A Concept Drift Stream Generator for Intrusion Detection Systems

Gabriele Nicolò Costa[1,†], Alessandra De Paola[1,2,†], Salvatore Drago[3,*,†], Pierluca Ferraro[1,2,*,†] and Giuseppe Lo Re[1,2,†]

[1]Department of Engineering, University of Palermo, Italy
[2]Cybersecurity National Lab, CINI - Consorzio Interuniversitario Nazionale per l'Informatica
[3]IMT School for Advanced Studies Lucca, Italy

Abstract

Intrusion Detection Systems (IDSs) based on machine-learning techniques have become a major research focus, as they are crucial for identifying anomalies in the network traffic logs to detect malicious activity. Although such systems achieve high performance during testing, they experience a decline in accuracy over time when deployed in real-world scenarios due to concept drift. Over time, patterns in both benign and malicious network traffic evolve, rendering the training data obsolete and leading to performance degradation. This has led to a growing interest in concept drift detection and the use of adaptation policies such as online and incremental machine learning. However, testing system performance over time, both for drift detection and adaptation, requires labeled real network datasets that exhibit concept drift, with temporal indications of when the drift occurs. The absence of such datasets has led to the use of synthetic drift data generators, which, however, force researchers to work with datasets that are overly simplistic and insufficiently challenging for machine learning algorithms compared to real network datasets. To overcome this limitation, this work proposes a Concept Drift Stream Generator for Intrusion Detection Systems that, starting from a real network dataset, generates data streams exhibiting concept drift. This enables the evaluation of system performance under realistic concept drift conditions while preserving the complexity of the original dataset.

Keywords

Threat Detection, Online Intrusion Detection System, Machine Learning, Concept Drift, Drift Data Generator

1. Introduction and Related Work

In the past decade, there has been a growing focus on cybersecurity, particularly in network security. The increasing number of devices interconnected in networks, also due to the growing adoption of IoT technologies in different contexts, such as industrial IoT, smart homes and smart cities [1], has made indispensable the need to detect threats to these distributed systems [2]. In such a scenario, Intrusion Detection Systems (IDSs) emerging as a potential solution to protect systems for malicious activities. These systems aim to detect intrusions early enough to raise alerts about malicious activity that deviates from normal traffic. The most prominent approaches for developing IDSs involve the adoption of Machine Learning (ML) and, more broadly, Artificial Intelligence (AI) algorithms. The recent literature [3, 4] showed that these systems are promising tools for assisting network administrators in handling network anomalies.

However, many recent studies [5, 6, 7, 8] have shown that, in the field of cybersecurity, although AI-based systems achieve high performance in experimental evaluations, they experience performance degradation over time due to the phenomenon of concept drift. Over time, patterns in both benign and malicious network traffic evolve, rendering the training data obsolete and leading to performance degradation [9, 10].

Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy
*Corresponding author.
†These authors contributed equally.
gabriele.costa03@community.unipa.it (G. N. Costa); alessandra.depaola@unipa.it (A. De Paola); salvatore.drago@imtlucca.it (S. Drago); pierluca.ferraro@unipa.it (P. Ferraro); giuseppe.lore@unipa.it (G. Lo Re)
0000-0002-7340-1847 (A. De Paola); 0009-0009-0367-0484 (S. Drago); 0000-0003-1574-1111 (P. Ferraro); 0000-0002-8217-2230 (G. Lo Re)

Typically, concept drift is classified into four types based on its temporal characteristics and intensity. An *abrupt drift* or *sudden drift* refers to a sudden change in distribution at a specific point in time. In contrast, *gradual drift* occurs when changes take place gradually over a period of time. In *incremental drift*, samples during the transition period are drawn from both distributions with varying probabilities. Finally, *recurrent drift* refers to the reappearance of past distributions, usually due to seasonality.

