Facilitating DoS Attack Detection using Unsupervised Anomaly Detection

Christos Bellas, Georgia Kougka, Athanasios Naskos, Anastasios Gounaris, Apostolos Papadopoulos and Athena Vakali
{chribell,anaskos,georkoug,gounaria,papadopo,avakali}@csd.auth.gr
Aristotle University of Thessaloniki, Greece

Christos Xenakis
xenakis@unipi.gr
University of Piraeus, Greece

ABSTRACT

Modern techniques in intrusion and DoS (Denial of Service) detection tend to be either supervised or semi-supervised, i.e., they require training and labelled data. In this work, we study the problem of correlating security attacks with anomalies reported at runtime by a fully unsupervised outlier detection module, i.e., a component that does not require any training at all. Through a concrete proof-of-concept case study, we demonstrate that unsupervised anomaly detection is both efficient and effective, but still, it needs to be combined with additional mechanisms to yield a complete intrusion detection and prevention solution.

CCS CONCEPTS

• **Networks → Denial-of-service attacks**; • **Security and privacy → Intrusion detection systems**.

KEYWORDS

security, intrusion detection, dos attack

ACM Reference Format:
Christos Bellas, Georgia Kougka, Athanasios Naskos, Anastasios Gounaris, Apostolos Papadopoulos and Athena Vakali and Christos Xenakis. 2022. Facilitating DoS Attack Detection using Unsupervised Anomaly Detection. In *34th International Conference on Scientific and Statistical Database Management (SSDBM 2022), July 6–8, 2022, Copenhagen, Denmark.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3538712.3538736

1 INTRODUCTION

Anomaly (or outlier) detection is a fundamental pillar in data mining [1] and comprises several variants inspired by both supervised and unsupervised learning. Intrusion detection is among the core anomaly detection endeavours, since host-based intrusion exploits sequence mining and network-based intrusion detection can be managed by using anomaly detection for (streaming) data [2].

Our focus is put on network intrusion detection using unsupervised learning techniques, which, counter-intuitively, is an overlooked subject. A careful look at the literature reveals that there are many open issues in incorporating such techniques into IDS (Intrusion Detection Systems), IPS (Intrusion Prevention Systems) and SIEM (Security Information and Event Management) systems. More specifically, there are three main limitations that have motivated our work: (*i*) **Need for training**: Most proposals for network intrusion detection based on outlier analysis operate in either a supervised or semi-supervised manner [2, 6], which implies the need for training and, similarly to signature-based techniques, rely on expert's knowledge for correct labelling. (*ii*) **Lack for DoS evidence**: Interestingly, there is little evidence that unsupervised anomaly detection can facilitate in identifying security attacks, such as DoS, although there is evidence that it can detect some other types of security incidents, such as an abnormal number of IP addresses for a specific user [10]. (*iii*) **Complexity constraints**: Unsupervised anomaly detection techniques suffer from drawbacks of a high number of false positives and high time complexity, since they need to check incoming records against potentially big datasets instead of evaluating a pre-trained model; due to these reasons, they have not been widely implemented in modern systems [17]. However, as reported in [8], there is a need for such techniques because signature-based and supervised learning solutions cannot cope with new attacks, and also attacks identified in popular test datasets used for model training are outdated and do not typically correspond to attacks in practice despite the fact that these datasets are described as real-world ones. Thus, there are no guarantees on how well trained systems perform in real-world systems.

In this work, we aim to address the above challenges and respond to the following question: *"to what extent can modern unsupervised streaming anomaly detection techniques facilitate attack detection and thus enhance modern IDS/SIEM systems?"*. In our previous work [4], we have introduced a system that employs sophisticated supervised learning models to detect patterns of previously unknown threats and performs continuous anomaly detection in parallel. The latter functionality has been shown to be effective in identifying abnormal phenomena, but it followed an implementation that is not tailored to big data and without providing evidence that the reported outliers were signs of an attack in progress. In this work, we fill this gap by investigating the issue of providing evidence that unsupervised anomaly detection with no training at all can detect outliers corresponding to DoS attacks while meeting real-time requirements.

The rest of the paper is organized as follows. Related work is presented briefly in Sec. 2. In Sec. 3, we describe both our methodology and its application based on a proof-of-concept case study. Finally, Sec. 4 concludes our work and describes briefly future work in the area.

SSDBM 2022, July 6–8, 2022, Copenhagen, Denmark

ACM ISBN 978-1-4503-9667-7/22/07…$15.00
https://doi.org/10.1145/3538712.3538736

Table 1: Summary of the main limitations of related work.

