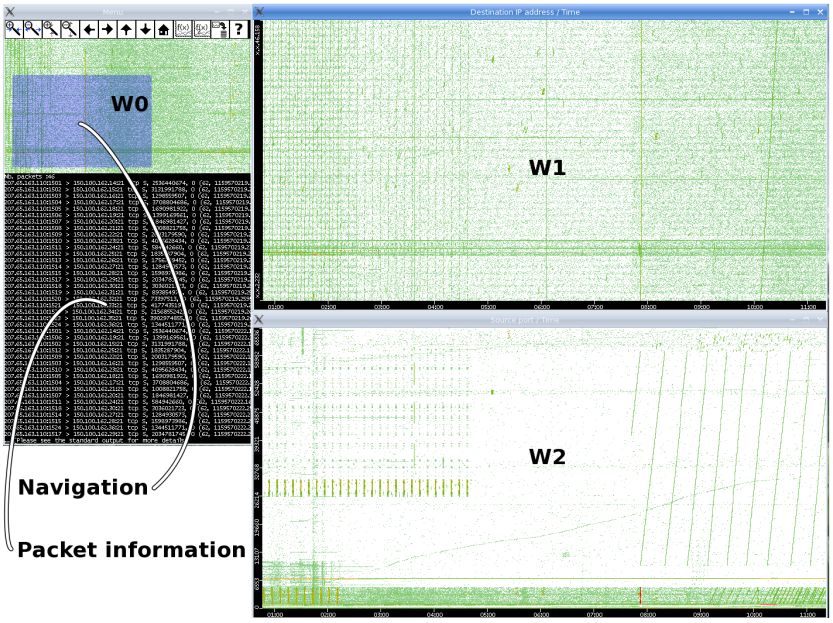


Figure A.2: Tool overview.

they are generally revealed through unusual uses of one or two traffic features [45]. For example, a network scan can easily be identified by analyzing only the destination address and destination port.

Sections A.3.3 and A.3.3 explain several operations for navigating in W1. Depending on these operations, W2 is automatically updated, providing more information about the traffic displayed in W1 as W2 displays only the packets shown in the W1 view. For example, W1 in Fig. A.2 displays a scatter plot of the destination addresses, whereas W2 displays a scatter plot of the source ports. When W1 is zoomed to select a particular sub-network, W2 only presents packets for this sub-network. In W0, the blue rectangle (labeled "Navigation" in Fig. A.2) helps us to figure out where the detailed view is located in the entire traffic. W0 also provides a packet header that corresponds to certain points selected by the user.

A.3.3 Other features

Multi-scale

Anomalies appear at different temporal and spatial scales. Namely, they can last for short or long periods (from an order of seconds to several hours), and they can aim at a single or multiple targets, on one or several ports. The proposed tool allows to zoom in/out independently on each axis. The length of time and feature space (e.g. address space) can be adjusted at any time. This is easily achieved with the mouse wheel, or corresponding buttons. Thus, when long and short-term anomalies are observed, their time duration and their impact in the feature space can easily be estimated.

Easy navigation

Inspecting network traffic and thoroughly investigating anomalous traffic requires movement along the traffic trace and a focus on a particular region. The proposed tool lets users conveniently navigate through the analyzed traffic. Only a click on a particular point is required to center the view on that zone.

Packet information

Characterizing anomalies is a complicated task, as some of them are only identifiable by inspecting the flags of the packet header. A combination of graphical and textual information is essential for identifying anomalies. Our tool helps users in their investigations by providing useful information about all the plotted pixels. A right click on a point in a figure brings up a zoomed view of the clicked zone, and a particular point can be selected to check the corresponding packets headers, and thus we can learn more about the displayed traffic. The tool also represents the selected data as a graphlet that is similar to those presented in BLINC [39]. These graphlets (or parallel coordinates [35]) allow us to simultaneously visualize more than two dimensions, and intuitively highlight communication patterns. The tool takes advantage of this graphical representation to display only small data sets pointed at by the user (graphlets representing large data sets are too confusing).

Input

The tool has to quickly display figures from several input files. Although it would be easier to access data, copying files into an intermediate database is too costly for analyzing daily backbone traffic. Instead of using a database, the tool reads directly from the dump files, like those produced by tcpdump. Also, the tool is able to directly read from compressed files (commonly used to save disk space). Moreover, several files can be given as inputs, and hence, the resulting figures are drawn as all the corresponding files are merged.

Anomaly description

Reports from anomaly detectors are passed on to the tool in the form of admd files[1], which is a XML schema allowing the annotation of traffic in an easy and flexible way. Thus, anomalies reported by anomaly detectors are quickly identified and inspected as they are displayed in black in all the scatter plots.

Portability

Our tool is designed for users utilizing different platforms. We avoided script and interpreted languages for performance purposes, and implemented this application in C++ using only portable libraries to make it available to most users (e.g. views are displayed with the CImg library [1]). Thus, the tool can currently be compiled and executed on different platforms: Unix (Linux and BSD), MacOS, and Windows.

[1] Meta-data format and associated tools for the analysis of pcap data: http://admd.sourceforge.net

Table A.1: Gain in performance due to mechanism for seeking in pcap files.

	User CPU time (clock ticks)	System CPU time (clock ticks)	Time elapsed (minutes:secs)
With "seek structure"	6.00	0.64	00:23.28
Without "seek structure"	10.25	1.43	00:58.42

Option

The tool is customizable through the command line interface to better fit the needs of the users. One important option from among the many options available permits to filter displayed traffic, thus, the tool monitors only certain sub-traffic from the entire traffic trace. Filters have the same syntax as pcap's filters (the same as those used in tcpdump) and are based on any field of the packet header. They allow specific sub-traffic to be accurately selected. For example, this option helps investigations into anomalous traffic by displaying only traffic from a suspicious host on certain ports, or by only selecting SYN packets to highlight the probing processes and SYN flood.

Snapshot

Saving pictures of traces previously observed is essential for visually comparing or illustrating traffic behaviors. Users can save a snapshot of a particular figure at any time. The tool can also be used to generate a batch of visualizations from a set of files with the command line interface. For example, visualizations of daily figures from a year of traces can be generated and stored using only one command line[2].