From the perspective of a flow-based IDS [11], which analyzes statistical features for each communication within a company or public administration subnet, drift in benign network traffic patterns may arise from the activation of new web services accessible both internally and externally, changes in the network infrastructure or variations in staff behavior. On the other hand, drift in malicious traffic may arise, for instance, from zero-day attacks [12, 13] or adversarial learning attacks [14, 15, 16, 17]. As a result, there has been increasing interest in concept drift detection [18] and the adoption of adaptive machine learning approaches to adjust models to evolving data distributions [19]. Current adaptation techniques can be broadly classified into two categories: detect and retrain, which involves discarding the existing model and training a new one using updated data; and detect and update [20, 21], which incrementally refines the existing model based on new data.

However, evaluating system performance over time, both in terms of drift detection and adaptation, requires labeled real-world network datasets that exhibit concept drift and include precise temporal annotations of drift events. The lack of such datasets has led researchers to rely on synthetic drift generators [22]. However, these concept drift generators lack realistic scenarios: they simulate concept drift by sampling from different datasets (Agrawal Generator [23]) or by perturbing features (LED-Generator [24]). While useful, these generators often produce data that is overly simplistic and fail to capture the complexity and challenges found in real network traffic, limiting their effectiveness for testing machine learning algorithms. In [25], the authors propose a drift generator based on real datasets. However, this system is limited to manipulating only the temporal location of the drift and does not allow control over other key aspects such as the type and intensity, which are critical to system performance and incremental adaptation mechanisms.

To overcome this limitation, this work proposes a Real Dataset-based Concept Drift Stream Generator for Intrusion Detection Systems (*RD-ConceptDriftGenerator*). Starting from a real network dataset, this tool enables the generation of controlled data streams that simulate scenarios affected by specific types and intensities of concept drift. This allows for the evaluation of IDSs while preserving the complexity of the original dataset.

The experimental evaluation, performed on a real network dataset, proves the effectiveness of the generator by showing that it can generate streams with different concept drift characteristics from the same dataset. The results also show how the performance of a machine learning-based IDS is affected over time by the characteristics of the concept drift present in the stream.

The remainder of the paper is structured as follows. Section 2 describes the proposed *RD-ConceptDriftGenerator*. Section 3 outlines the experimental evaluation. Finally, Section 4 presents the conclusions and future research directions.

2. RDConceptDriftGenerator Architecture

This section presents the architecture of *RD-ConceptDriftGenerator*, which consists of two main components: a *concept generator* and a *drift streamer*. Additionally, as shown in Figure 1, the *RD-ConceptDriftGenerator* requires as input the source dataset and a set of parameters that define the desired characteristics of the concept drift in the generated stream (Streamed Dataset).

First, the Concept Generator module divides the original dataset into two macro-clusters, then further divides each macro-cluster into different micro-clusters. These steps are carried out using classical clustering algorithms [26]. When the source dataset has multi-class labels, this step can be guided by creating macro- and micro-clusters aligned with those labels. The macro-cluster phase splits the data into benign and malicious clusters, while the micro-cluster phase creates a further subdivision reflecting different types of malicious traffic and various benign communication behaviors.

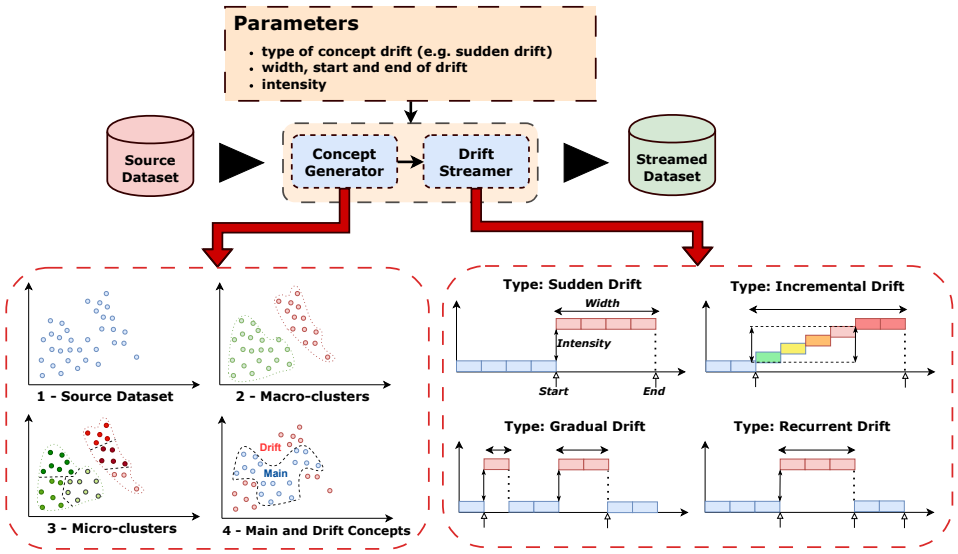

Figure 1: Components of RD-ConceptDriftGenerator.