Technique	Type	Short Description
[3, 10, 26]	Unsupervised	targets security incidents but no attacks
[5–7, 11, 14, 18, 25]	Unsupervised	inefficient in a streaming scenario
[12]	Semi-supervised	assumption of a period without attacks
[9]	Supervised	relies on training

2 RELATED WORK

We briefly review the similar proposals for anomaly Detection in IDS/SIEM environments. As already mentioned, and according to multiple authors, e.g. [9, 16], the vast majority of ML-based anomaly detection techniques are supervised or semi-supervised, which suffer from the aforementioned limitations [6, 8]. Because of these limitations, we focus exclusively on unsupervised techniques. One of the works closest to us has appeared in [10], where a multivariate unsupervised anomaly detection algorithm is proposed. This algorithm is evaluated on a real-world dataset and is one of the first proposed unsupervised anomaly detection techniques for SIEM systems. Anomalous instances cover aspects such as *events per privileged user per hour per day*. A similar rationale has appeared in a subsequent work presented in [3], in which Asanger et al. show how the different security events can be pre-processed in order to apply unsupervised anomaly detection techniques. The focus is on the same techniques and the same aspects as previously, e.g., *tickets per user* or *IP addresses per user*, and so on. A third example of such an approach has appeared in [26]. The main difference between the proposals in [3, 10, 26] and our work is that they do not correlate outliers with attacks, such as DoS ones, while no real-time issues are directly targeted.

In [18], an anomaly detection technique is presented, where log data from SAP Hana are mapped to a vector space; each vector is derived from count metrics of predefined event types found in the logs. This method is combined with normalisation and the use of clustering algorithms on top of which anomaly detection is performed. Interestingly, this work shows that the anomalies detected correspond to two attacks, namely the unsuccessful password brute force attack using Hydra over the Remote Desktop Protocol (RDP) on the domain controller and the successful brute force attack over Lightweight Directory Access Protocol (LDAP) using Hydra. As also explained in [17], to account for real-time constraints, the authors resorted to a hybrid technique that also includes a training phase, while anomaly detection techniques suffer from generating many false positives. Other techniques, which also employ a similar mapping of logs to a vector space, are also described in [19], but the anomalies detected are not related to specific attack types.

A work targeting a specific attack type is [14], which focuses on Kerberoasting. Both unsupervised and semi-supervised anomaly detection were used. The evaluation was performed on a real-world dataset consisting of logs referring to Windows events for Kerberos service ticket requests. The results showed that the one-class SVM is characterized by a higher capability to detect this type of security-related incidents, but this technique cannot be efficiently applied to an online setting due to its high complexity.

A survey of Usama et. al [24] provides a summary of a set of techniques, such as [7] and [11] that focus on DoS, Probes, U2R (User to Root), and R2L(Remote to Local) attacks using anomaly detection. For example, the work presented in [7] focuses on reducing false positive rates in IDSs by applying a fuzzy rough clustering technique after a three-step pre-processing phase; the exact clustering technique is fuzzy C-means (FCM). Additionally, the work in [11] considers the application of K-Means in order to classify log data and eventually, detect intrusions. Similarly, the authors in [11] consider raw network data in offline scenarios. Our technique does not require patterns matching methodology to detect anomalies in network data or any kind of pre-processing that is inapplicable in streaming (real-time) scenarios. Inefficiency in streaming scenarios is also the main limitation of the work in [5, 6, 25]. Another application of anomaly detection techniques in an online scenario is in the context of smart grids, where anomalies usually correspond to power distribution failures and are caused by malicious attacks that overload the system infrastructure. In their work [12], Karimipour et al., propose an unsupervised anomaly detection technique, which is based on statistical correlation between system measurements. However, in their approach it is assumed that there is an initial part of the data stream without attacks.

Table 1 summarizes the techniques discussed above. The main conclusion from studying the literature is that there is a gap in developing unsupervised anomaly detection IDS techniques that directly target security attacks and are based on big streaming data, although there are unsupervised proposals that target security incidents but not attacks and/or cannot meet real-time requirements.