A.4 Results

A.4.1 Performance

The comfort of navigation and inspection of traffic with our tool is strongly related to its performance and reactivity to one's actions. Since the tool directly reads pcap files, some performance issues are addressed. The main problem is that libpcap does not offer the possibility to directly access a subset of packets corresponding to a given time interval. In practice, the whole traffic trace has to be scanned consuming substantial resources for large traffic traces. Therefore, we consider a dump file to be several parts of the same duration where the first packets of these time slices are called "key packets". Our implementation consists of a data structure that retains information on the "key packets", such as their timestamps and their offsets in the trace file. This data structure helps us to directly access a "key packet" regarding its timestamp. Thus, "key packets" are used as indexes in order to quickly go through the traffic trace. For example, to read a packet at a particular time, t0, the data structure helps us to jump to the "key packet" preceding t0, thereby avoiding having to read numerous unwanted packets prior to this "key packet". Table A.1 lists the gains in performance we obtained with this improvement. The numbers in this table

[2] An example resulting from this feature is the website MAWIviz illustrating all traffic traces of the MAWI archive [14]: http://www.fukuda-lab.org/~romain/MAWIviz/

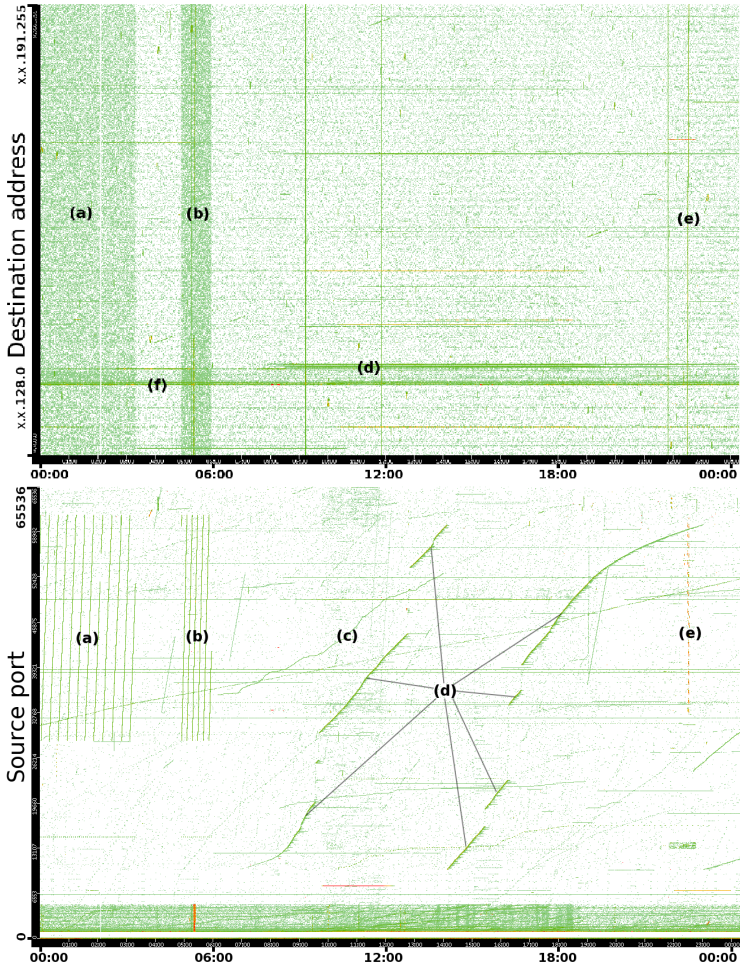


Figure A.3: Scatter plots representing darknet data.

represent the average results from five executions of the same scenario. The scenario consisted of five consecutive zooms in the time space on an uncompressed trace of about 800 MB. The measurements were done on a Linux system with the *time* command, using a computer with 2 GB of RAM and an Intel Core 2 Duo CPU operating at 2.6 GHz. This improvement makes for a comfortable multi-scale navigation through large traffic traces.

A.4.2 Darknet data

Figure A.3 shows an example of the scatter plots generated from darknet traces taken from a /18 sub-network. As described by Pang et al. [57], darknet (or background radiation) is a type of nonproductive traffic sent to unused address spaces. Darknet data are usually analyzed to characterize anomalies and useful for demonstrating the efficiency of our tool. The vertical axis in the first panel of Fig. A.3 stands for the destination addresses, whereas this axis represents the source port numbers in the second panel.

The vertical "lines" in the first panel represent the exploited attacks or any processes using network scans (e.g., (e)). The horizontal "lines" stand for the hosts or sub-networks under heavy attack. They could be the targets of any flood attacks or misconfigurations (e.g., (d) and (f) in the figure).

Other kinds of anomalies are observed in the second panel, and more information about those found in the previous scatter plot are available. Here the vertical "lines" or oblique "lines" represent any procedure using an increasing number of source ports. This is the case in most operating systems when a process opens as many connections as possible. The horizontal "lines" in this panel indicates the constant and heavy traffic from a single port, emphasizing floods, misconfigurations, or heavy-hitters. We can see two sets of consecutive vertical "lines" ((a) and (b) in Fig. A.3) appearing at the same time as sudden heavy noise in the first panel. These two behaviors are interpreted as a process trying to access many of the computers of a sub-network within a short time duration (e.g. exploit or worm) as possible. Checking the headers information revealed that all these packets are directed to port 445. Windows has vulnerabilities in its protocols using this port and several worms have spread through these vulnerabilities. The vertical "line" (e) depicts the same behavior, but within a shorter time frame. Indeed, the packet header information emphasizes an exploit on ssh. We also analyzed the oblique curves (see (c) an (d) in Fig. A.3) and detected attacks aimed at services sensitive to attacks. These attacks are not linear because of the variations in time processing or network delays (due to another activity (d) has some jumps in its source port numbers). The ports concerned are 80 for (c) and 161 for (d). These services are the targets of well-known attacks driving DoS or buffer overflows. (d) aims at a small sub-network (see (d) in the first panel), whereas (c) is aimed at a single target easily identifiable by zooming in on (f).