Then, to create a concept drift with specific characteristics, a cluster similarity metric, such as AMI [27], is computed among micro-clusters within each macro-cluster. Thus, micro-clusters are grouped into two sets: Main Concept and Drift Concept. The Drift Concept is created by selecting the most similar micro-clusters, starting with one micro-cluster from each macro-cluster, until the number of records required by the *width* parameter (expressed as a number of elements) is reached. The remaining micro-clusters form the Main Concept. The goal is to obtain two sets, each containing records from both macro-clusters (both benign and malicious traffic). Additionally, the method aims to maximize the similarity between benign and malicious micro-clusters within each set while minimizing the similarity between the two sets. Once the Main Concept and Drift Concept are formed, the drift elements are assigned an anomaly score based on their deviation from the Main Concept.

Finally, the drift streamer samples from the Main and Drift Concept sets to generate concept drift according to the input parameters (the time in which the drift starts, its width and its ending time). The streamer uses the anomaly scores of the Drift Concept elements to control the desired intensity and type of drift while respecting the *width* parameter.

Note that the Streamed Dataset does not necessarily have the same number of elements as the original dataset. This allows the streamer to create drifts with different characteristics more easily. Furthermore, to avoid biasing the performance of the machine learning model used as an IDS, the streamer samples from the two sets without replacement. Additionally, no data augmentation is applied, ensuring that no unrealistic records are introduced in the Streamed Dataset.

3. Experimental Evaluation

This section highlights the ability of *RD-ConceptDriftGenerator* to introduce concept drift of different types based on a realistic network dataset.

The experiments were conducted using the *CIC-IDS2017* [28] dataset as the source dataset. This dataset offers a comprehensive representation of modern network traffic, including benign traffic and a wide range of attack types, covering various phases and attack methods. These attacks include Distributed Denial of Service (DDoS), Brute Force (SSH and HTTP), Web attacks (XSS, SQL Injection), Infiltration, Botnet traffic, and others, all simulated in a real enterprise environment, with flow-based features and labels.

RD-ConceptDriftGenerator was used to create two streamed datasets, each exhibiting sudden and recurrent concept drift, respectively, using the parameters summarized in Table 1. These represent the

Table 1
Parameters used with *RD-ConceptDriftGenerator* to generate the two streamed datasets with concept drift.

Type	width	start	end	intensity	Total dataset dimension
Sudden	100 000	100 000	200 000	medium	200 000
Recurrent	50 000	100 000	150 000	high	200 000

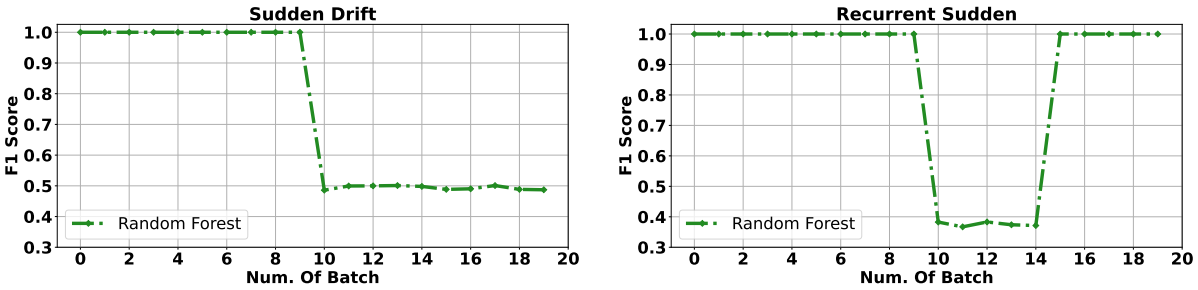

Figure 2: Performance (F1-score over time) of the IDSs tested with two streamed datasets exhibiting different concept drifts, generated by the *RD-ConceptDriftGenerator* from the CIC-IDS2017 dataset.

two main types of drift, as the other types can be derived from them. Specifically, gradual drift can be seen as a recurrent drift with different starting points and widths, while incremental drift is generated by dividing the drift width into multiple sudden drifts with progressively increasing intensity.

