Intelligent Thresholding

Alban Siffer

Amossys, Univ. Rennes, Inria, CNRS, IRISA

Abstract. Intrusion Detection Systems (IDS) have made a great progress for decades but even with the increasing power of AI, we struggle to design a general machine able to detect all kinds of cyber-attacks, especially those still unknown a.k.a. zero-day. Indeed, one reason why previous approaches failed can be the complexity of the cyber-security field. However, some research works on anomaly detection have made significative progress on aspects related to real-world issues. In particular, the calibration of the algorithms does not draw as much attention as their performance while all the intelligence can vanish through fine-tuning steps. Here, we tackle issues around the final decision threshold and show how it can be cleverly set.

1 Introduction

AI is currently one of the most trendy topic over the world. Outright far from the dream of AI with which movies and media delude us, both research and industrial actors are aware that our progresses in AI increase our overall knowledges and abilities. They provide much understanding about processes and behaviours through data mining and tasks formerly hard then become trivial thanks to machine learning.

However, all these cutting-edge methods seem to elude the cyber-security field. Paradoxically, data are plentiful and our expertise is very advanced. Actually, the aims are more ambitious and the challenges are more constrained.

Two intrusion detection approaches exist: rule-based and anomaly-based. As they are efficient and mature, rule-based solutions [2, 1, 3] are the most widely deployed but anomaly detection tries to extend beyond the scope of attack signature matching. Indeed, the latter has several weaknesses: building signatures is lengthy and laborious, the payload is likely to be inspected although it is an expensive task which becomes deprecated with the wide use of end-to-end encryption and eventually, rule-based methods are static thus new attacks cannot be detected. Anomaly-based techniques aim to build a model of *normality* (in a supervised or unsupervised way) in order to discriminate abnormal observations (often considered as attacks).

Much work has been done to develop anomaly detectors and with the help of cutting edge algorithms, smarter IDS have been proposed [31]. Despite the feeling that research is far from the industrial context, several works attempt to solve real-world issues.

One of these problems is what is generally called *parametrization*. Anomaly detectors can be very powerful provided that they are well configured. So they generally need a great amount of expertise to be set up in a specific network, in particular to set decision thresholds. Actually, they often have no meaning and require a time-consuming fine-tuning step to avoid too many false positive.

First, we will present why such a distrust of Machine Learning/Data Mining (ML/DM) techniques for intrusion detection. Then, smart thresholding ideas and techniques will be enhanced showing how intelligence can be integrated within real-world systems.

2 AI for intrusion detection

When we talk with a security expert about machine learning and behavioural detection, he would say it does not work and probably that it will never work. This feeling is quite reasonable in light of the original (and more current) research works.

2.1 Cyber-complexity

The first reason explaining why AI failed in cybersecurity while it revolutionizes other fields is the context specificity. Solving the GO game, recognizing human faces or making a car autonomous are real feats but detecting zero-attacks is far harder. Designing algorithms to this purpose is really demanding and we can list some of the difficulties:

- All the networks are different, so adaptable techniques are paramount
- To detect new attacks, algorithms must not be stuck to training observations, so supervised methods are not suitable
- As the context is constantly evolving, dynamic models are preferred
- Online detection is required to prevent systems from being damaged as soon as possible

2.2 Historical background

The pioneering work of Denning [14] is one of the most relevant in intrusion detection. She has given a model for a real-time IDS based on the detection of abnormal behaviours. Then several solutions, anomaly-based and also rule-based (expert systems), have been proposed and some of them are described in [32]. After these initial works (until the late 1990s), automation has taken a great part in the research in order to exploit larger data volumes and make systems more efficient while reducing human intervention. Underlying models were now build from training data and not from expertise [40].

A large amount of work was done in the early 2000s. Actually, many surveys about ML/DM techniques for intrusion detection [10, 12, 21, 40] analyse publications from those years.

