0day anomaly detection made possible thanks to machine learning

Philippe Owezarski, Johan Mazel, and Yann Labit

[1] CNRS; LAAS; 7 Avenue du colonel Roche, F-31077 Toulouse, France
[2] Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France

Abstract. This paper proposes new cognitive algorithms and mechanisms for detecting 0day attacks targeting the Internet and its communication performances and behavior. For this purpose, this work relies on the use of machine learning techniques able to issue autonomously traffic models and new attack signatures when new attacks are detected, characterized and classified as such. The ultimate goal deals with being able to instantaneously deploy new defense strategies when a new 0day attack is encountered, thanks to an autonomous cognitive system. The algorithms and mechanisms are validated through extensive experiments taking advantage of real traffic traces captured on the Renater network as well as on a WIDE transpacific link between Japan and the USA.
Key Words: 0day anomaly detection, machine learning
Acknowledgment — This work is achieved in the framework of the European ECODE project, granted and funded by the European Commission's ICT program under reference FP7-ICT-2007-2/223936.

1 Introduction

Security in the Internet is a very important and strategic problem which raised and still raises significant research and engineering effort, but need to be continuously addressed. The main reason is that the threat in the Internet is moving fast: new kinds of attacks, worms, viruses appear almost every day, they use more and more advanced spreading and corruption strategies, and act so as to remain very hardly detectable. One of the problems then stands in detecting the new attacks (also called 0day - or 0d for short - attacks) the first time they are perpetrated. Current systems are unable of detecting such 0d attacks. When they are first observed, engineers first need to analyze them before searching for a detection and defense strategy, implement it, and finally deploy it. This is a reactive process which lets the network vulnerable for a too long period.

In this paper, we present our first work on designing new cognitive strategies and algorithms for detecting 0day attacks in the Internet. The idea is to design autonomous cognitive systems able to increase autonomously their knowledge database on attacks. As the object under concern in our research is the Internet, we will specifically focus on volume based DoS (Denial of Service) attacks which aim at decreasing network QoS (Quality of Service) and performance level by denying the access to network resources for legitimate users. In networking,

such DoS attacks are part of a broader family of unwanted events called traffic anomalies. We then aim at designing a new cognitive system which is able to autonomously classify anomalies in different categories. The idea is then to give the cognitive system the capability to analyze the anomaly for discovering whether it is legitimate or not, but also to autonomously extend the attack signature database of the related anomaly detection system (ADS) if the new encountered anomaly is classified by the system as an unknown attack. For this purpose, our algorithm relies on the use of machine learning techniques for autonomously issuing models of normal traffic, as well as attack signatures when attacks are encountered for the first time. This signature is then prone to be integrated in an associated defense system (whose description is out of the scope of this paper). This approach allows a significant reduction of the time the network is not protected against a new attack as it takes a short time to issue a new detection signature for classical IDS (Intrusion Detection System) or IPS (Intrusion Protection System) which can be immediately and automatically deployed.

The paper is organized as follows. Section 2 provides an overview on related work. Section 3 presents how the new detection and classification cognitive algorithm works, and justifies our choice of using unsupervised machine learning techniques. In Section 4, the validation data and methodology are presented, as well as the evaluation results. Section 5 then concludes the paper.

2 Related work

There is now a large literature on the detection of network traffic anomalies. Most of the approaches analyze statistical variations of traffic volume (i.e. number of packets, bytes or new flows), traffic attributes (i.e. IP addresses and ports) distributions, or both, on a temporal or spatial manner. The anomalies can be observed from single links or network-wide data. Standard references include [3] [1] [9] [11], with some notable recent work as [13] [5] [4] [16]. Dimensionality reduction of aggregated traffic data has also received recent attention, and techniques like sketches [9] [13] [5] and principal components analysis [11] are very promising for online anomaly detection. Sketches based algorithms can detect low intensity anomalies and can identify the anomalous IP flows (something that might not be possible with techniques that operate only on the aggregated traffic or on origin-destination flows).