A.4.3 Network traffic from trans-Pacific link

As an example of anomalies surrounded by legitimate traffic, we analyzed a traffic trace from the MAWI archive [14], which is a set of traffic traces that has been collected by the WIDE Project from 1999. This archive provides large-scale traces taken from trans-Pacific links. The traffic traces are in pcap form without any payload data with both addresses anonymized. Also, the time duration of each trace is fifteen minutes.

Figure A.4 depicts views from ten consecutive files of the MAWI database. The total size of these ten files is about 7.6 GB, for a time of 2.5 h and more than 22 million packets. The vertical axis in the first panel stands for source ports. We can easily see that traffic is heavier than in the example presented in previous section. However, we can still distinguish several red "lines" highlighting some intensive uses of network resources. In the following, we focus on the right part of this figure. Consequently, the next scatter plot results from zooming in on the time axis.

The second panel has also been drawn in regard to source ports. Header information helps us to understand plotted pixels; the two oblique "lines" crossing the figure (see (a) in Fig. A.4) represent a SYN flood. This is an attack from a single host to several targets, the attacker floods targets on port 443 (usually used for HTTP over SSL). This method is well known and results in buffer overflows in the Private Communications Transport (PCT) protocol im-

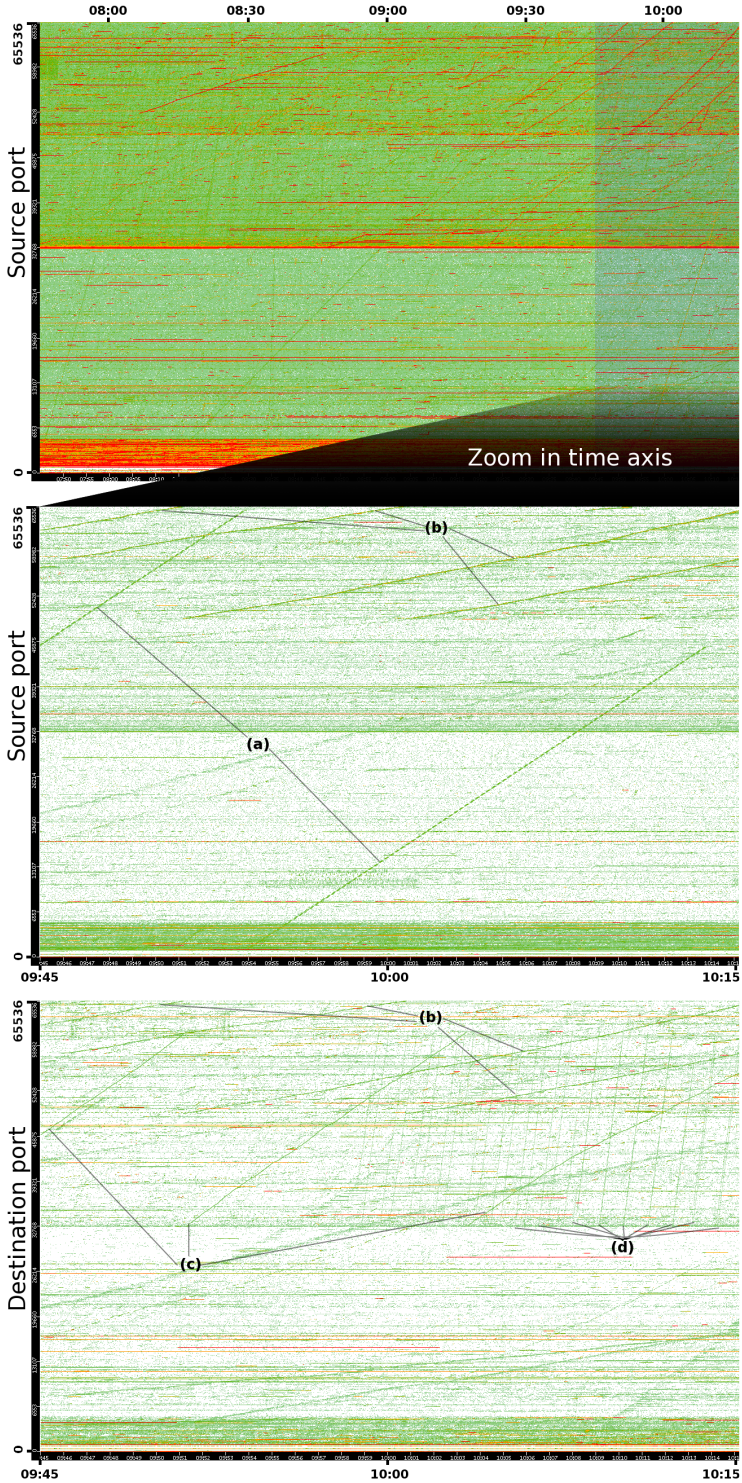


Figure A.4: Samplepoint-F from MAWI Working Group Traffic Archive, 2007/01/09.

plementation in the Microsoft SSL library. The other oblique "lines" represent the same kinds of attacks against other services and from different hosts. In particular, (b) stands for a DDoS attack against a few HTTP servers. The horizontal red "lines" are anomalies consuming bandwidth as in DoS attacks, misconfiguration or heavy-hitters from peer-to-peer networks.

The last panel in Fig. A.4 shows the same traffic but in regard to the destination ports. Similar "lines" to those found in the previous panel (b) appear. They stand for the server's reactions to the DDoS attacks previously observed. Also, two kinds of "lines" repeated several times (see (c) and (d)) are highlighted. Both of these are DoS attacks of ACK packets from two distinct hosts against different targets.

A.4.4 Manual inspection

Inspecting a specific anomaly

The tool helps in inspecting a particular sub-traffic by filtering the entire data before plotting it. The given filters are similar to those in tcpdump allowing for a powerful data extraction. Using filters, the tool is also useful for creating the visualizations of reported anomalies providing additional information in anomaly detector reports.