2.3 KDD99 performance race

Several evidence can explain the increasing research interest in the early 2000s. Undoubtedly, the improvement of ML/DM algorithms (in concert with the computation power) is a key element. Moreover, the releases of specific datasets from the DARPA [26, 27] made the training, the evaluation and the comparison of anomaly-based intrusion detection techniques far easier.

In particular, KDD99 [38], a derivative of DARPA98 is currently the most widely used dataset to benchmark IDS through the whole literature. However, in spite of its practical aspect, KDD99 is a problematic dataset. From 2000, McHugh [30] was yet criticizing it, mainly pointing the procedure to generate the data as they were synthetic and not representing real world traffic. Then, the analysis of Mahoney and Chan [28] and Tavallaee *et al.* [39] followed, highlighting some simulation artefacts, the records redundancy and the lack of exact definition of attacks.

More than that, this dataset may encourage supervised methods as the observations are labeled. In fact, such approaches only generalize signature-based techniques making the known rules potentially more robust [18]. They logically cannot be used to build a pure anomaly detector.

However, the 2000s saw the application of all the common and well proven ML/DM methods on this dataset (ANN [11], HMM [24], Nave Bayes [8], Random Forest [41], SVM [23] etc.). See for instance [33, 12] for a rich description of these techniques to cyber-security.

Their results cannot be denied nevertheless the deployment of such methods in real-world contexts remains uncertain. The need of labeled data for training, the use of computationally expensive algorithms and the required expertise for parametrization may break the demands from the cybersecurity field.

2.4 Improvements and new challenges

Many recent publications still use KDD99 to test their (often supervised) algorithms [4, 6, 7]. However, several research works have gone beyond by developing unsupervised techniques [35, 25] and testing it in real world contexts [13, 15]. Unsupervised algorithms are very appealing because they do not require labeled data for learning. Nonetheless, the counterpart is that they have several parameters to tune so as to reduce their [often high] false alarm rate.

3 Automatic thresholding

The power of unsupervised anomaly detectors is to find what human cannot as they take into account thousands, millions or even more data at the same time without any previous knowledge.

3.1 Design of unsupervised anomaly detectors

Technically, these algorithms rank the observations according to a certain normality/abnormality degree but they are unable to decide which ones are real anomalies without human intervention. In practice, a threshold z is set in back-end to as to take the final decision (see [31] for a recent example). If the abnormality score is higher than z, the observation is an anomaly, otherwise it is normal (see Fig. 1).

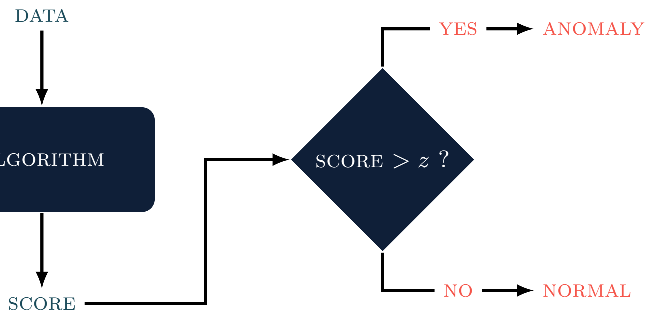

Fig. 1. Generic design of anomaly detectors

The parameter z can often be seen as a detection/false-positive regulator: a “high” value for z will catch only the most abnormal data, then many other real anomalies will be missed (low detection rate) but the probability to be wrong will be reduced (low false-positive rate). The converse phenomenon happens when z is “low”.

Using such a threshold is not a problem insofar as it has a meaning, so a procedure exists to set it. Unfortunately many shortcomings remain as research focus more on the “scoring” part. Usually, decision thresholds are set (after a fine-tuning step) to get the best results on some datasets and they lack interpretability.

3.2 The probabilistic point of view

The main solution to tackle both limits is to apply a statistical approach. Indeed, if we know the probability distribution of the output score S we can compute a threshold z such that

$$\mathbb{P}(S > z) < q,$$

where q (given as an input parameter) is the maximum false alarm rate desired. Thus, the threshold z has a real meaning and can be set (through the value of q) with a non substantial expertise.