To assess whether the two generated datasets contain concept drift with the desired characteristics, they were used to evaluate the online performance of two IDSs. These systems are tested in static mode, which is often used in online learning under concept drift [19] as a baseline to evaluate the impact of concept drift over time. To this end, the machine learning model is trained offline on the first data batch, and its performance, in terms of F1-score, is then evaluated over time, batch by batch, as shown in Figure 2. If the first data batch is representative, the system's performance will remain good and stable until the occurrence of a potential concept drift. A sudden or gradual degradation in performance, on the other hand, will indicate the presence of concept drift. In these experiments, the machine learning model used is Random Forest, which has shown good performance on high-dimensional datasets, even in multi-class settings [29]. Moreover, the two systems were tested using batches of 10 000 samples on a binary task, i.e., distinguishing benign versus malicious traffic.

As shown in Figure 2, in both cases the systems maintain high performance, above 0.9 and close to an F1-score of 1.0, as long as the evaluated records belong to the Main Concept set. Conversely, a sudden drop in performance is observed when concept drift occurs, specifically when samples from the Drift Concept set appear. As can be seen from the batch number (x-axis), the concept drift occurs at batch 10, that is, starting from the 100 000th sample considering batches of 10 000 samples, and it lasts for the expected duration, as reported in Table 1.

For the recurrent concept drift, after the drift ends, performance returns to excellent levels, similar to pre-drift performance. The intensity of the drift significantly affected the systems' performance. In the first case, after the drift, performance stabilized at a value slightly above 0.5, whereas in the recurrent drift case, it oscillates between 0.4 and 0.37, confirming a high drift intensity, which is higher than the intensity set for the sudden drift (medium).

4. Conclusions and Future Work

This work presents *RD-ConceptDriftGenerator*, a novel tool for generating realistic concept drift scenarios in network intrusion detection datasets by leveraging real network data. Unlike existing drift generators, the proposed approach preserves the complexity of real network traffic while enabling precise control over drift characteristics like type and intensity. The experimental evaluation, conducted using two

streamed datasets with sudden and recurrent drift, demonstrated that the generator effectively simulates different drift characteristics, impacting the performance of machine learning-based IDSs as expected. Specifically, IDS performance remained stable before drift and showed significant degradation upon drift occurrence, with recovery observed in the case of recurrent drift. These results highlight the importance of realistic drift simulation for developing and evaluating IDS performance in evolving network environments.

Future work will focus on extending this framework by evaluating the impact of different clustering algorithms, similarity techniques between clusters, and anomaly scoring strategies used by the Concept Generator. Additionally, it may be useful to monitor, through the drift streamer, the balance between benign and malicious records over time, to avoid excessively unrealistic imbalances within a batch.

Acknowledgments

This work was partially supported by the AMELIS project, within the project FAIR (PE0000013), and by the ADELE project, within the project SERICS (PE00000014), both under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] V. Agate, A. De Paola, G. Lo Re, A. Virga, Reliable reputation-based event detection in V2V networks, in: International Conference on Advanced Research in Technologies, Information, Innovation and Sustainability, Springer, 2023, pp. 267–281.

[2] B.-X. Wang, J.-L. Chen, C.-L. Yu, An AI-powered network threat detection system, IEEE Access 10 (2022) 54029–54037.

[3] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, M. A. Khan, Performance analysis of machine learning algorithms in intrusion detection system: A review, Procedia Computer Science 171 (2020) 1251–1260.