For example, an anomaly detector [23] reported anomalous traffic on port 515. As this is not a typical target for attacks, we investigated the traffic related to this port. We monitored only the traffic for port 515 (Fig. A.5) with the filtering option of our tool. The upper panel of Fig. A.5 depicts two different traffic behaviors; the left-hand side of the scatter plot shows many short communications dispersed over numerous destination hosts, whereas, the right-hand side of the scatter plot displays longer communications concentrated on a few hosts. This can be interpreted as an attacker probing sub-networks to identify hosts with specific security holes, and a few connections are established to compromise detected victims. The bottom part of Fig. A.5 represents the average packet size corresponding to the traffic displayed in the scatter plot. This time series also exhibits two different phases; it clearly indicates that the size of the packets during the first half of the analyzed traffic is abnormally constant while the second half is more typically fluctuating. The average size of packets in the first phase is particularly small due to the lack of packet payload used during the probing process. However, the following communications have packet payloads that considerably increase the average packet size.

The traffic behavior can intuitively be understood from Fig. A.5, but actual information is still needed to confirm this. The tool supplies header information that corresponds to the displayed plots. Textual header information and a corresponding graphlet are obtained by pointing to a particular plot in the graph.

We retrieved information from several of the plots in Fig. A.5 to clearly comprehend the displayed traffic. Figure A.6 shows a graphlet corresponding to the header information from various plots selected from the first half of the analyzed traffic. The structure of the graphlet is more interesting than the exact values of the IP addresses or port numbers. It clearly indicates that one host using many ports probes numerous hosts on the same port. The textual data reveals that all packets had a SYN flag set, and confirms that the plotted traffic

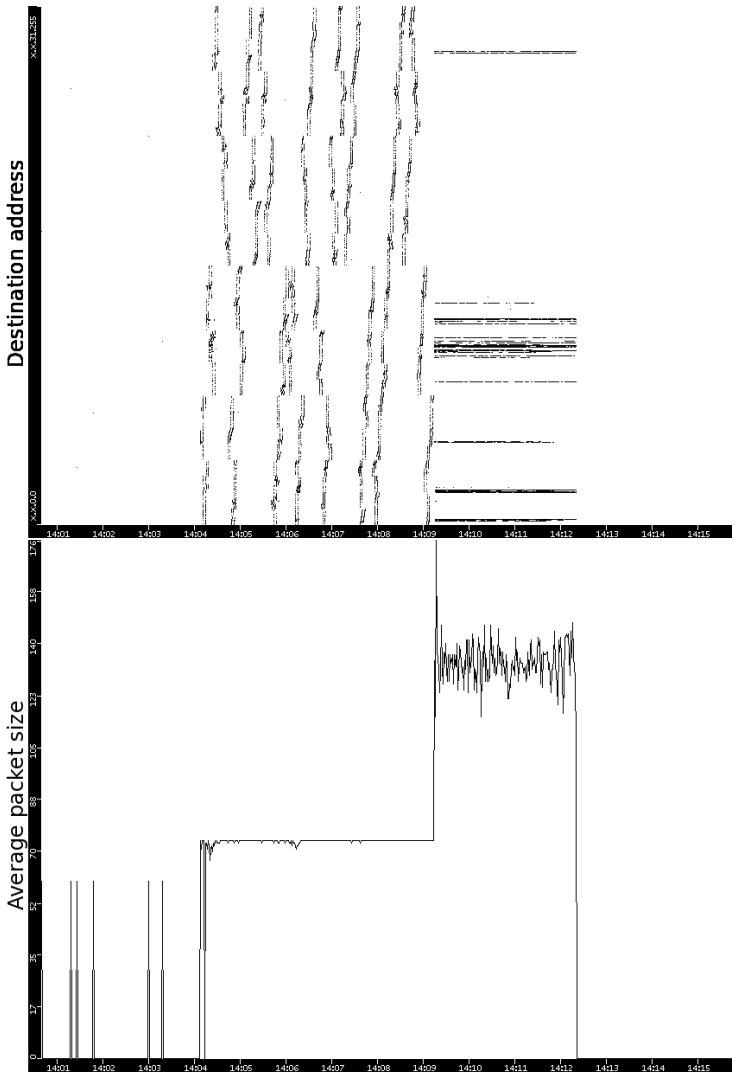


Figure A.5: Exploit on port 515. Top: destination address vs. time. Bottom: average packet size vs. time (MAWI archive, 2001/04/14).

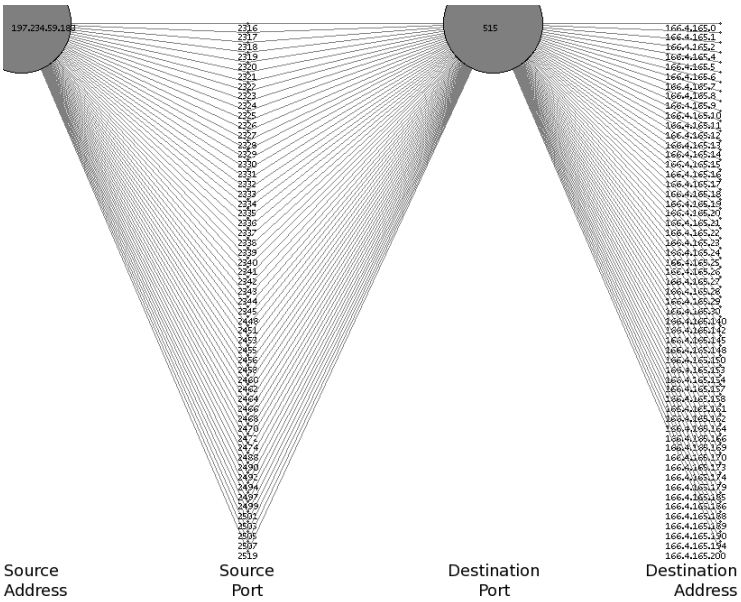


Figure A.6: Header information corresponding to several pixels representing traffic from MAWI archive (2004/10/14).