Albeit this approach looks powerful, current statistical approaches to perform anomaly detection suffer from the inherent problem, namely the distribution assumption for S [5, 16]. Unfortunately, such hypothesis are very strong: algorithm lacks adaptability and validation is required. In a nutshell, it corresponds to a large amount of exogenous information which restricts application cases.

Beyond these hindrances, we propose SPOT [37], a new algorithm based on Extreme Value Theory (EVT) which is able to infer the correct probabilistic model in the decision area. Thus, without any knowledge about the probability distribution of the input data, SPOT can compute the correct threshold z with respect to the given probability q (z is then noted z_q). The value z_q is then used to perform anomaly detection in streaming data thanks to a continuous learning design.

4 SPOT algorithm

In this section, we present the SPOT algorithm. We introduce its theoretical basis (EVT), its architecture and a real world use case to detect network SYN scan attacks.

4.1 Extreme Value Theory

Fundamentals. Many techniques allow the scientist to find statistical thresholds which are actually quantiles (values z_q such that $\mathbb{P}(S > z_q) = q$. For instance, we can compute them empirically or assume a distribution. However data do not necessarily follow well-known distributions (Gaussian, uniform, exponential etc.) so the model step (the choice of the correct distribution) could be hard, even inappropriate. Moreover, if we want to predict *extreme* events, like rare or unprecedented events, the empirical method will not give accurate estimation (an unprecedented event would have a probability equal to zero). The extreme value theory addresses these problems by inferring the distribution of the extreme events we might monitor, without strong hypothesis on the original distribution. Here, we suppose that all the observations $X_1, X_2 \dots$ are independent random variables drawn from an unknown but stationary distribution (it does not change over time).

A beautiful result from Fisher, Tippett [19] and later Gnedenko [22] states that, under a weak condition (satisfied in practice), these extreme events have the same kind of distribution, regardless of the original one (called Extreme Value Distributions). This theorem is the theoretical principle of EVT. As extreme events are in the *tail* of the distribution, this theorem shows that all the possible tails can be caught through a single distribution family (so we only have a few parameters to estimate). So the aim is to fit the right tail (an example is described in figure 2).

Peaks-Over-Threshold approach. A further result from Pickands, Balkema and de Haan [9, 34] gives a more practical method to infer the tail of a distribution.

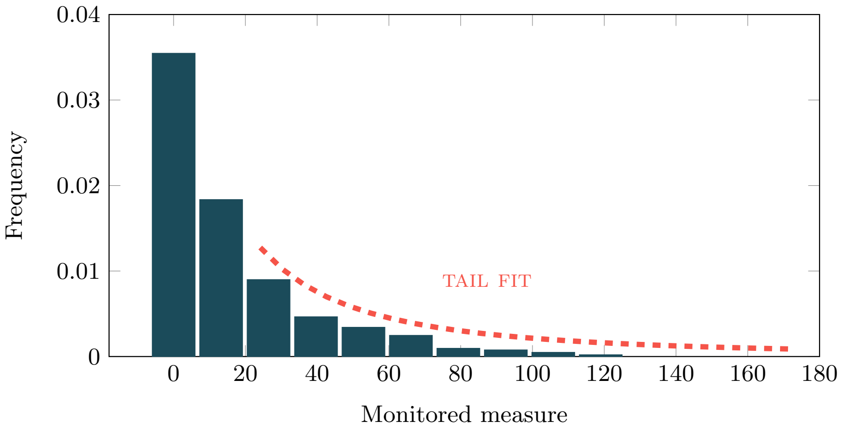

Fig. 2. Example of tail inference from empirical observations

In a word, their theorem stems to the following asymptotic equivalence[1]:

$$\mathbb{P}\left(X - t > x \mid X > t\right) \underset{t \to \tau}{\sim} \left(1 + \frac{\gamma x}{\sigma(t)}\right)^{-\frac{1}{\gamma}}. \tag{1}$$

This result shows that the tail of a distribution can be inferred with the greatest observations we may have. Once the model is recovered (i.e. the parameters γ, σ are estimated) we can easily compute the quantile z_q such that $\mathbb{P}(X > z_q) < q$ for q as small as desired.