3 CORRELATING DISTANCE-BASED OUTLIERS WITH ATTACKS

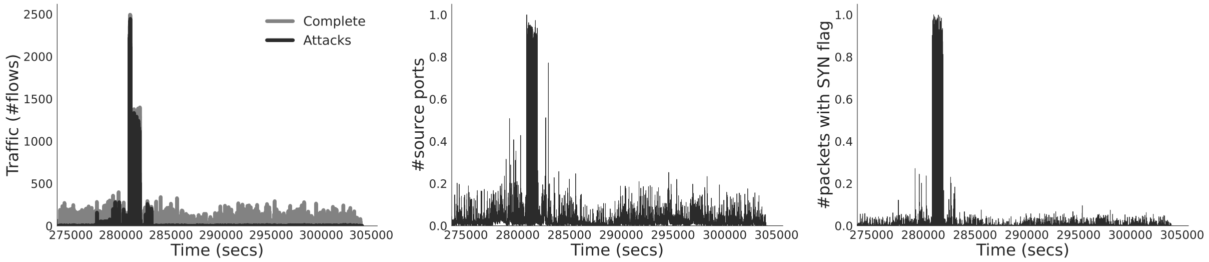
Our aim is to provide concrete insights into the capability of unsupervised anomaly analysis to facilitate real-time attack detection, as targeted by modern IDS and SIEM systems. To this end, we first describe our higher-level methodology and then, we explain the details of our proof-of-concept application.

3.1 Methodology

A key part in incorporating an unsupervised anomaly detection technique for network intrusion detection is the selection of the appropriate low-level metrics (features), on which we will conduct the outlier detection. In our case, we chose simple network traffic descriptors, such as the number of source/destination ports and the number of packets, which according to [6] are sufficient to describe DoS and DDoS attacks.

We employ distance-based outlier techniques, because they come with variants tailored to real-time processing [21, 23]. These techniques do not require any training at all. The specific technique reported in [4] is MCOD [13], but any similar technique is applicable as well. To determine if a data point is an outlier, its number of neighbors within a radius r must be less than k; apart from these two parameters, a distance function is required, and we typically choose Euclidean distance.

To process data over a streaming environment, there are two examined methods. In the first one, noted as *sliding window*, two additional parameters are involved: the size of the sliding time-based window w, which maintains only the most recent points and the slide size s, which defines the frequency of re-assessing the



(a) Traffic overview. **(b) Number of unique source ports.** **(c) Number of packets with SYN flag.**

Figure 1: A portion of traffic and two features measured per second.

outlierness status of all records that are currently active, i.e., they belong to the current window. In the second method, noted as *fixed window*, instead of a sliding time-based window, we have a fixed window of size w and every point of the slide of size s is compared against all the points in the fixed window. The latter method may be considered less adaptive to changes in a data stream, but it remains effective even when the majority of recent measurements in a window slide are actual outliers that need to be reported.

3.2 A case study scenario

Dataset. We use the Intrusion Detection Evaluation Dataset, noted as CIDS-2017[1], provided in [20]. As there are limited reliable datasets for validation, we have processed this dataset as it includes the most common attacks and it is continuously updated. Furthermore, the raw network traffic simulation is provided through PCAP files, and also, the network traffic analysis data that are generated by CICFlowMeter [15] are included. The produced data are annotated with timestamp, source and destination IPs, source and destination ports, protocols and attack incidents. We extract an eight hour portion of the original CIDS-2017 dataset, which consists mostly of DoS attack types. Out of the total 692703 network flows generated for this timespan, 36.48% are labeled as malicious, but most of them are concentrated in specific time periods.

Pre-processing. The data pre-processing scheme consists of two main steps. First, we use the provided attack metadata and we group the records by the labeled flows per second in order to derive the number of attacks per second. Out of 30453 seconds, only 10.92% are associated with an attack. In order to render attacks even rarer, we artificially increase the dataset by replicating benign behavior. The resulting dataset corresponds to 304086 seconds with 1% out of the total records associated with an attack. We use this information as ground truth in our evaluation. The data expansion has a small impact on the performance results presented later but prevents a dummy mechanism that always reports an attack to wrongly appear as effective. A portion of the network traffic alongside the respective number of attacks per second is illustrated in Figure 1a.

Second, using the `tshark` tool [2], we extract the following four features per second: (i) the number of packets, (ii) the number of unique source ports, (iii) the number of unique destination ports, and (iv) the number of packets with the TCP SYN flag. Figures 1b and 1c depict the number of unique source ports and the number of packets with SYN flag per second respectively, as a visual evidence that these features are correlated with attack incidents. However, as shown in Table 2, there is no strong correlation between data features and attacks, which renders the application of distance-based outlier detection particularly challenging.

Table 2: The feature - attack Pearson correlation.

Feature	Attack Correlation
Number of packets	0.19
Number of unique source ports	0.35
Number of unique destination ports	0.32
Number of packets with TCP SYN flag	0.27

Table 3: Performance results (the group of features are G1: Src/Dst ports & SYN pkts and G2: Src/Dst ports).