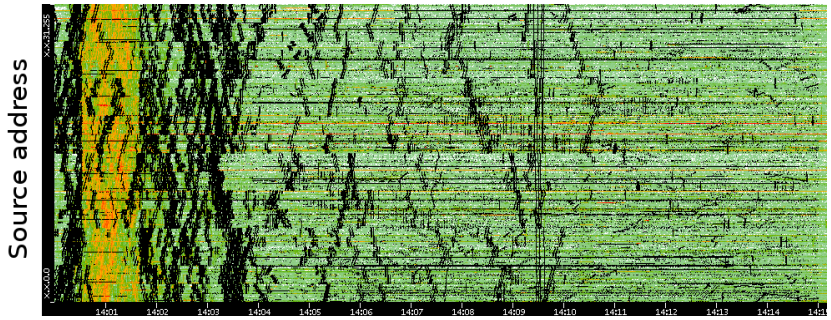
corresponds to a probing process.

Inspecting outputs from anomaly detectors

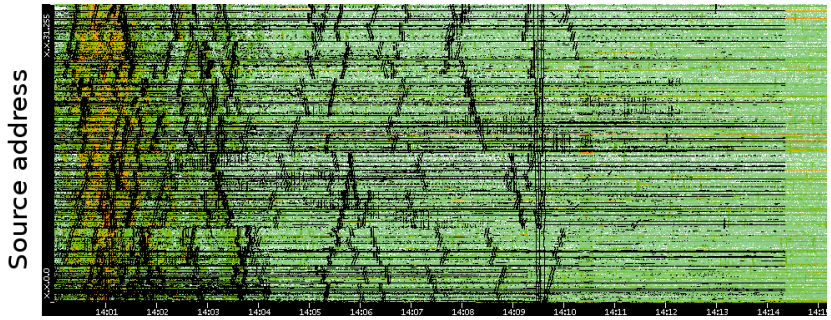
The proposed tool provides valuable assistance to understand and evaluate anomaly detection methods by displaying their results at any temporal and spatial scales in various views. Indeed, by passing the anomaly detector results and original traffic to the tool, it monitors the reported anomalies and helps in rapidly validating them. Thus, researchers designing anomaly detectors are able to validate at a glance the traffic reported by their anomaly detectors and thoroughly inspect anomalies by retrieving anomalous packet header information.

Two examples of anomalies reported by two distinct anomaly detectors are depicted in Figure A.7, where the anomalous traffics are displayed in black. The two anomaly detectors analyzed a MAWI traffic trace in which the first quarter of the traffic is strongly altered by the spreading of the Sasser worm (see the main peak in Fig. A.7(c)). The upper scatter plot (see Fig. A.7(a)) depicts 337 anomalies reported by an anomaly detector based on image processing [23]. This view exhibits the inability of this anomaly detector (with the specified parameter set) to detect all Sasser activities during the main outbreak of the worm. This case emphasizes the valuable support provided by the tool as this fact could not be deduced by only inspecting the textual results outputted by the anomaly detector.

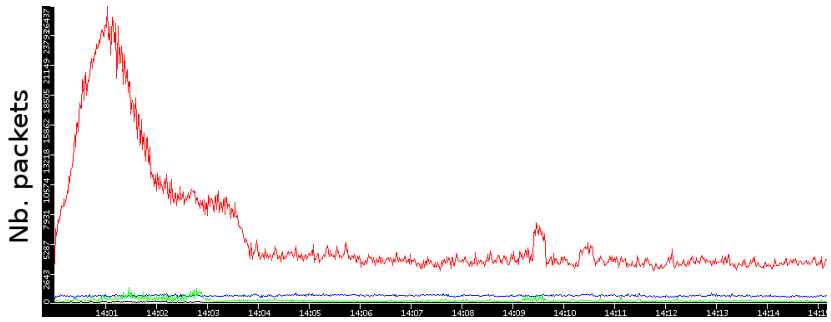
The middle scatter plot depicts 332 anomalies obtained with another anomaly detector based on multi-scale gamma modeling [16]. A quick visual comparison of the two views (Fig. A.7(a) and Fig. A.7(b)) indicates that these two anomaly detectors identified many distinct traffics — particularly during the



(a) Anomalies reported by anomaly detector based on Hough transform.



(b) Anomalies reported by anomaly detector based on gamma modeling.



(c) Number of packets of analyzed traffic trace.

Figure A.7: Highlighting anomalies reported by anomaly detectors in a traffic trace from MAWI archive (2004/08/01) altered by Sasser worm.

peak identified in the first quarter of the trace — although they reported a similar amount of anomalies. This comparison is quickly derived from the two views provided by the tool, whereas, similar conclusions are usually deduced from a time-consuming manual analysis of the two anomaly detectors outputs.

A.4.5 Temporal-Spatial patterns in anomalous traffic

During our experiments we observed particular patterns that stood for certain kinds of anomalies. These patterns exhibit some important properties of the anomalies such as its range of targets and sources, its operational speed, and its time duration. It also provides certain information on the mechanisms used by the anomalies, particularly the uses of the source ports.

Coarse view

At large scales certain anomalies are easily identified as sudden changes in the main traffic behavior or in the usage of a particular protocol. For example, Figure A.8 displays three months of darknet traffic recorded while the first two versions of the Conficker worm were released. This figure shows that first, a sharp increase in the number of source IP addresses and number of packets clearly signaling the start of the worm spread (labeled *Conficker.A* in Fig. A.8). Second, another growth of these quantities depicts the release of the second version of the worm and its aggressive behavior in terms of the network resources consumption (labeled *Conficker.B* in Fig. A.8). The scatter plot of the destination port (see middle scatter plot of Fig. A.8) reveals that the first version of the worm is communicating with the other hosts using random port numbers ranging over $(1024, 5120)$. These types of communications disappear after the second release is unveiled, highlighting that different mechanisms are implemented in this new version.