In more details, equation (1) gives a "natural" approach to estimate the tail of a distribution from an initial batch of observations $X_1, X_2 \dots X_n$, called Peaks-Over-Threshold (POT). As it is an asymptotic result we need to be close to the theoretical context (i.e. t close the upper bound of the distribution τ). However, through observations we do not know exactly the value of τ (it can also be infinite), so in practice we choose t equal to a high empirical quantile (99% for instance). In this condition, the equivalence (1) says that the *excesses* $X - t$ when $X > t$ follows the right side distribution called Generalized Pareto Distribution (GPD) with parameters (γ, σ) (here the location μ is null). So we have to estimate these two parameters and then we get a right model for extreme events (and we can compute z_q). This basic procedure is summed up below (algorithm 1).

4.2 SPOT algorithm

The SPOT (Streaming Peaks-Over-Threshold) algorithm generalizes the POT approach by updating the tail model with new incoming data. It makes use of this model to compute a quantile z_q, at the user-defined level q. This quantile

[1] τ is the upper bound of the distribution support, γ, σ are two parameters to infer

Algorithm 1 Peaks-over-Threshold

```
procedure POT($X_1, \dots X_n$)
    Set $t$ as an high quantile
    Retrieve the excesses $Y_i = X_i - t$ when $X_i > t$
    Fit a GPD to the excesses (estimate $\gamma$ and $\sigma$)
end procedure
```

is used as a decision threshold to discriminate normal values from outliers. The SPOT design is described in figure 3.

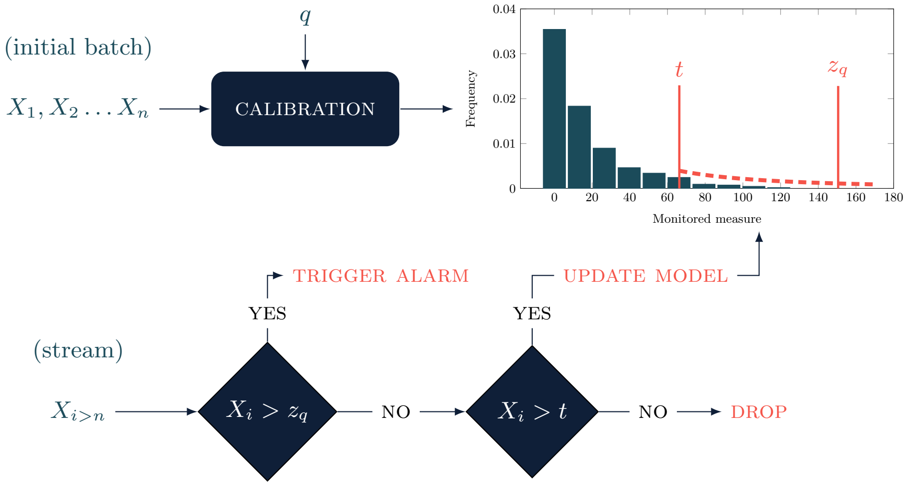

Fig. 3. SPOT algorithm design

First, it needs an initial batch of n values (few thousands in practice) to *calibrate*. This step runs the POT procedure to get an initial model (and the value of t). With this fit, we can compute a first threshold z_q according to the desired q (a very low value in practice). Then, we can process the stream. The three possible cases are described below:

1. [TRIGGER ALARM] The incoming data X_i is greater than z_q, so it is declared as an outlier;
2. [UPDATE MODEL] The incoming data X_i is between t and z_q (X_i is called a *peak*). In this case, it is in the tail of the distribution (but not an outlier), so we update the model (we perform POT with this additional observation);
3. [DROP] The incoming data X_i is lower than t, so this is just a normal observation and nothing is done.