In this work, we base our research on the anomaly detection approach presented in [7]. It presents a two steps anomaly detection and classification algorithm that will be presented in section 3.1.

Some work has tried to apply machine learning techniques to anomaly or intrusion detection. Kuang and Zulkernine [10] used a modified KNN algorithm called CSI-KNN for Combined Strangeness and Isolation measure K-Nearest Neighbors. They perform supervised learning on the KDD dataset [8]. They use the feature provided by the dataset to generate two values (strangeness and isolation). These values are then processed to generate a graded confidence over the classification. Some papers push forward the use of machine learning with the

goal of classifying automatically the traffic [12] or to discover new anomalies [6]. In [12], Lakhina et al. use clustering on the entropy of several parameters (IP addresses and ports). This approach groups anomalies with similar characteristics, but does not distinguish between different types of anomalies. Network operators still need to manually check each anomaly, but, if enough pre-labeled anomalies are part of a given cluster, they have a better way to prioritize between clusters than if no classification is done. In [6], Eskin et al. use unsupervised learning to detect new intrusions inside the KDD dataset. However, it remains a work mainly oriented on intrusion detection and it does not consider network anomalies. The authors even consider their work as inoperant in the case of Syn flood DDoS anomalies.

3 Application of machine learning to 0day anomaly detection

3.1 Two steps approach

Because of the limits of both the profile and signature based approaches for detecting attacks and anomalies, a new trend deals with combining both of them in a two steps approach. In general, the flow of alarms provided by a signature based IDS is analyzed with a profile based method in order to detect anomalies in the alarm flow. Performance of such an approach is very low [17]. We therefore argue that it is necessary to combine both detection techniques in a two steps approach. But we do think that the right approach deals with first using the profile based technique in order to detect traffic profile anomalies. In that case, the detection thresholds are set with very pessimistic values in order to avoid false negatives. Then, we apply a signature based detection technique which has also the capability of classifying the anomalies. It then helps to eliminate false positives, but also, by classifying the detected anomalies, to identify the kind of anomaly as well as the intension behind the anomaly (legitimate or malicious).

This paper then relies on our NADA [7] anomaly detection tool which has been designed following this two steps approach principle. The criterias used for the detection step are very simple and rough: it computes the number of packets, the number of bytes and the number of SYN packets. It monitors the evolution of these criterias and if a significant change is discovered, it raises an alarm. When it is the case, the network traffic is then deeper analyzed and several attributes are built from, either the detection step, either some indices built on network packets fields. All these attributes are then used by the classification system. This system uses a set of signatures that use attributes directly linked to the packet headers and are thus easily understood by network operators. This is one of the key features of this system.

These signatures have been built through expert knowledge in the domain of network traffic anomalies. Therefore, human intervention is required for the creation and tuning of the signatures. The purpose of our method is to create these signatures automatically in order to build a system that would be able to work autonomously. In order to achieve this goal, we are using machine learning.

3.2 Representations of traffic

In all the previous work that we are aware of [12, 6], two possible representations of network traffic through unsupervised learning have been considered. In the first representation, the network traffic is represented by several classes and each class is associated with a part of the network traffic. This situation is shown in figure 1 (a) (each cluster represents a part of the network traffic (legitimate or anomalous) and in figure 1 (c) (each curve represents a class of traffic). In the other representation, each class is a part of the normal traffic and any isolated point (or outlier) is considered anomalous. In figure 1 (d), the gaussian curve is the normal traffic and any point located too far from this curve is anomalous. In figure 1 (b), one cluster represents the normal traffic and each outlier represents an anomaly (here, the isolated dots).

