Visual Comparison of Network Anomaly Detectors with Chord Diagrams

Johan Mazel
NII/JFLI
johanmazel@nii.ac.jp

Romain Fontugne
NII/JFLI
romain@hongo.wide.ad.jp

Kensuke Fukuda
NII
kensuke@nii.ac.jp

ABSTRACT

Network anomaly detection is a crucial task in traffic monitoring. During the past years, statistical algorithms have been a popular approach to this end. Network administrators are traditionally the ones that are deploying and maintaining network anomaly detection systems. They thus are in great need of information regarding detectors behaviors. However, network administrators lack techniques to further analyze and understand detection algorithms.

In this paper, we present several new visualization-based analysis methods that provide in-depth detectors results analysis. These methods consequently enable a detailed account of detectors behaviors. We apply our proposal to the four anomaly detectors used in MAWILab, a documentation of anomalies located in real backbone traffic traces from the publicly available MAWI dataset. We use four years of traffic from this particular network traffic repository.

Our analysis shows that: (1) observed detectors exhibit two different behaviors regarding parameter settings, (2) most detectors share a consistent proportion of their results, (3) algorithms contribute differently to the results of a detector combination strategy, and (4) we improve MAWILab ground-truth anomalies through detectors settings tuning by approximately 19 percentage points.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.3.8 [**Computer Graphics**]: Application

General Terms

Measurement, Security, Visualization

Keywords

Network anomaly, Detector analysis, Visualization, Chord diagram

SAC'14 March 24-28, 2014, Gyeongju, Korea.
Copyright 2014 ACM 978-1-4503-2469-4/14/03 ...$15.00.

1. INTRODUCTION

Network traffic anomalies have a detrimental effect on legitimate users access to Internet resources. Identifying anomalous events is a crucial network management task that requires automation. A great deal of attention has been paid to this problem and led to many proposals relying on statistical methods such as wavelet [1], Kalman filters [15], hash projection [3, 5, 10], Principal Component Analysis (PCA) [9, 13], pattern recognition [8]. Due to this variety of theoretical background, these methods potentially exhibit extremely diverse behaviors regarding anomaly characteristics: volume, distributed nature, and so on.

In the network management ecosystem, network administrators are tasked with handling security matters. They are thus the end-users of network anomaly detection algorithms. It is therefore critical for network administrators to understand detectors behaviors in order to make an appropriate use of these systems. Current efforts towards this goal is currently limited to techniques targeting performance evaluation.

Our goal is to help network administrators to extend their understanding of network anomaly detector behaviors. However, this task is excessively complicated. Detector outputs are usually composed of hundreds of thousand of various anomalies. This problem becomes even more complicated when one intends to tackle the analysis of several detectors, either to evaluate their performance or compare their results. To this end, we intend to leverage advanced visualization techniques to help analyze detection results. The space efficiency of visualization pleads for its use in order to handle the great amount of information that needs to be processed. Performance evaluation techniques have already been proposed in the literature. However, up to our knowledge, no existing proposition has been made regarding the direct comparison of network anomaly detection algorithm results. This is promising because it allows one to assess whether a single detector output is similar to a wide consensus among several detectors or, oppositely, detect his own anomalies. Detector comparison also helps algorithm tuning regarding the trade-off between these two opposites, i.e. either being consensual regarding other detection algorithms or reporting a potentially high number of irrelevant anomalies.

The contribution of this paper is a set of methods that compare network anomaly detection algorithms behaviors. These methods rely on visualization techniques, and more precisely chord diagram, to allow easy, intuitive and in-depth understanding of detection results, and thus, detectors be-

Table 1: Network anomaly detectors

Name	Principle/Theoretical background	Reference
KL	Kullback-Leibler divergence & Association rule mining	[3]
Gamma	Hash projection & Gamma distribution	[5]
Hough	Feature extraction (Hough transform)	[8]
PCA	Hash projection & Principal Component Analysis	[9]

haviors.

The paper is structured as follows. Related work are presented in Section 2. The context of this work is exposed in Section 3. Our contribution and findings are described in Section 4. We then discuss our findings in Section 5. Finally, we conclude in Section 6.

2. RELATED WORKS

We distinguish two different research areas that are related to our study. One refers to previous works that study network anomaly detection algorithms. The other one covers visualization techniques aiming at analyzing network anomaly detection results.

Several works aims at understanding existing network ano-maly detectors. In [6], Eestevez-Tapiador et al. provide a survey of existing detectors principles and consequently build a taxonomy. This work however does not provide insights regarding detection results in a real-world context. In [8], Fontugne et al. detail four detectors behaviors while trying to combine their results. They actually provide an embryonic breakdown of detectors results comparison through overlapping analysis in the context of the MAWI repository, a network trace repository. However, the analysis presented in this paper is relatively simple and we think that detector analysis and output analysis can subsequently be greatly improved.