Fine view

On smaller scales, we observe other kinds of patterns exhibiting anomalies through their abnormal uses of the traffic features. We emphasize that these patterns are in accordance with those identified by an anomaly detector based on pattern recognition [23]. For example, Figure A.9 is composed of different anomalies observed on the same day (2004/10/14). The vertical axis represents the destination addresses for scatter plots at the top of the figure and source ports for those at the bottom. Three different anomalies are emphasized in this figure.

The two representations ((A) and (B)) on the left-hand side of Fig. A.9 stand for an exploit against a Windows service operating on port 445. These were obtained by displaying only the traffic related to a specific IP address, X. The upper representation (A) shows long vertical lines meaning that X contacted numerous hosts within three short periods of time. The header information revealed that all the packets corresponding to these connections were directed to port 445 with the TCP SYN flag set. The representation of the source port (B) indicates that the traffic was initiated from a limited pool of high number ports (< 1024). This traffic is clearly malicious and corresponds to a probing process looking quickly for victims.

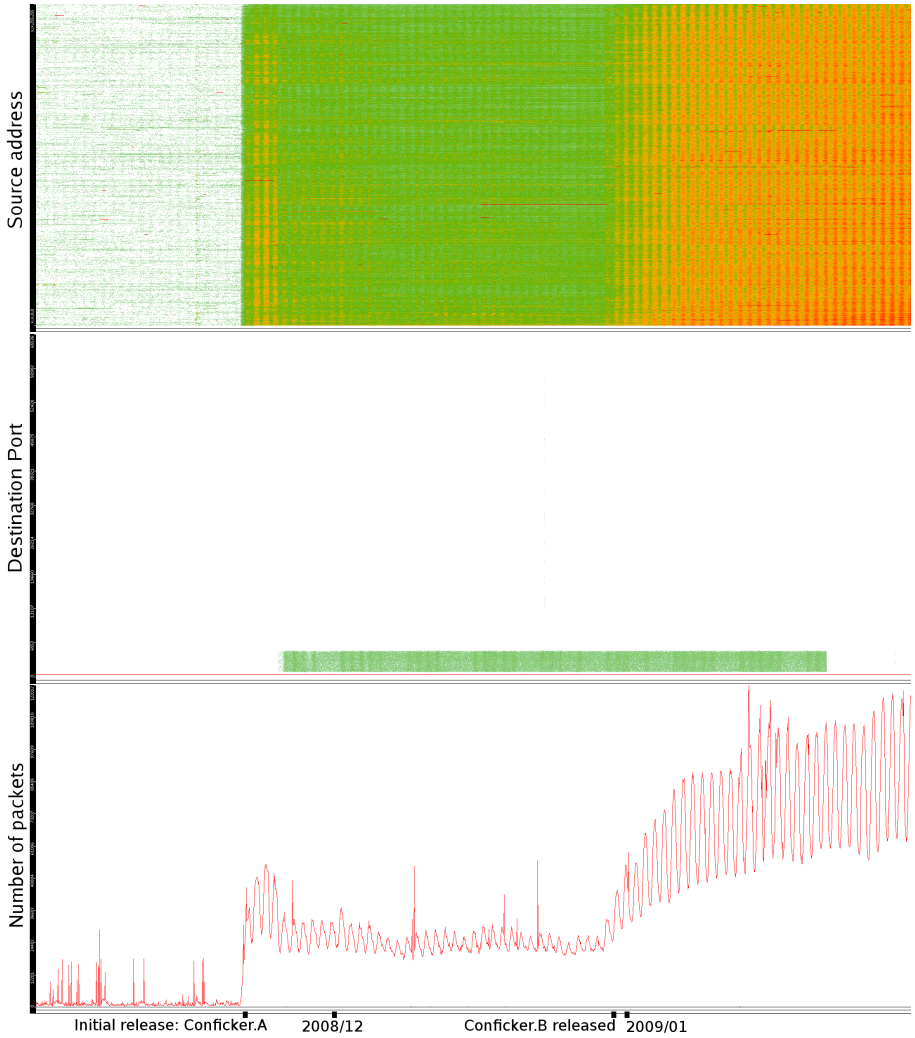


Figure A.8: Three months (2008/12, 2009/01-02) of darknet traffic related to port 445 during Conficker outbreak.

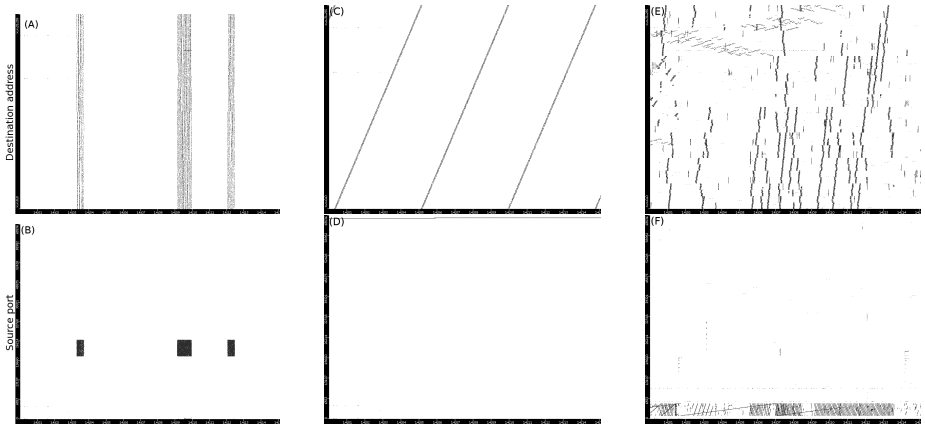


Figure A.9: Different patterns observed in same traffic trace (MAWI archive, 2004/10/14). Top: destination IP vs. time, Bottom: source port vs. time.

The two scatter plots labeled (C) and (D) in Fig. A.9 stand for network traffics from a single host lasting for the entire traffic trace. The upper scatter plot displays long oblique lines, meaning that this traffic also correspond to a probing process. However, the inclination of the lines indicates a slower process than the one previously discussed. Moreover, the lower scatter plot (labeled (D)) shows a horizontal line representing only a couple of source ports.