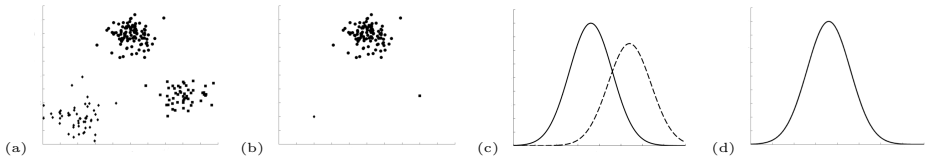

Fig. 1. Model with clusters (a), Model with one cluster and outliers (b), Model with one class (c), Model with two class (d)

3.3 Choice of machine learning techniques

Supervised/semi-supervised learning presents limitations because their use implies that we have labelled data at our disposal, i.e., in this case, traces for which we know that, at a certain time and for a certain duration, an anomaly has occurred. This, of course, implies that the considered anomalies are known and characterized, what is completely opposed to the goal addressed in this paper: real-time discovery of *0day anomalies*.

Unsupervised learning does not present this limitation. In fact, its purpose is precisely to find structure inside unknown data. Therefore, unsupervised learning appears as the technique to use.

Among the unsupervised techniques, we need one able to identify all the classes of traffic and that can keep some understandable attributes. We will only consider dimensional reduction, density estimation and clustering as they appear as the three most represented techniques in the literature.

- Dimensional reduction
  The principle of dimensional reduction deals with projecting the data from a vector space of high dimensions to a vector space of low dimensions. In our case, it means that we would end up with a vector space with a basis built on vectors that would have no physical/concrete meaning. One of our goal being

to keep some understandable attributes in order to have easy to understand and meaningful signatures (i.e. for expressing anomaly characteristics), the dimensional reduction is in clear contradiction with our requirements.

- Density estimation
  Density estimation is a family of methods for "one-class" problems. Its objective is to estimate the distribution of a set of observations and then predict whether or not a new observation should be considered as an outlier or a "normal" member of the single class. Density estimation is an efficient technique to detect outliers if the normal traffic is a single class and anomalies are only outliers. On the other hand, density estimation is inoperant if it has to consider several classes or if some anomalies are represented as classes. We cannot guarantee any of these conditions, therefore, density estimation is not suited to our case as global traffic can consist of several traffic classes.
- Clustering
  Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar under some chosen criterias.
  Clustering does not have any of the limitations listed above: it keeps all the attributes in a clear and intelligible form and it can consider and analyze them without any limitation on the number of classes (in our case, the number of classes of traffic including both normal and anomalous ones). Based on our first experiences, we selected the clustering technique as it appears as the most adapted and promising form of unsupervised learning for our problem.

3.4 Discovering unknown new anomalies with machine learning

In the previous subsection, we established two facts. First, it is possible to represent the traffic inside a vector space built on attributes. Second, unsupervised learning is able to extract the structure of a dataset from its representation in a vector space.

The interest of the extracted structure is directly linked to the pertinence of the considered vector space, i.e. the considered attributes. In our case, this pertinence is also related to the choice of two parameters: first, the aggregation metric used to structure the vector space during the traffic processing which determines how the group of packets are built, and second, the attributes built from the aggregated traffic.

Signatures are most of the times using different attributes. This implies that for trying to find these new signatures from scratch, it is needed to search for new previously unused attributes. Therefore, the discovery of new types of anomaly seems to be heavily linked to the discovery of new pertinent attributes.

The method that we intend to use to find new anomaly signatures is to look for anomalous representations (i.e. clusters or outliers) in the representation of the network traffic inside new attributes. In order to do so, we intend to generate new attributes, systematically try to find anomaly representations to assess the

presence of a new anomaly, and if it is the case, build the corresponding new signature.

The problem of creating new pertinent signatures can then be split into three tasks: first, process the network traffic, second, generate new attributes, and third, search for new anomalies inside combinations of generated attributes.

Traffic aggregation and processing Aggregation of traffic is the first part of traffic data processing. It is an important function because it allows us to change the point of view on the network traffic by changing the aggregation criteria. In fact, by aggregating the traffic according to the destination address of the network traffic with a certain network mask, one aggregate traffic destined to a restrained number of destinations hosts. One can then find anomalies that are impacting a small number of destination addresses (in some case targets of attacks) no matter how many sources are involved. This enables us to target anomalous traffic such as DDoS (Distributed Denial of Service) attacks. Corollary, for searching anomalies having a few number of sources and paying attention to the number of destinations, i.e. network scan or SYN port scan for instance, aggregation through the source IP address is the aggregation criteria to use. Similarly, it is also needed to target anomalies linked to the port number. The port number can then be used as an important aggregation parameter.