Regarding the use of visualization techniques to improve detector behavior understanding, ROC curves [11] have been widely used. They allow one to visualize the trade-off between detecting many events at the price of extracting irrelevant ones, and only reporting a small number of events and missing anomalous ones. ROC curves are thus able to provide graphical representation of the overall performance of one or several algorithms against ground-truth data. However, these curves only provide indirect comparison between detectors, i.e. a comparison relative to a common reference, here the ground-truth. ROC are unable to perform direct comparison: estimate detector similarities. To the best of our knowledge, this work is the first one to address direct comparison between multiple detection results.

3. COMPARING NETWORK ANOMALY DETECTORS

Detection results analysis is one of the critical task in network anomaly detection. In this work, we especially aim at comparing detectors. Detection results comparison can, among other things, be used to cross-validate detectors, i.e., against the state of the art. It also helps understanding how detectors results overlap. In the next subsections, we will first explain the background of this work, and then, we will discuss the different problems and questions that arise regarding detection results comparison and how to leverage visualization techniques in this context.

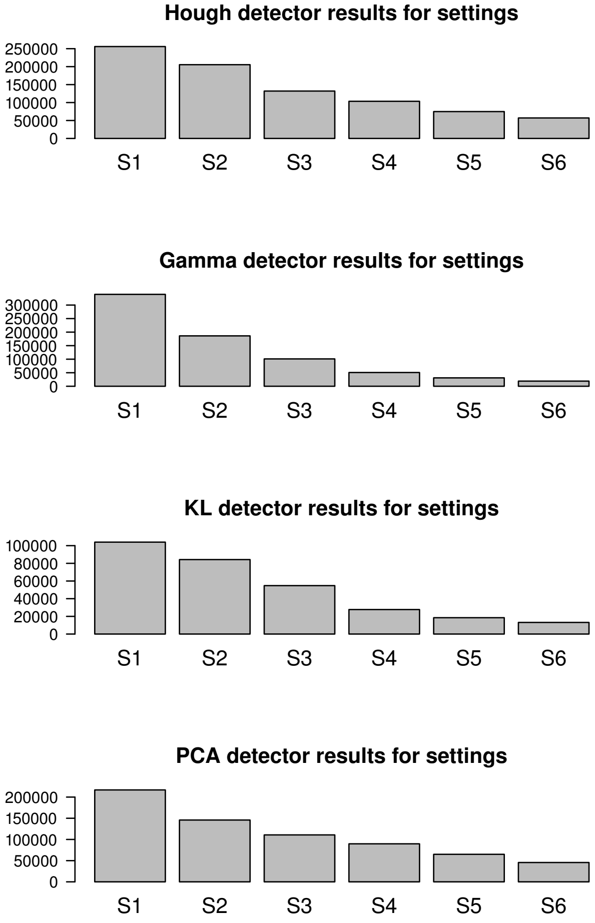


Figure 1: Histograms of anomalies number reported by each setting for each detector.

3.1 Background

The general context of this work is the study of the MAWI repository[1]. MAWI is a public collection of 15 minutes long network traffic traces captured everyday on a backbone link between Japan and the USA since 2001. Upon this repository, the MAWILab project [7] dataset[2] aims at identifying anomalies present in MAWI traces. MAWILab takes advantage of a combination of four anomaly detectors based on different theoretical backgrounds (cf. Table 1). In this article, we study four years of traffic from 2003 to 2006. The main advantages in combining anomaly detectors are that:

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

[2] http://www.fukuda-lab.org/mawilab/

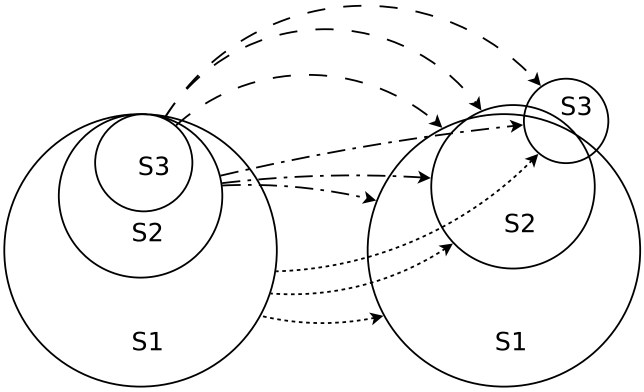


Figure 2: Euler diagrams of alarm sets intersection between three settings ($N = 3$), conservative (S_3), optimal (S_2) and sensitive (S_1) inside two detectors and possible intersections between these settings.

(1) the detectors diversity allows to identify a wide range of anomaly types and (2) the consensus of independent detectors provides reliable results.