Features	Precision (%)	Recall (%)	Time per slide in msec
All	19.15	95.13	118.06
G1	27.19	85.94	126.51
G1	33.81	75.48	126.72
G2	45.73	61.27	9.46

Application of outlier detection. The input to the outlier detection (OD) algorithm is a time-series of extracted network features per second. In the conducted experiments, we use either all features or a group of features as input. If an evaluated point in the current slide is returned as an outlier and there are one or more attacks in the same second, then this result is considered as a true positive. Similarly, if OD reports an outlier in a second that does not contain any attack, we consider it as a false positive.

We employ a GPU-tailored implementation of OD[3]. Table 3 provides some indicative experimental results alongside the respective runtimes, which refer to runs using an NVIDIA Titan XP graphics card and the fixed window method. We omit the details on OD launch configuration, i.e. the w, s, r, k arguments due to space constraints and refer the reader to the source code repository for more details. However, we note that the slide corresponds to 1,000 seconds (i.e., more than 15 minutes) or more. As it can be seen, for higher recall values, OD has lower precision due to the high

[1] https://www.unb.ca/cic/datasets/ids-2017.html

[2] https://www.wireshark.org/

[3] The source code is available at https://github.com/chribell/cu-od/tree/curex

number of false positives, i.e., we experience the same problem as other efforts in encapsulating unsupervised anomaly detection in IDSs [17]. By experimenting with different groups of features as input, we manage to increase the precision at the expense of lower recall values. However, the main advantage of the parallel OD solution, is the very low execution time, a crucial property for real-time anomaly detection.

More specifically, the main observation is that we may process new data after each window slide in less than a second for the parameters tested. In our experimental evaluation, we manage to process data spanning more than 15 minutes in less than one second, which means that our solution can be efficient even when we desire to update results very frequently (e.g., in every second). Although the runtime numbers reported in the table are not directly comparable, they show that GPUs can offer much higher efficiency than other massively parallel streaming solutions, such as MCOD based on Apache Flink [22]. In addition, the works in [13, 22] employ the sliding window method, whereas in our case, we employ a fixed window, which is superior by at least 5-10%.

In addition, the results reveal that we manage to capture more than 60% of the attacks raising a false alarm in half of the cases. Training machine learning models and relying on signature-based solutions may seem to achieve higher performance, but, in light with the remarks reported in [8], we stress that these precision and recall values prove that unsupervised streaming anomaly detection is not only applicable at runtime, but can be used as a detection component targeting unknown behavior.

In summary, we observe that our unsupervised anomaly detection technique monitors network traffic online both effectively (i.e., detecting the majority of the attacks in this specific dataset) and efficiently, but with the disadvantage of a high false positive rate. The key remarks are presented below:

(1) We provide concrete evidence that distance-based outliers in traffic data are correlated with security attacks.
(2) We provide evidence that we can report outliers in a few milliseconds, much more efficiently than other parallel streaming solutions thus addressing a key limitation of previously reported unsupervised anomaly detection techniques.

4 DISCUSSION

In this work, motivated by the fact that there is a lack of techniques operating in a totally unsupervised manner for detecting attacks in real-time, we investigate (*i*) the correlation between outliers detected through a completely unsupervised techniques and DoS attacks; and (*ii*) the applicability of the former techniques in real-time scenarios. We provide evidence that unsupervised outlier mining can indeed detect attacks and these techniques can be extremely fast. However, due to the high false positive rates, encapsulating anomaly detection in current IDS and SIEM tools seems practical only in combination with additional mechanisms forming an ensemble solution. In such ensembles, continuous unsupervised anomaly detection can act both as an effective and efficient filter for further processing by downstream modules and as a core component that is responsible for detecting attacks for which no models and/or signatures exist. We leave the investigation of ensembles as future work. Another interesting direction is the study of additional types of attacks and systematic selection of features on top of which anomaly detection is performed.

Acknowledgements. This work has been supported by the European Commission under the H2020 Programme, through funding of the "CUREX: seCUre and pRivate hEalth data eXchange" project (No. 826404) and the "RAINBOW" project (No. 871403).