Attribute generation

- Create new attribute
  Currently, considered attributes are built on the distribution of the values of fields of packet headers. Some attributes are even built from values obtained over two different packet fields. We generalize this construction by using two steps. First, process values over the distributions of values of packet fields of the layers network and transport of the OSI model (IP address, TCP/UDP ports source/destination, flags, ...). The operators used on the distributions will be simple: number of different elements, proportion of the biggest element over the total, ... Second, sweep all possible combinations of one or two elements of the previously generated values. Once the combinations are obtained, we generate the attribute. If the combination contain one value, then the value is turned into an attribute. If the combination contain two values, then, we process the ratio of the first value over the second. At the end, we obtain a set of attributes built over the packet headers.
  However, it is obvious that such a variety of possible combinations applied to a big number of packet fields will generate a huge amount of possible combinations. The next issue will be to eliminate the attributes that seem to be of less interest.
- Attribute interest assessment
  If we want to extract clusters/outliers from the data spaces, we will need attributes that have a quantity of information as significant as possible. If all the values of the parts of the traffic for the considered attribute are

close from a certain value, the search for network classes or outliers inside this restricted space will be very complex and unreliable. Therefore, a first elimination of the attribute with poor interest seems relevant. Entropy is the mathematical tool that will be used for the evaluation of the quantity of information contained in the considered attributes.

Anomaly search After the attribute generation step, our algorithm have several attributes built on the packet fields. The next step is now to apply unsupervised learning in order to find new anomalies.

We intend to sweep all possible combinations of the previously generated attributes and search for anomalies inside each combination. As previously said, each combination of attributes can be used to build a vector space with a basis constituted of the attributes selected. Therefore, we search for the presence of an anomaly inside the chosen vector space.

There are two steps in the search for anomaly: first, characterization of the traffic in the chosen vector space, and then application of unsupervised learning according to the result of the first substep. These two steps will be repeated on each vector space and on each combinations of attributes.

- Characterization of the traffic in the chosen vector space
  In order to find a new anomaly, we try to find a pertinent representation of a 0day anomaly in a new vector space built from new attributes. This representation may be either a cluster or an outlier.
  For this anomaly detection step, we apply a clustering technique on the traffic. The result of this step gives us the structure of the network traffic.
  At this point, two cases arise. First, we obtain one cluster. By doing the assumption that the normal traffic is much more important in volume than the anomalous one, we deduce that the normal traffic is composed of the only found cluster. Therefore, in this case, anomaly detection will be using outlier detection.
  Second, we obtain several clusters: this means that the traffic is composed of several classes. However, nothing guarantees that one of these clusters is not actually an anomaly. This step will then require human intervention to manually identify the clusters between genuine and anomalous network traffic. As far as our statistical study have advanced (cf. 4.3), this case seems to be rare. Therefore, human intervention seems not to be needed. However, our study being statistical, we cannot guarantee that this case is totally irrelevant.
- Search for anomalies
  According to the results of the characterization of the network traffic, we use the appropriated unsupervised learning technique to search for a 0day anomaly and its associated signature. Several situations are possible. If all clusters belong to legitimate traffic, outlier detection will be applied in order to search for anomalies. If there are some anomalous clusters, we will still apply outlier detection because there might be other anomalies (represented as outlier) than the ones in the clusters. At this point, we are able to identify

the legitimate and the anomalous network traffic and know wether there is a new anomaly inside the considered vector space or not.
If it is the case, the matter of building the new signature is simple. In fact, once the anomalousness of each cluster and the presence of outliers is assessed, a convex or concave hull is drawn at half-distance between the normal representation(s) of traffic and the anomalous one(s). To obtain the updated signatures in terms of thresholds on a specific attribute, the hull is projected onto each axis of the vector space. In order to improve the system, we also could keep the hull and use it as a unique multi-dimensional threshold.