MAWILab combines detectors results as follows. First, each detector is run on a traffic trace. A graph-based similarity estimator is then used to systematically uncover the relations between alarms reported by detectors. Several groups of alarms are thus obtained where each group is associated with a single anomaly. Sets of false positive alarms representing legitimate traffic needs to be discerned from sets of true positive alarms standing for anomalous traffic. This task is done using SCANN [14], an unsupervised combination strategy based on correspondence analysis. The final step of SCANN assigns a class membership (normal or anomalous) for each group of similar alarms (i.e. anomaly).

In order to increase the diversity of detection results, SCANN actually uses several settings for each detector. N settings, each noted Si with $i = 1..N$, are thus used. If the number of anomalies for each setting Si is too dissimilar across detectors, SCANN may generate an unbalanced and biased output. To this end, each detector is tuned in order to obtain several (in this case $N = 6$) settings where each setting yields a similar number of anomalies for each detector. The produced results are exposed on Figure 1. This figure shows that three detectors (PCA, Hough and Gamma) indeed provide a similar output in terms of number of alarms (between 210,000 and 340,000 alarms). KL generates around 100,000 alarms. This is a bit far from what other detectors are reporting but it is linked to inherent limitations of this detector. In the context of MAWILab, we want to obtain good overlaps among detectors that would then induce pertinent combination results. However, there is a tuning trade-off between not reporting enough anomalies, and thus, diminishing overlaps sizes, and obtaining large overlaps at the price of detecting too many irrelevant anomalies.

3.2 Problem statement

Regarding the context exposed in the previous section, several questions about detection results arise. We detail these questions along four axes.

3.2.1 *Comparing global results of detectors*

The first axis is the global comparison of the four detectors. Intuitively, some anomalies may be detected by more than one detector. This means that detectors results are overlapping. The obvious question is: how big overlaps are? Is there a "particular" detector with a greater similarity to other(s) detector(s)?

We thus intend to compare detectors results. The canonical visualization technique to achieve this role is Euler diagrams (or Venn diagrams). Euler diagrams can hardly display more than 3 sets [4]. Since we are aiming at comparing 4 detectors, they are unfit for our use case.

3.2.2 *Comparing detection results across settings for each detector*

We then shift our focus to detectors and their settings Si. We are especially interested in the following questions: for a single detector, how do results of different settings overlap? Are conservative setting results completely included in sensitive setting results?

3.2.3 *Comparing detection results between detectors and settings*

We here intend to investigate similarities between settings across the four detectors and answer the following question: how do parameter settings affect the overlap of detectors results?

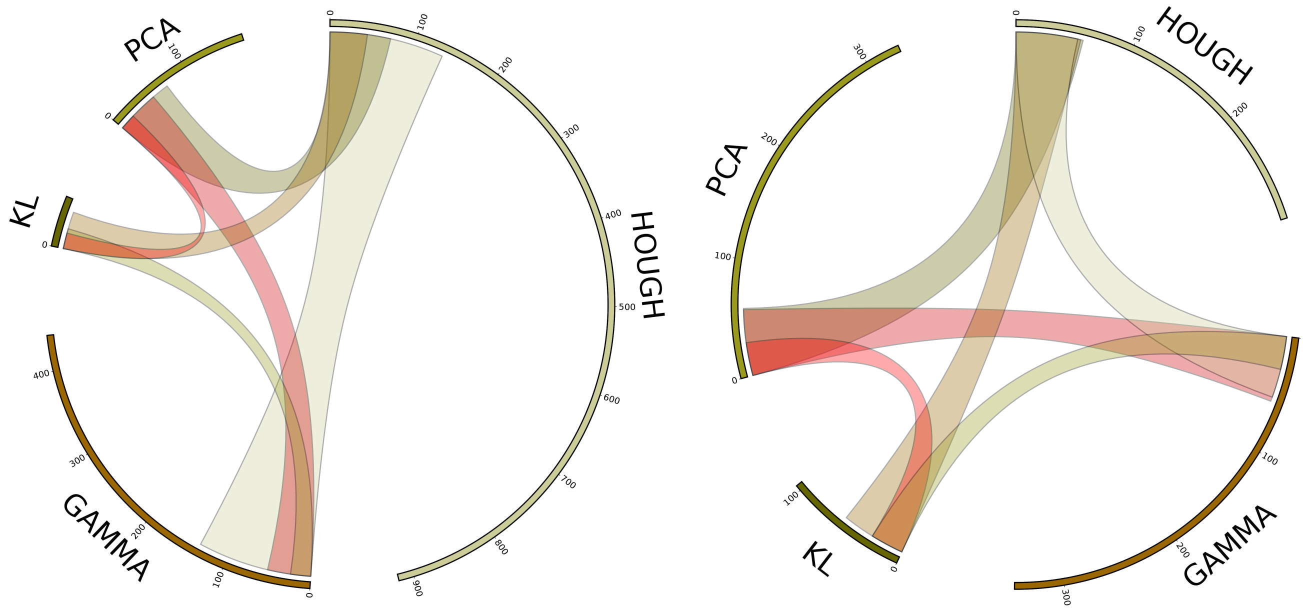
Figure 2 schematically represents detection results as circles and similarities as arrows. Note that here each detector is used with three settings. In this case, we need to compare 9 alarm sets. Yet, this scheme only represents 2 detectors and 3 settings. This means that 4 detectors and 6 settings used in this work will generate a much greater number of elements and make visualization harder. This figure thus further emphasizes the important number of elements to be compared.