The two representations, (E) and (F), on the right-hand side of Fig. A.9 correspond to a spreading of the Sasser worm. Traffic from different hosts are displayed in these figures. The vertical structures in the upper scatter plot represent the probing procedure done by the worm, and we noticed that different spreading are observed. The scatter plot representing the source ports (labeled (F)) indicates that this implementation of the Sasser worm generates traffic with only low source ports numbers that are linearly increasing. The shape and height of the observed "lines" provides a signature for this variant of the worm that can be easily identified in other traffic traces.

A.5 Summary

We outlined the need for understanding the network traffic behavior and evaluating anomaly detectors. To achieve these purposes, we designed and implemented a tool graphically representing the network traffic on any temporal and spatial scales. The main contribution of this tool is to display global and detailed views of the network traffic focusing on anomalies. Interesting traffic behaviors are uncovered by interactively exploring the traffic traces, and detailed information is also provided to enable data to be thoroughly investigated. Traffic from specific hosts or services is extracted by using a filtering mechanism. Thus, particular types of sub-traffics are displayed without surrounding noise and can easily be investigated. Furthermore, anomalies reported by anomaly detectors are highlighted in full view and their validation can then be facilitated. The tool runs on different platforms and is freely downloadable[3]. We verified the usefulness

[3]The tool is available at http://www.fukuda-lab.org/~romain/mulot

of our tool by evaluating it on several traffic traces; darknet traces highlighting several patterns for different anomalies, and traces taken from a backbone link where anomalies surrounded by heavy noise were still identifiable. Observation of recent threats, such as the Conficker worm, can also be carried out. We conducted manual inspections of the alarms reported by an anomaly detector and visually compared the outputs of two distinct approaches. Also, we listed several patterns standing for distinct anomalies and noticed that they are consistent with those found in [23].

Bibliography

[1] The c++ template image processing library. (Cited on page 97.)

[2] CoralReef. *http://www.caida.org/tools/measurement/coralreef/*. (Cited on pages 20 and 51.)

[3] MAWILab. *http://www.fukuda-lab.org/mawilab/*. (Cited on pages 76 and 78.)

[4] Tcpdump and libpcap, http://www.tcpdump.org/. (Cited on page 92.)

[5] *Feature Extraction & Image Processing, Second Edition*. Academic Press, 2 edition, Jan. 2008. (Cited on page 29.)

[6] P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, 1998. (Cited on page 11.)

[7] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94*, pages 487–499, 1994. (Cited on page 66.)

[8] A. B. Ashfaq, M. Javed, S. A. Khayam, and H. Radha. An information-theoretic combining method for multi-classifier anomaly detection systems. *ICC '10*, page 5, 2010. (Cited on pages 15 and 55.)

[9] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. *IMW '02*, pages 71–82, 2002. (Cited on pages 11, 25, 32, 57, 91 and 95.)

[10] J.-P. Benzécri. *Correspondence Analysis Handbook*. Marcel Dekker, New York, 1992. (Cited on page 63.)

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J.STAT.MECH.*, 2008. (Cited on page 60.)

[12] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho. Seven years and one day: Sketching the evolution of internet traffic. *INFOCOM '09*, pages 711–719, 2009. (Cited on pages 17, 19, 32, 43, 66 and 72.)

[13] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. *IMC '09*, pages 28–34, 2009. (Cited on pages 3, 14, 24, 40, 41, 48, 57 and 64.)

[14] K. Cho, K. Mitsuya, and A. Kato. Traffic data repository at the WIDE project. In *USENIX 2000 Annual Technical Conference: FREENIX Track*, pages 263–270, 2000. (Cited on pages 17, 26, 58, 98 and 100.)

[15] H. chul Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: Myths, caveats, and the best practices. *CoNEXT '08*, 2008. (Cited on pages 17 and 77.)

[16] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. *SIGCOMM LSAD '07*, pages 145–152, 2007. (Cited on pages 3, 5, 13, 17, 22, 25, 32, 35, 40, 48, 57, 91 and 104.)

[17] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972. (Cited on pages 28, 39 and 42.)

[18] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. *VizSec '08*, pages 111–118, 2008. (Cited on page 92.)

[19] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Trans. Netw.*, 9(4):392–403, 2001. (Cited on pages 4, 14 and 58.)

[20] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. MAWILab : Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. *CoNEXT '10*, 2010. (Cited on pages 17 and 55.)

[21] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. Uncovering relations between traffic classifiers and anomaly detectors via graph theory. In *International Workshop on Traffic Monitoring and Analysis (TMA '10)*, pages 101–114, 2010. (Cited on pages 59 and 77.)

[22] R. Fontugne and K. Fukuda. A Hough-transform-based anomaly detector with an adaptive time interval. *ACM SAC '11*, pages 468–474, 2011. (Cited on pages 17, 57 and 66.)

[23] R. Fontugne, Y. Himura, and K. Fukuda. Evaluation of anomaly detection method based on pattern recognition. *IEICE Trans. on Commun.*, E93-B(2):328–335, February 2010. (Cited on pages 40, 43, 91, 92, 102, 104, 106 and 109.)

[24] R. Fontugne, T. Hirotsu, and K. Fukuda. An image processing approach to traffic anomaly detection. *AINTEC '08*, pages 17–26, 2008. (Cited on pages 25 and 26.)

[25] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. (Cited on page 60.)

[26] K. Fukuda. An analysis of longitudinal tcp passive measurements (short paper). *International Workshop on Traffic Monitoring and Analysis (TMA '11)*, pages 29–36, 2011. (Cited on page 17.)

[27] K. Fukuda and R. Fontugne. Estimating speed of scanning activities with a hough transform. *ICC '10*, page 5, 2010. (Cited on pages 17 and 42.)

[28] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Focusing on context in network traffic analysis. *IEEE Comput. Graph. Appl.*, 26(2):72–80, 2006. (Cited on page 92.)

[29] H. Gupta, V. J. Ribeiro, and A. Mahanti. A longitudinal study of small-time scaling behavior of internet traffic. In *Proceedings of NETWORKING 2010*, pages 83–95, 2010. (Cited on page 17.)