At each new peak observation, we update and then refine the model, therefore z_q is continuously learnt. In this approach, all the peaks must be stored (so it theoretically requires an infinite memory) but in practice we can bound this amount of data by keeping only the last m peaks (with m large). This also allows to adapt in case of slight drift (as the oldest peaks are dropped). In this description we focus on upper thresholding but the algorithm can easily be applied to get lower bounds.

4.3 Experiments

To test our algorithm we use real data from the MAWI repository which contains daily network captures (15 minutes a day stored in a `.pcap` file). In these captures, MAWIlab [20] finds anomalies and labels them with the taxonomy proposed by Mazel *et al.* [29]. The anomalies are referred through detailed patterns. To be close to real monitoring systems we converted raw `.pcap` files into NetFlow format, which aggregates packets and retrieves meta-data only, and is commonly used to measure network activity. Then we labeled the flows according to the patterns given by the MAWIlab. In this experiment we use the two captures from the 17/08/2012 and the 18/08/2012.

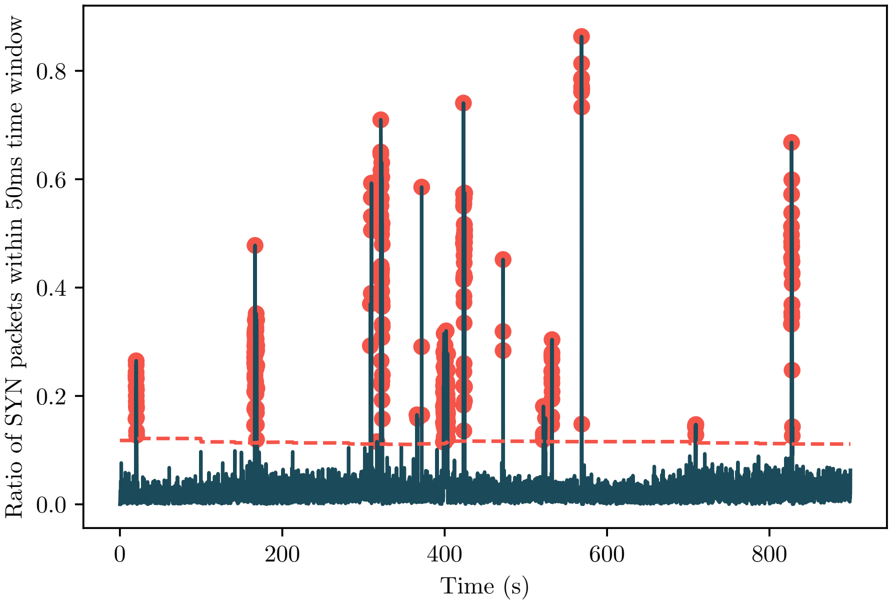

Fig. 4. SYN flood detection at level $q = 5.10^{-4}$

Classical attacks are network scans where many SYN packets are sent in order to find open and potentially vulnerable ports on several machines. A relevant feature to detect such attack is the ratio of SYN packets in a given time window [17]. From our NetFlow records we compute this feature on successive 50 ms time windows and we try to find extreme events. To initialize SPOT we use the last 2000 values of the 17/08 record and we let the algorithm working on the 18/08 capture.

The figure 4 shows the alerts triggered by SPOT (red circles). We recall that each point represents a 50 ms window gathering several flows (possibly benign and malicious). The computed threshold (dashed line) seems nearly constant but this behavior is due to the stability of the measure we monitor (SPOT has quickly inferred the behavior of the feature). By flagging all the flows in the triggered windows, we get a true positive rate equal to 86% with less than 4% of false positives.

False-positive regulation In the previous section we have given some arguments about the role of the main parameter q. Here, we study its impact on the MAWI dataset.