3.2.4 *Evaluating detectors contribution in the combination results*

The final aspect that needs to be investigated is the relation between the final result of the combination strategy (cf. Section 3.1) and detectors results. How do detectors contribute to the final results? Is there a similar number of anomalies from each detector in the combination results or are there detectors contributing more than the others?

4. COMPARING DETECTORS OUTPUTS WITH CHORD DIAGRAMS

The previous section presents the context and the questions we intend to answer. The cornerstone of our approach is the comparison of detection results. We actually intend to compare anomaly sets through visualization. We cannot use Euler diagrams because they cannot fulfill the previously exposed requirements in terms of number of elements to be compared (cf. Section 3.2). We therefore leverage chord diagrams built with Circos [12], a genome data visualization tool. This type of diagram does not exhibit the same limitation as Euler diagrams and thus is suitable for our use. A chord diagram is composed of several arcs located on a circle (see also Figure 3). Arcs can be split into several bands if needed. Arcs and bands can be labeled. Elements to be compared can be represented by arcs only or arcs subdivided in bands. Links or ribbons are inside the circle and connect arcs or bands together in order to display similari-



(a) Before detector tuning ($N = 3$)

(b) After detector tuning ($N = 6$)

Figure 3: Intersection of alarm sets between all 4 detectors (unit is equal to 1000 alarms, e.g.: 100 → 100000).

ties. Circos generated figures can be customized regarding many parameters: line thickness, color, link shape.

In our use case, detection results are displayed as arcs. We also use bands in several cases: Figures 4, 5 and 6. The meaning of these bands will be explained on a case-by-case basis. Links or ribbons between arcs or bands represent similarities between detection results. Ribbon width represents the actual number of anomaly detected by both arcs or bands. Furthermore, ribbons are sometimes superimposed but that does *not* mean that ribbons actually intersect between themselves. In other words, intersections of alarm set intersection between detectors are not represented by overlaps between ribbons. This observation is valid for *every* chord diagram in this paper.

Our analysis follows the four axes previously exposed in Section 3.2. We will also study the impact of tuning for the first and fourth axes. The influence of tuning on the second and third axes will not addressed due to the lack of space.

4.1 Comparison between detectors global results

We here investigate anomalies that are common to several detectors. Figure 3 shows the similarity between detectors before (Figure 3a) and after (Figure 3b) tuning as presented in Subsection 3.1.

Figure 3a displays detectors overlaps before tuning. Number of detectors alarms are clearly uneven, as seen through the inconsistent arc lengths. For example, Hough yields twice as much alarms as Gamma, the detector with the second biggest number of alarms. Figure 3b shows a much more balanced number of alarms across detectors. Figure 3b also shows that intersections between detectors are very similar across three detectors: Hough, Gamma and PCA. The KL-based detector exhibits a small intersection with Gamma and PCA. This can be explained by the fact that its number of reported anomaly is much smaller than those of other detectors (this is also visible on Figure 1). KL-based based detector also has a relatively important intersection with Hough. We explain this behavior later. Figure 3 thus constitutes a simple and preliminary way to evaluate the efficiency of the detector tuning step and emphasizes how tuning between detectors impacts the equilibrium between detectors in terms of reported alarms. This figure also clearly shows the influence of tuning on overlaps between detectors. A balanced tuning induces similarity of overlaps across detectors and thus favors efficient combination results.

4.2 Comparison between settings for each detector

We next focus our analysis on results across different settings for each detector as shown on Figure 4. Each detector is displayed as an arc. Inside each arc, bands represent settings (S1 being the most sensitive setting and S6, the most conservative one). Each ribbon represents the intersection between two detection results of a detector. For each setting or band, the blue percentage represents the percentage of overlap with the closest more conservative setting $(i+1)$, e.g. for Hough, 63% for S_2 is $card(S_2 \cap S_3)/card(S_2)$. The red percentage is the percentage of overlap with the closest more sensitive setting $(i-1)$, e.g. for Hough, 91% for S_2 is $card(S_1 \cap S_2)/card(S_2)$.