[30] A. S. Haakon Ringberg and J. Rexford. Webclass: adding rigor to manual labeling of traffic anomalies. *SIGCOMM CCR*, 38(1):35–38, 2008. (Cited on page 91.)

[31] P. E. Hart. How the Hough transform was invented [DSP History]. *Signal Processing Magazine, IEEE*, 26(6):18–22, Oct. 2009. (Cited on page 28.)

[32] Y. Himura, K. Fukuda, K. Cho, and H. Esaki. An automatic and dynamic parameter tuning of a statistics-based anomaly detection algorithm. *ICC '09*, page 6, 2009. (Cited on pages 1, 3, 39 and 80.)

[33] P. Hough. Method and means for recognizing complex patterns. *U.S. Patent 3,069,654*, 1962. (Cited on page 28.)

[34] J. Illingworth and J. Kittler. A survey of the hough transform. *Comput. Vision Graph. Image Process.*, 44:87–116, August 1988. (Cited on page 28.)

[35] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, V1(4):69–91, 1985. (Cited on page 97.)

[36] B. Irwin and J. P. Riel. Using inetvis to evaluate snort and bro scan detection on a network telescope. *VizSEC '07*, pages 255–273, 2007. (Cited on page 92.)

[37] Y. Kanda, K. Fukuda, and T. Sugawara. An evaluation of anomaly detection based on sketch and PCA. *GLOBECOM '10*, 2010. (Cited on pages 12, 13, 22 and 48.)

[38] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary poisson view of internet traffic. In *INFOCOM '04*, 2004. (Cited on page 17.)

[39] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: multilevel traffic classification in the dark. *SIGCOMM '05*, 35(4):229–240, 2005. (Cited on pages 91 and 97.)

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

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

[42] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: methods, evaluation, and applications. *IMC '03*, pages 234–247, 2003. (Cited on page 22.)

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

[44] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. *SIGCOMM '04*, pages 219–230, 2004. (Cited on pages 12, 14, 22 and 57.)

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

[46] K. Lakkaraju, R. Bearavolu, A. Slagell, W. Yurcik, and S. North. Closing-the-loop in nvisionip: Integrating discovery and search in security visualizations. *VIZSEC '05*, page 9, 2005. (Cited on page 93.)

[47] S. Lau. The spinning cube of potential doom. *Commun. ACM*, 47(6):25–26, 2004. (Cited on page 92.)

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

[49] Y.-s. Lim, H.-c. Kim, J. Jeong, C.-k. Kim, T. T. Kwon, and Y. Choi. Internet traffic classification demystified: on the sources of the discriminative power. *CoNEXT '10*, pages 9:1–9:12, 2010. (Cited on page 17.)

[50] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579 – 595, 2000. (Cited on page 15.)

[51] W. Lu and A. A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP J. Adv. Signal Process*, 2009:4:1–4:16, January 2009. (Cited on page 13.)

[52] R. Marty. *Applied Security Visualization*. Addison-Wesley Professional, 1 pap/cdr edition, 2008. (Cited on page 93.)

[53] J. Mchugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur.*, 3(4):262–294, 2000. (Cited on page 15.)

[54] C. J. Merz. Using correspondence analysis to combine classifiers. *Mach. Learn.*, 36(1-2):33–58, 1999. (Cited on page 63.)

[55] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An empirical evaluation of entropy-based traffic anomaly detection. *IMC '08*, pages 151–156, 2008. (Cited on pages 13, 14 and 57.)

[56] P. Owezarski. A database of anomalous traffic for assessing profile based ids. In *International Workshop on Traffic Monitoring and Analysis (TMA '10)*, pages 59–72, 2010. (Cited on page 15.)

[57] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. *IMC '04*, pages 27–40, 2004. (Cited on page 99.)

[58] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson. Idgraphs: Intrusion detection and analysis using histographs. *VizSEC '05*, 2005. (Cited on page 93.)

[59] H. Ringberg, M. Roughan, and J. Rexford. The need for simulation in evaluating anomaly detectors. *SIGCOMM Comput. Commun. Rev.*, 38(1):55–59, 2008. (Cited on pages 4, 14 and 58.)

[60] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 35(1):109–120, 2007. (Cited on pages 1, 3, 6, 12, 13, 22, 39, 41 and 80.)

[61] A. Rosenfeld. Picture processing by computer. *ACM Comput. Surv.*, 1(3):147–176, 1969. (Cited on page 28.)

[62] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. *IMC '09*, pages 1–14, 2009. (Cited on pages 6, 12, 13, 14, 22, 57, 74 and 80.)

[63] R. Sadoddin and A. A. Ghorbani. A comparative study of unsupervised machine learning and data mining techniques for intrusion detection. *MLDM '07*, pages 404–418, 2007. (Cited on pages 5, 25 and 35.)

[64] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry. Non-Gaussian and Long Memory Statistical Characterisations for Internet Traffic with Anomalies. *IEEE Transaction on Dependable and Secure Computing*, 4(1):56–70, 02 2007. (Cited on pages 14 and 57.)

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

[66] F. Silveira and C. Diot. Urca: pulling out anomalies by their root causes. *INFOCOM'10*, pages 722–730, 2010. (Cited on pages 3 and 41.)

[67] F. Silveira, C. Diot, N. Taft, and R. Govindan. Astute: detecting a different class of traffic anomalies. *SIGCOMM Comput. Commun. Rev.*, 40:267–278, August 2010. (Cited on pages 13, 14 and 55.)

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

[69] A. Soule, K. Salamatian, and N. Taft. Combining filtering and statistical methods for anomaly detection. *IMC '05*, pages 331–344, 2005. (Cited on page 12.)

[70] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. *IEEE international conference on Computational intelligence for security and defense applications (CISDA '09)*, pages 53–58, 2009. (Cited on page 15.)

[71] J.-P. van Riel and B. Irwin. Inetvis, a visual tool for network telescope traffic analysis. *Afrigaph '06*, pages 85–89, 2006. (Cited on page 92.)

[72] K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Trans. Netw.*, 16(6):1241–1252, 2008. (Cited on pages 40 and 64.)