REFERENCES

[1] Charu C. Aggarwal. 2015. *Data Mining - The Textbook.* Springer.
[2] Charu C. Aggarwal. 2017. *Outlier Analysis, 2ed.* Springer.
[3] Stefan Asanger and Andrew Hutchison. 2013. Experiences and Challenges in Enhancing Security Information and Event Management Capability Using Unsupervised Anomaly Detection. In *ARES*.
[4] Christos Bellas, Athanasios Naskos, Georgia Kougka, George Vlahavas, Anastasios Gounaris, Athena Vakali, Apostolos Papadopoulos, Evmorfia Biliri, Nefeli Bountouni, and Gustavo Gonzalez Granadillo. 2020. A Methodology for Runtime Detection and Extraction of Threat Patterns. *SN Comput. Sci.* (2020).
[5] Rodrigo Braga, Edjard Mota, and Alexandre Passito. 2010. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *IEEE LCN*.
[6] Pedro Casas, Johan Mazel, and Philippe Owezarski. 2012. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. *Comput. Commun.* (2012).
[7] Witcha Chimphlee, Abdul Hanan Abdullah, Mohd Noor Md Sap, Surat Srinoy, and Siriporn Chimphlee. 2006. Anomaly-Based Intrusion Detection using Fuzzy Rough Clustering. In *ICHIT*.
[8] Dylan Chou and Meng Jiang. 2022. A Survey on Data-driven Network Intrusion Detection. *ACM Comput. Surv.* (2022).
[9] Abhishek Divekar, Meet Parekh, Vaibhav Savla, Rudra Mishra, and Mahesh Shirole. 2018. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. *CoRR* abs/1811.05372 (2018).
[10] Markus Goldstein, Stefan Asanger, Matthias Reif, and Andrew Hutchison. 2013. Enhancing Security Event Management Systems with Unsupervised Anomaly Detection.. In *ICPRAM*.
[11] Meng Jianliang, Shang Haikun, and Bian Ling. 2009. The Application on Intrusion Detection Based on K-means Cluster Algorithm. In *2009 International Forum on Information Technology and Applications*.
[12] Hadis Karimipour, Ali Dehghantanha, Reza M. Parizi, Kim-Kwang Raymond Choo, and Henry Leung. 2019. A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids. *IEEE Access* (2019).
[13] Maria Kontaki, Anastasios Gounaris, Apostolos N. Papadopoulos, Kostas Tsichlas, and Yannis Manolopoulos. 2016. Efficient and flexible algorithms for monitoring distance-based outliers over data streams. *Inf. Syst.* (2016).
[14] Lukás Kotlaba, Simona Buchovecká, and Róbert Lórencz. 2021. Active Directory Kerberoasting Attack: Detection using Machine Learning Techniques. In *ICISSP*.
[15] Arash Habibi Lashkari, Amy Seo, Gerard Drapper Gil, and Ali Ghorbani. 2017. CIC-AB: Online ad blocker for browsers. In *ICCST, IEEE*.
[16] Rafath Samrin and D. Vasumathi. 2017. Review on anomaly based network intrusion detection system. *ICEECCOT* (2017).
[17] Andrey Sapegin. 2019. *High-Speed Security Log Analytics Using Hybrid Outlier Detection.* Ph. D. Dissertation. University of Potsdam, Germany.
[18] Andrey Sapegin, Marian Gawron, David Jaeger, Feng Cheng, and Christoph Meinel. 2017. Evaluation of in-memory storage engine for machine learning analysis of security events. *Concurr. Comput. Pract. Exp.* (2017).
[19] Andrey Sapegin, David Jaeger, Feng Cheng, and Christoph Meinel. 2017. Towards a System for Complex Analysis of Security Events in Large-Scale Networks. *Comput. Secur.* (2017).
[20] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* (2018).
[21] Theodoros Toliopoulos, Christos Bellas, Anastasios Gounaris, and Apostolos Papadopoulos. 2020. PROUD: PaRallel OUtlier Detection for streams. In *SIGMOD*.
[22] Theodoros Toliopoulos, Anastasios Gounaris, Kostas Tsichlas, Apostolos Papadopoulos, and Sandra Sampaio. 2020. Continuous outlier mining of streaming data in flink. *Inf. Syst.* (2020).
[23] Luan Tran, Liyue Fan, and Cyrus Shahabi. 2016. Distance-based outlier detection in data streams. *PVLDB* (2016).
[24] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. 2019. Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. *IEEE Access* (2019).
[25] Stefano Zanero and Sergio M. Savaresi. 2004. Unsupervised Learning Techniques for an Intrusion Detection System. In *SAC*.
[26] Julina Zhang, Kerry Jones, Tianye Song, Hyojung Kang, and Donald E Brown. 2017. Comparing unsupervised learning approaches to detect network intrusion using NetFlow data. In *SIEDS, IEEE*.