We observe two classes of detectors. The first class exhibits a behavior that looks intuitive and that is very similar to the principle of matryoshka dolls (or Russian nested dolls). In a set of matryoshka dolls, the smallest doll is included in a bigger one which is in turn located inside a bigger

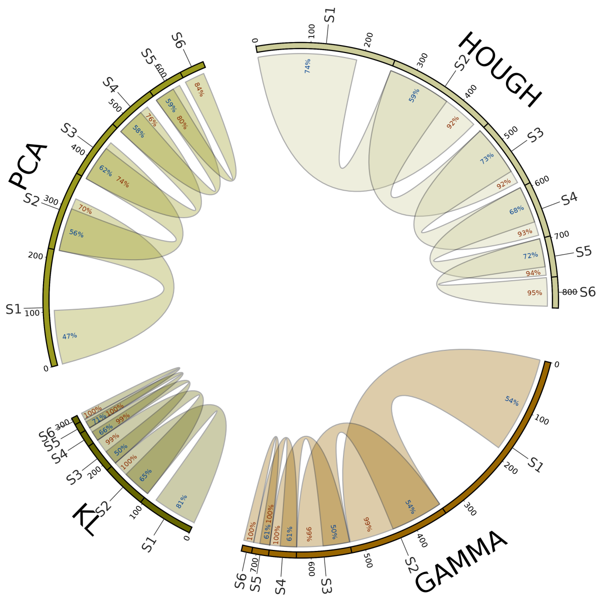


Figure 4: Intersection between settings for each detector (unit is equal to 1000 alarms, e.g.: 100 → 100000). The blue percentage represents the percentage of overlap with the closest more conservative setting $(i+1)$, e.g. for Hough, 63% for S_2 is $card(S_2 \cap S_3)/card(S_2)$. The red percentage is the percentage of overlap with the closest more sensitive setting $(i-1)$, e.g. for Hough, 91% for S_2 is $card(S_1 \cap S_2)/card(S_2)$.

one, and so on. This behavior is similar to the detector on the left-hand side of Figure 2. If we consider alarms reported by two settings (S5 and S6 for example), all alarms reported by the more conservative setting (here S6) are included in the set of alarms detected by the more sensitive setting (here S5). KL and Gamma belong to this class. This behavior is displayed on Figure 4 as a high value for the red percentage (i.e. close to 100%).

For the other class, the Russian doll-like behavior (or successive inclusions) is not present. This means that, counterintuitively, only a fraction of anomalies reported by the more conservative setting (here S6) are also identified by the more sensitive setting (here S5). A detector of this class is shown on the right-hand side of Figure 2. PCA and Hough are members of this class. In this case, red percentage values are much lower: between 92% and 95 % for Hough and between 70% and 84% for PCA. This can be explained by the randomness of some processing step of these algorithms. In fact, since we launch algorithms one time for each setting, randomness can cause some anomaly to be detected by a setting (in our previous example S6) and not by a more sensitive one (S5).

4.3 Comparison between settings across detectors

We then examine detection result similarities between settings across different detectors. The total theoretic number of ribbons to display is 216 (every setting to every other setting for each detector: $\sum_{d=1}^{3} N*(d*N)$). Due to readability concerns, it is clearly impossible to display such a number of ribbons. We thus deliberately only display data for a single detector on each figure and create 4 figures. This choice limits the ribbon number to 108 (every setting from a single detector to every setting for the 3 other detectors: $N*(3*N)$) for each figure.