4 Validation

4.1 Data

A proper statistical validation of anomaly detection procedures requires the use of data with known, documented anomalies which can serve as the ground truth. Data might be collected from a real network and labelled afterwards by expert network operators. This would generate a dataset with known real anomalies (i.e. anomalies that happened on the wild), but might be prone to human errors (i.e. network operators might manually misclassify an anomaly), and does not permit control over the anomalies' characteristics (e.g. their intensity). Generating such datasets is expensive and currently very few are publicly available. The other way to generate labelled data is to artificially produce anomalies in real or simulated networks. With this approach, anomalies can be fully documented and are not subject to misinterpretations. Characteristics of the anomalies can also be controlled (i.e. varying its intensity, duration, etc.) to permit evaluation under different settings. The drawback is that the anomalies might not be too representative of current occurrences. We use both types of datasets to validate our algorithm: the METROSEC project [15] traces with artificially created anomalies and the MAWI traffic repository [14] with anomalies seen in the wild.

The first part of the traces used during our experiments comes from the MAWI dataset. It is composed of 15 minutes packets traces collected daily at 2PM from a Japanese network called WIDE since 1999 to present. These traces are provided publicly after being anonymized and stripped of their payload data. These traces are undocumented, but the authors of [5] started to label anomalies found in this database (http://www.manaworld.org/wide/anomalies/). Traces used are from samplepoint-B which is a trans-Pacific link between Japan and the United States. Traffic on this link is mostly exchanged between Japanese universities and commercial ISPs and consistently contain anomalies [2].

The second part comes from the METROSEC project. These traces consist of real traffic collected on the French National Research and Education Network (RENATER) with simulated attacks performed using real DDoS attack tools. This dataset was created in the context of the METROSEC research project. Traces contain anomalies that range from very low intensity (i.e. less than 4% of normal traffic volume) to very high (i.e. more than 80%). The traces are fully

documented with start and ending time of capture and attack, intensity, type and number of bots (i.e. attacking sources) of the attacks.

4.2 Methodology

We want to demonstrate that our system is able to find an unknown anomaly inside network traffic. The unknown aspect is to be considered from the point of view of the detection system: it means that the system has no a priori knowledge over this kind of anomaly.

In order to validate our approach, we plan to use an incremental implementation and validation of our algorithm. It will allow us to validate each part of our algorithm separately. The validation will then consist of two steps:

- First, we want to detect an anomaly, unknown by the system, inside a trace where we know that the anomaly is present, and build its signature. The only parameters given to the algorithm will be the attributes (and thus the vector space) that will be used to find the anomaly. This implies that we skip the steps of our algorithm related to attribute generation and attribute selection. Then, we apply our algorithm and search for an anomaly using only the attributes related to the targeted anomaly. Therefore, by looking at the right attributes inside a documented tracefile which contains the anomaly that will fit these chosen attributes, we are supposed to find it. We also extend this work to several types of anomalies with their appropriate attributes.
- Second, we want to validate the global behavior of our algorithm. In order to do so, we use a documented trace file with a known anomaly inside. We do not proceed to any restriction over the used attributes and use instead the attribute generation and attribute selection steps of our algorithm. The validation of this step will be the finding of (at least) the documented anomaly and maybe other undocumented anomalies.

4.3 Experimentations

We chose to focus ourselves on the detection of TCP SYN DDoS as if it was an unknown attack, i.e. without any prior knowledge other than the attributes used for its signature, in this case: #respdest (number of responsible destinations), spprop (ratio of the number of SYN to the number of packets), oneportpred (occurence of main port over every other ports) and #rpkt/#rdstport (ratio of the number of packets to responsible destination ports). In order to apply our method, we use the parameters that fit this type of anomaly. The aggregation parameter used is the destination since we want to target an anomaly with several sources and only one destination. The attributes used to build the vector space are the ones related to this type anomaly cited above (#respdest, spprop, oneportpred and #rpkt/#rdstport). The next part explains the results of our investigations about the structure of the traffic in this restricted vector space, and then, the result of the anomaly search.