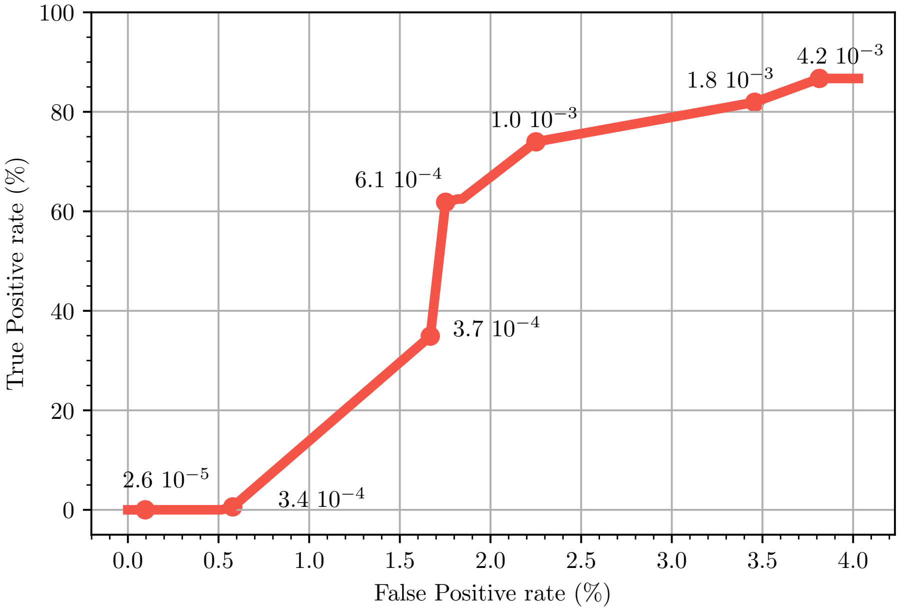

Fig. 5. ROC curves on MAWI dataset (the markers give the corresponding value of q)

On the figure 5, the ROC curve shows the effect of q on the False Positive rate (FPr). First we notice that the higher is q, the more we detect anomalies (and higher is the false positive rate too). Furthermore, values of q between 6.10^{-4}

and 10^{-3} allow to have a high True Positive rate (TPr) while keeping a rather low FPr: this leaves some room for error when setting q.

5 Conclusion

We have presented a novel approach to detect outliers in high throughput numerical time series, with a direct application to intrusion detection. This technique has many advantages which meet the requirements mentioned in 2.1. Actually, the key points of our approach is that it does not assume prior knowledge of the data distribution, and it does not require manually set thresholds, therefore it may adapt on multiple and complex contexts, learning how the interest measure behaves.

Many issues to build smarter IDS remain. In [36], Sadik and Gruenwald focus on research issues to detect outliers in streaming data. Logically, these demands on the algorithmic parts are really close to those from the cyber-security field. They tackle some aspects different from the thresholding problem as the concept drift and the multidimensionality. Taking into account all the requirements seems very hard and the new advances on some specific features might rely on a trade-off with others. Maybe intrusion detection research should focus on the design of specific but powerful components instead of a more general solution.

References

1. Bro, https://www.bro.org/
2. Snort, https://www.snort.org/
3. Suricata, https://suricata-ids.org/
4. Aburomman, A.A., Reaz, M.B.I.: A novel svm-knn-pso ensemble method for intrusion detection system. Applied Soft Computing **38**, 360372 (2016)
5. Agarwal, D.: An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In: Data Mining, Fifth IEEE International Conference on. pp. 8–pp. IEEE (2005)
6. Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A.: Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. Expert Systems with Applications **67**, 296303 (2017)
7. Aljawarneh, S., Aldwairi, M., Yassein, M.B.: Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science (2017)
8. Amor, N.B., Benferhat, S., Elouedi, Z.: Naive bayes vs decision trees in intrusion detection systems. In: Proceedings of the 2004 ACM symposium on Applied computing. pp. 420–424. ACM (2004)
9. Balkema, A.A., De Haan, L.: Residual life time at great age. The Annals of probability (1974)
10. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE communications surveys & tutorials **16**(1), 303336 (2014)