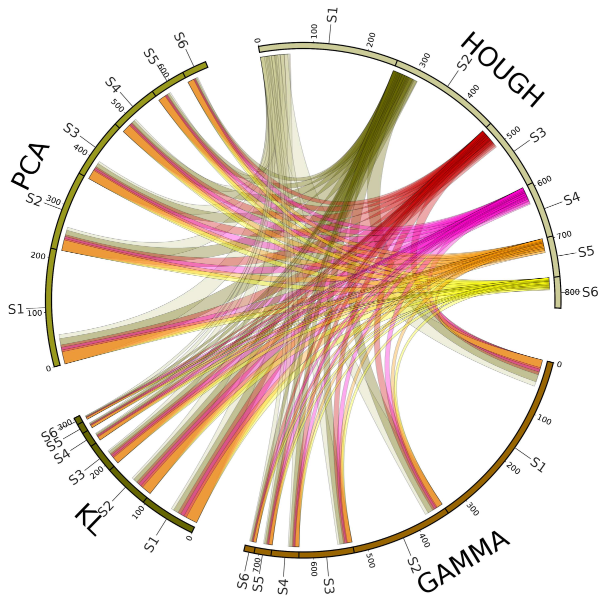


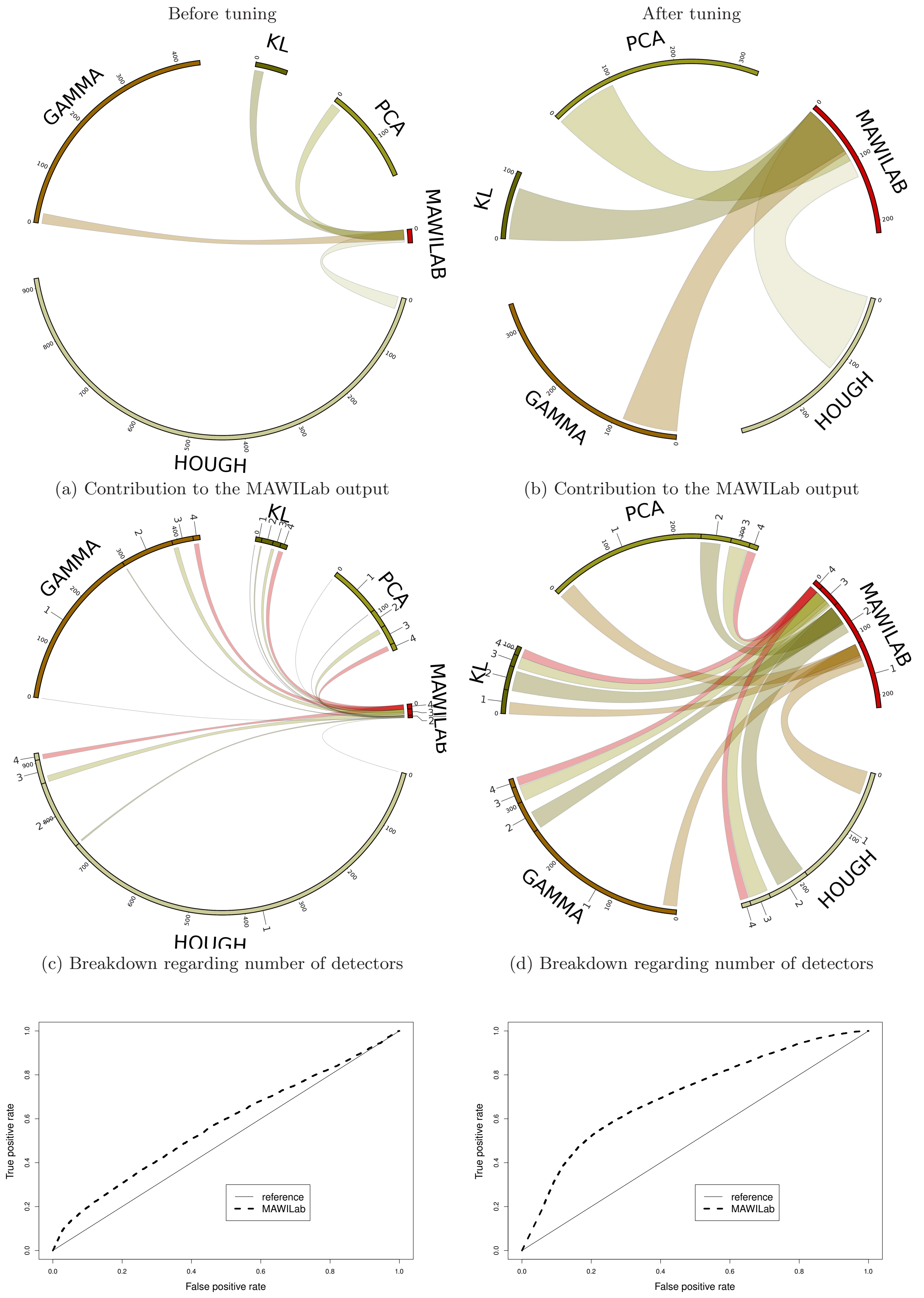
Figure 5: Intersection of alarm sets for each settings of Hough detector with every other detectors settings (unit is equal to 1000 alarms, e.g.: 100 → 100000).

Figure 5 displays similarities between settings across detectors but only for the Hough detector. We observe that the more sensitive a setting is, the bigger the anomaly number is and consequently, the bigger is its intersection with detection result for settings of other detectors. This observation is consistent across settings and detectors. We also note that difference between successive overlaps from or to the same setting are not constant. There may be many reasons for this observation: the same setting will not yield the same number of alarms for two detectors, overlap across settings and detectors may behave in a non-linear way.

4.4 Evaluating detectors contribution in the combination results

We finally investigate the contributions of each detector to the final result obtained with the combination technique exposed in Section 3.1. On Figures 6(a), 6(b), 6(c) and 6(d), all alarms from each detector are displayed on arcs. However, the MAWILab arc only displays events classified as anomalous by the combination method presented in 3.1.

Figure 6(a) (resp. 6(b)) shows the contributions of each detector to the combination results before (resp. after) the tuning exposed in Section 3.1. The shapes of arcs are very similar to the ones of Figure 3 which also displays data before and after tuning. When tuning detectors settings, we also modify the combination strategy results to increase the number of reported anomalies. As a consequence, the number anomalies contributed by each detectors increases after



(a) Contribution to the MAWILab output

(b) Contribution to the MAWILab output

(c) Breakdown regarding number of detectors

(d) Breakdown regarding number of detectors

(e) ROC curve

(f) ROC curve

Figure 6: Analysis of detectors contributions (6(a) and 6(b)), breakdown regarding the number of detectors that detected the anomalies reported by MAWILab (6(c) and 6(d)), and ROC curves (6(e) and 6(f)) before tuning (left column) and after tuning (right column) (unit is equal to 1000 alarms, e.g.: 100 → 100000).