Characterization of network traffic We proceed to the analysis of the traffic on the TCP SYN Flood DDoS attributes. In order to do this, we study several traces from the datasets cited in 4.1. We use 64 traces from the MAWI dataset. We also use one trace from Metrosec which has not any provoked anomaly inside.

We observe manually the data in the vector space considered. In order to do so, we generated two 3D images to be able to cover all the attributes of the chosen vector space and this, for all the traces. In the first image, the attributes used were spprop, oneportpred and #rpkt/#rdstport, in the second, we used #respdest, spprop and oneportpred. Then, we analyze them by hand.

It appears that for all images of the first type, the network traffic is composed of only one cluster. A good example is provided by figures 2 (a) and (b). The images of the second type are generally composed of only one class. The data often presents more noise than for the first attributes (cf figure 2 (c)). Some clusters arise, as in figure 2 (d), but as far as we now, they are directly related to scan events (network scans or port scans or both at the same time).

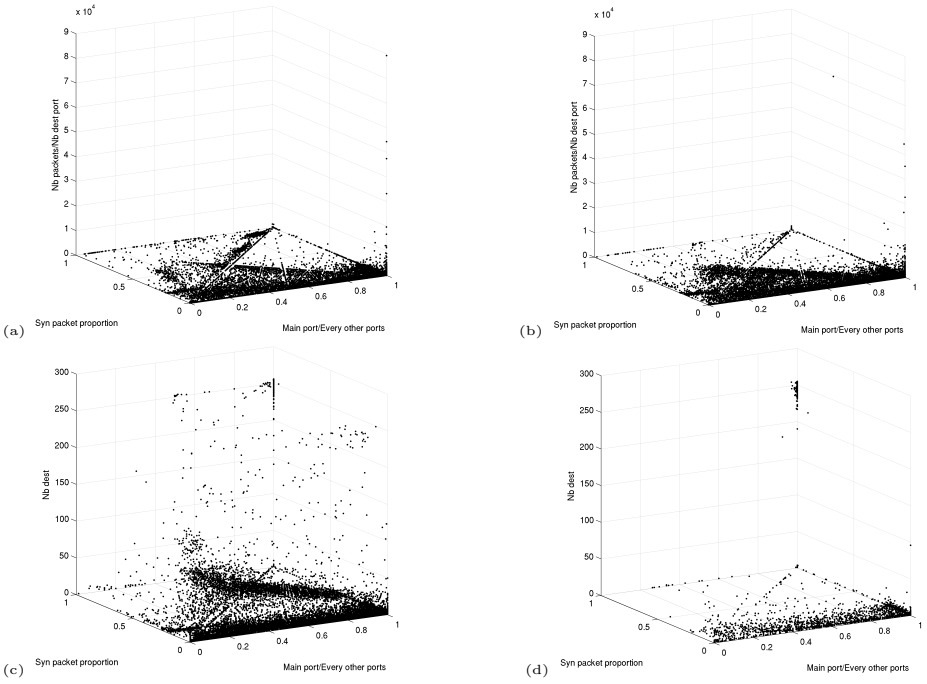

Fig. 2. Traffic representation for *spprop*, *oneportpred* and *#rpkt/#rdstport* (curves a et b) and *spprop*, *oneportpred* and *#respdest* (curves c et d)

However, considering the whole set of generated images, we can consider that the traffic is statistically composed of one class in the considered vector space.

0day anomaly search As the traffic is generally composed of only one main class, the technique used to find a 0day anomaly in the vector space is outlier detection. We use a documented trace from the Metrosec project where a TCP SYN Flood DDoS has been produced and captured. We extracted a segment of the original trace situated in the middle of the attack. We worked on part of the trace in order to reproduce the behavior of an online system that would operate on a finite windows of time. We then apply an outlier detection algorithm.