11. Bivens, A., Smith, R., Embrechts, M., Palagiri, C., Szymanski, B.: Network-based intrusion detection using neural networks. In: Proc. ANNIE 2002 Conference. pp. 10–13. ASME Press (2002)
12. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials **18**(2), 11531176 (2016)
13. Casas, P., Mazel, J., Owezarski, P.: Unada: Unsupervised network anomaly detection using sub-space outliers ranking. In: International Conference on Research in Networking. p. 4051. Springer (2011)
14. Denning, D.E.: An intrusion-detection model. IEEE Transactions on software engineering (2), 222232 (1987)
15. Dromard, J., Roudire, G., Owezarski, P.: Online and scalable unsupervised network anomaly detection method. IEEE Transactions on Network and Service Management **14**(1), 3447 (2017)
16. Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: In Proceedings of the International Conference on Machine Learning. Citeseer (2000)
17. Fernandes, G., Owezarski, P.: Automated classification of network traffic anomalies. In: ICSPCS (2009)
18. Ferragut, E.M., Laska, J., Bridges, R.A.: A new, principled approach to anomaly detection. In: Machine Learning and Applications (ICMLA), 2012 11th International Conference on. vol. 2, p. 210215. IEEE (2012)
19. Fisher, R.A., Tippett, L.H.C.: Limiting forms of the frequency distribution of the largest or smallest member of a sample. In: Mathematical Proceedings of the Cambridge Philosophical Society (1928)
20. Fontugne, R., Borgnat, P., Abry, P., Fukuda, K.: MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In: ACM CoNEXT '10 (2010)
21. Garcia-Teodoro, P., Diaz-Verdejo, J., Maci-Fernndez, G., Vzquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security **28**(12), 1828 (2009)
22. Gnedenko, B.: Sur la distribution limite du terme maximum d'une serie aleatoire. Annals of mathematics pp. 423–453 (1943)
23. Hu, W., Liao, Y., Vemuri, V.R.: Robust support vector machines for anomaly detection in computer security. In: ICMLA. p. 168174 (2003)
24. Joshi, S.S., Phoha, V.V.: Investigating hidden markov models capabilities in anomaly detection. In: Proceedings of the 43rd annual Southeast regional conference-Volume 1. pp. 98–103. ACM (2005)
25. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Computer Communication Review. vol. 35, pp. 217–228. ACM (2005)
26. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 darpa off-line intrusion detection evaluation. Computer networks **34**(4), 579595 (2000)
27. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K.: Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In: DARPA Information Survivability Conference and Exposition, 2000. DISCEX00. Proceedings. vol. 2, p. 1226. IEEE (2000)
28. Mahoney, M.V., Chan, P.K.: An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In: International Workshop on Recent Advances in Intrusion Detection. p. 220237. Springer (2003)

29. Mazel, J., Fontugne, R., Fukuda, K.: A taxonomy of anomalies in backbone network traffic. In: IWCMC (2014)
30. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Transactions on Information and System Security (TISSEC) **3**(4), 262294 (2000)
31. Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A.: Kitsune: An ensemble of autoencoders for online network intrusion detection. NDSS **5**, 2
32. Mukherjee, B., Heberlein, L.T., Levitt, K.N.: Network intrusion detection. IEEE network **8**(3), 2641 (1994)
33. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks **51**(12), 3448–3470 (2007)
34. Pickands III, J.: Statistical inference using extreme order statistics. the Annals of Statistics (1975)
35. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001. Citeseer (2001)
36. Sadik, S., Gruenwald, L.: Research issues in outlier detection for data streams. ACM SIGKDD Explorations Newsletter **15**(1), 33–40 (2014)
37. Siffer, A., Fouque, P.A., Termier, A., Largouet, C.: Anomaly detection in streams with extreme value theory. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1067–1075. ACM (2017)
38. Stolfo, S., et al.: Kdd cup 1999 dataset. UCI KDD repository. http://kdd. ics. uci. edu (1999)
39. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on. p. 16. IEEE (2009)
40. Wu, S.X., Banzhaf, W.: The use of computational intelligence in intrusion detection systems: A review. Applied soft computing **10**(1), 135 (2010)
41. Zhang, J., Zulkernine, M.: A hybrid network intrusion detection technique using random forests. In: Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. pp. 8–pp. IEEE (2006)