Table 2: Heuristics labeling the traffic corresponding to a set of alarms into three categories ("Attack", "Special", and "Unknown"). These are originated from the anomalies previously reported [2, 8] and the manual inspection of MAWI.

Label	Category	Details
Attack	Sasser	Traffic on ports 1023/tcp, 5554/tcp or 9898/tcp
Attack	RPC	Traffic on port 135/tcp
Attack	SMB	Traffic on port 445/tcp
Attack	Ping	High ICMP traffic
Attack	Other attacks	Traffic with more than 7 packets and: SYN, RST or FIN flag $\geq$ 50% Or, http, ftp, ssh, dns traffic with SYN flag $\geq$ 30%
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

tuning. We note here that the contributions of each detector are much more different across detectors after tuning. If we only consider Figure 6(b), we note that Hough has the biggest overall contributions. PCA and Gamma have the second and third most important contributions to the final results. KL has the smallest contribution in terms of volume. In terms of percentage of anomalous events compared to the total number of alarms, KL exhibits the biggest percentage and can be considered as the most reliable detector. This high percentage may explain the relatively large intersection between KL and Hough observed on Figure 3. PCA and Gamma seem to yield many false positive alarms. Hough is located in between KL and PCA-Gamma.

Figure 6(c) (resp. 6(d)) displays another breakdown for each detector results before (resp. after) the tuning. On these figures, band labels show the number of detectors that commonly found the same anomalous events. For example, an anomaly located on the band "2" of the Hough detector has been detected by two detectors: Hough and another detector. The impact of the number of detectors that commonly found the same anomalous events on MAWILab output is the same on both Figures 6(c) and 6(d): the more detectors detected an event the higher are the odds that this event is classified as anomalous by the combination strategy. However, there is a difference between Figure 6(c) and 6(d). On Figure 6(c), events detected by 3 or 4 detectors (band "3" and "4") represent more than the majority of all anomalous events. This is not the case for Figure 6(d). If we only look at Figure 6(d), we note that the majority of anomalous events in MAWILab is constituted by alarms detected by at least 2 detectors. It is also interesting to note that every detector exhibit an individual contribution to the final results (band "1"). This means that all detectors individually yield pertinent alarms. Figure 6(d) also shows that the combination technique classifies all events detected by all detectors (band "4") as anomalous. Alarms detected by 3 detectors (band "3") are mostly classified as anomalous. The combination method classifies as anomalous a small proportion of alarms detected by 1 or 2 detectors (bands "1" and "2"). By inspecting contributions of each detectors, this figure emphasizes the singularities of this combination strategy over traditional strategies, e.g.: majority voting.

Figures 6(e) and 6(f) expose ROC curves of the combination method presented in Section 3.1. ROC curves display the trade-off between true positive rate (or TPR, i.e. proportion of anomalous events detected as anomalous) and the false positive rate (or FPR, i.e. proportion of normal events detected as anomalous). A perfect curve would exhibit a perfect true positive rate with a null false positive rate. This would be represented on the plot as a step-shaped curve with a point located at the top-left corner. In our case, it is important to note that instances used to generate the TPR and FPR are events detected by detectors, i.e., they are not traffic flows (5-tuple or else) extracted from network traces. This makes performance look worse than it actually is. In fact, if we use traffic flows instead of detected events, TPR stays the same or roughly close but FPR decreases (because traffic flows not flagged by detectors are added to normal instances). The decrease of FPR would then make the results look much better. The groud-truth used here is based on the heuristics presented in Table 2. Detected events labeled as Attack by the table are considered as anomalous. We here compare the two ROC curves: before tuning on Figure 6(e) and after tuning on Figure 6(f). Generally, the ROC curve closest point to the top left corner as the optimal point (the top-left corner being the theoretical perfect point). For Figure 6(e), the optimal point TPR is 56% and its FPR is 45%. For Figure 6(f), the closest point to the top left corner is located at TPR equals to 69% and FPR equals 38%. If we fix FPR at 40% (i.e. between each optimal points), the TPR value before (resp. after) tuning is 51% (resp. 70%). This shows that tuning of the detectors increases by appreciatively 19 percentage points the performance of the combination strategy used in MAWILab.

5. DISCUSSIONS

Circos generated chord diagrams provides a very good visual support. These diagrams greatly help network operators to understand how network anomaly detectors behave. Chord diagrams also exhibit a good scalability: we could easily increase the number of detectors or the number of settings and keep diagrams readable. This is a paramount criteria because we actually intend to increase the number of detectors combined in MAWILab in the near future. All these aspects make chord diagrams far more fit to our use case than Euler diagrams.