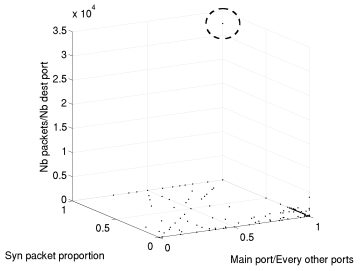

Fig. 3. Network traffic with a TCP SYN DDoS occuring

Our anomaly detection system is able to detect the outlier corresponding to the attack. This outlier corresponds to the first attack. Figure 3 shows the data space representation of the traffic inside three of the four attributes used for the outlier detection. It clearly appears that the point that represents the anomaly is on the top corner of the figure, while the normal traffic appears on the horizontal bottom plan. The generated signature will then be the one on equation 1.

$$\#rpkt/\#rdstport > 15000 \tag{1}$$

5 Conclusion

We propose a complete method to detect 0day network traffic anomalies and corresponding signatures through the use of machine learning. This method uses an automatic generation of attributes in order to generate semantically interesting attributes. Machine learning is then applied to several combinations of these attributes. At the end, our algorithm is able to find an anomaly it did not know before which could have been a 0day attack. It was then able to build the related signature automatically which can be integrated in security devices as IDS, IPS, firewalls, ... It was illustrated in this paper by a TCP SYN DDoS attack which was unknown from the system before it encounters it for the first time.

References

1. Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A signal analysis of network traffic anomalies. In *IMW '02: Proceedings of the 2nd ACM SIG-*

COMM Workshop on Internet measurement, pages 71–82, New York, NY, USA, 2002. ACM.
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, April 2009.
3. Jake D. Brutlag. Aberrant behavior detection in time series for network monitoring. In *LISA '00: Proceedings of the 14th USENIX conference on System administration*, pages 139–146, Berkeley, CA, USA, 2000. USENIX Association.
4. P. Chhabra, C. Scott, E.D. Kolaczyk, and M. Crovella. Distributed spatial anomaly detection. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1705–1713, April 2008.
5. Guillaume Dewaele, Kensuke Fukuda, Pierre Borgnat, Patrice Abry, and Kenjiro Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. In *LSAD '07: Proceedings of the 2007 workshop on Large scale attack defense*, pages 145–152, New York, NY, USA, 2007. ACM.
6. Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer, 2002.
7. Guilherme Fernandes and Philippe Owezarski. Automated classification of network traffic anomalies. In *5th International ICST conference on Security and Privacy in Communication networks (SecureComm'2009)*, Athens Greece, sept 2009.
8. KDD99. Kdd99 cup dataset, 1999. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.
9. Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. Sketch-based change detection: methods, evaluation, and applications. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247, New York, NY, USA, 2003. ACM.
10. Liwei Kuang and Mohammad Zulkernine. An anomaly intrusion detection method using the csi-knn algorithm. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 921–926, New York, NY, USA, 2008. ACM.
11. Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, New York, NY, USA, 2004. ACM.
12. Anukool Lakhina, Mark Crovella, and Christophe Diot. Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, 35(4):217–228, 2005.
13. Xin Li, Fang Bian, Mark Crovella, Christophe Diot, Ramesh Govindan, Gianluca Iannaccone, and Anukool Lakhina. Detection and identification of network anomalies using sketch subspaces. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 147–152, New York, NY, USA, 2006. ACM.
14. MAWI. Mawi dataset. At `http://mawi.wide.ad.jp/`.
15. METROSEC. Metrosec dataset. At `http://www.laas.fr/METROSEC`.
16. Antoine Scherrer, Nicolas Larrieu, Philippe Owezarski, Pierre Borgnat, and Patrice Abry. Non-gaussian and long memory statistical characterizations for internet traffic with anomalies. *IEEE Trans. Dependable Secur. Comput.*, 4(1):56–70, 2007.
17. Jouni Viinikka, Herv Debar, Ludovic M, and Renaud Sguier. Time series modeling for ids alert management. In *Proceedings of the ACM Symposium on InformAtion, Computer and Communications Security (AsiaCCS)*, march 2006.