As exposed in Section 4.1, Figure 3 provides a way to verify whether algorithms are appropriately tuned. The use of this particular figure is not as precise and informative as Figure 1 for tuning purposes but it constitutes an interesting and much more direct preliminary way to check whether the alarm balance between detectors is good while also comparing detectors results.

On a general note, some of the exposed chord diagrams exhibit a less than optimal readability (for example, Figure 5). However, they are difficult to improve due to the fact that they intend to display a great and hardly reducible amount of information. We therefore intend to keep them as they are presented in this work. It is also interesting to note that, for this particular figure, visualization is an extremely efficient and concise way of representing a great number of similarities. This further motivates the use of visualization.

Ambiguity of overlapping links is also worth discussing. As explained at the beginning of Section 4, overlapping between links does *not* mean that the represented anomaly sets actually have some common elements. This lack of clearness is problematic since it can induce a biased understanding of our figures. However, using a greater number of links to perform a more detailed breakdown would reduce readability. We thus choose to keep a good readability at the expense of expressiveness.

The two classes of behaviors found in section 4.2 are really interesting from the point of network anomaly detection algorithms users. From a practical point of view, once deployed, the class of detectors that exhibit a random behavior may produce inconsistent result. This new fact also constitute an interesting feedback that should help authors of these proposals to improve their work.

6. CONCLUSIONS

We present several techniques to understand network anomaly detection results with the help of visualization. We detail detection results similarities across detectors and settings in a synthetic way. We also analyze the contribution of each detector to the results of a combination strategy in order to assess their individual performance. Visualization is instrumental in enabling easy analysis for our use case. Chord diagrams display a large number of detection results similarities in a compact way. They also exhibit a very good scalability by allowing us to easily increase the number of detectors and settings without affecting readability.

In this work, we make four findings: (1) detectors are divided in two categories: either conservative settings results are completely included in sensitive ones (matryoshka doll-like behavior), as KL and Gamma, or it is not the case, as Hough and PCA, (2) most detectors share a consistent proportion of their results across detectors and settings, (3) detectors do not contribute equally to the final result (absolute value-wise and percentage-wise) and (4) detector tuning helps provide balanced output across detectors and improve overall performance of the combination strategy used in MAWILab by approximately 19 percentage points. All these findings demonstrate the pertinence of the proposed detection results analysis methods.

In the near future, we plan to extend this work and make it available for MAWILab users. We intend to setup a dynamic online version of this analysis with the help of the D3 library[3]. This would allow the user to perform detailed forensics in an intuitive and interactive way. We also envision to allow the user to perform detailed breakdown of anomalies (regarding, for example, anomaly classification information or combination results) and display associated graphical representations. This would greatly improve the embryonic data provided by the MAWILab website. We also intend to extend this study to the whole MAWI repository, i.e. from 2001 to 2013.

7. ACKNOWLEDGMENTS

This research has been supported by the Strategic International Collaborative R&D Promotion Program of the Ministry of Internal Affairs and Communication, Japan, and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 608533 (NECOMA).

[3] http://d3js.org/

8. REFERENCES

[1] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. IMW '02, pages 71–82, 2002.

[2] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM 2009, IEEE*, pages 711 –719, 2009.

[3] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. IMC '09, pages 28–34, 2009.

[4] S. Chow. *Generating and Drawing Area-proportional Euler and Venn Diagrams*. 2007.

[5] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. LSAD '07, pages 145–152, 2007.

[6] J. M. Estévez-Tapiador, P. Garcia-Teodoro, and J. E. Díaz-Verdejo. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, pages 1569–1584, 2004.

[7] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. Co-NEXT '10, pages 1–12, 2010.

[8] R. Fontugne and K. Fukuda. A hough-transform-based anomaly detector with an adaptive time interval. *ACM SIGAPP Applied Computing Review*, pages 41–51, 2011.

[9] Y. Kanda, R. Fontugne, K. Fukuda, and T. Sugawara. Admire: Anomaly detection method using entropy-based pca with three-step sketches. *Computer Communications*, 36(5):575–588, 2013.

[10] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: methods, evaluation, and applications. IMC '03, pages 234–247, 2003.

[11] W. J. Krzanowski and D. J. Hand. *ROC Curves for Continuous Data*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. 2009.

[12] M. I. Krzywinski, J. E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009.

[13] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. SIGCOMM '04, pages 219–230, 2004.

[14] C. J. Merz. Using correspondence analysis to combine classifiers. *Machine Learning*, pages 33–58, 1999.

[15] A. Soule, K. Salamatian, and N. Taft. Combining filtering and statistical methods for anomaly detection. IMC '05, pages 331–344, 2005.