[4] V. Agate, F. Concone, A. De Paola, P. Ferraro, G. Lo Re, M. Morana, Adaptive ensemble learning for intrusion detection systems, in: CEUR Workshop Proceedings, volume 3762, 2024.

[5] A. Augello, A. De Paola, G. Lo Re, Hybrid multilevel detection of mobile devices malware under concept drift, Journal of Network and Systems Management 33 (2025) 36.

[6] S. Agrahari, A. K. Singh, Concept drift detection in data stream mining : A literature review, Journal of King Saud University - Computer and Information Sciences 34 (2022) 9523–9540. URL: https://www.sciencedirect.com/science/article/pii/S1319157821003062. doi:https://doi.org/10.1016/j.jksuci.2021.11.006.

[7] V. Agate, A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Enhancing IoT network security with concept drift-aware unsupervised threat detection, in: 2024 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2024, pp. 1–6.

[8] A. Augello, A. De Paola, G. L. Re, M2fd: Mobile malware federated detection under concept drift, Computers & Security (2025) 104361.

[9] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM computing surveys (CSUR) 46 (2014) 1–37.

[10] V. Agate, S. Drago, P. Ferraro, G. Lo Re, Anomaly detection for reoccurring concept drift in smart environments, in: 2022 18th International Conference on Mobility, Sensing and Networking (MSN), IEEE, 2022, pp. 113–120.

[11] M. F. Umer, M. Sher, Y. Bi, Flow-based intrusion detection: Techniques and challenges, Computers & Security 70 (2017) 238–254.

[12] T. Zoppi, A. Ceccarelli, T. Puccetti, A. Bondavalli, Which algorithm can detect unknown attacks? comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection, Computers & Security 127 (2023) 103107. URL: https://www.sciencedirect.com/science/article/pii/S0167404823000172. doi:https://doi.org/10.1016/j.cose.2023.103107.
[13] A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Detecting zero-day attacks under concept drift: An online unsupervised threat detection system, in: CEUR Workshop Proceedings, 8th Italian Conference on Cybersecurity, ITASEC, volume 2024, 2024.
[14] K. He, D. D. Kim, M. R. Asghar, Adversarial machine learning for network intrusion detection systems: A comprehensive survey, IEEE Communications Surveys & Tutorials 25 (2023) 538–566.
[15] F. Batool, F. Canino, F. Concone, G. Lo Re, Morana, A black-box adversarial attack on fake news detection systems, in: CEUR Workshop Proceedings, volume 3731, 2024.
[16] T. S. Sethi, M. Kantardzic, Handling adversarial concept drift in streaming data, Expert systems with applications 97 (2018) 18–40.
[17] F. Concone, S. Gaglio, A. Giammanco, G. Lo Re, M. Morana, Adverspam: Adversarial spam account manipulation in online social networks, ACM Transactions on Privacy and Security 27 (2024) 1–31.
[18] F. Hinder, V. Vaquet, B. Hammer, One or two things we know about concept drift—a survey on monitoring in evolving environments. part a: detecting concept drift, Frontiers in Artificial Intelligence 7 (2024) 1330257.
[19] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE transactions on knowledge and data engineering 31 (2018) 2346–2363.
[20] M. A. Shyaa, N. F. Ibrahim, Z. B. Zainol, R. Abdullah, M. Anbar, Reinforcement learning-based voting for feature drift-aware intrusion detection: An incremental learning framework, IEEE Access (2025).
[21] F. Camarda, A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Managing concept drift in online intrusion detection systems with active learning (2025).
[22] J. Montiel, J. Read, A. Bifet, T. Abdessalem, Scikit-multiflow: A multi-output streaming framework, Journal of Machine Learning Research 19 (2018) 1–5.
[23] R. Agrawal, T. Imienski, A. Swamy, Database mining: A performance perspective, ieee tran, On Knowledge and Data Engg (1991).
[24] L. Breiman, J. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Routledge, 2017.
[25] B. Lin, C. Huang, X. Zhu, N. Jin, Realdriftgenerator: A novel approach to generate concept drift in real world scenario, in: 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2024, pp. 1124–1129.
[26] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Annals of data science 2 (2015) 165–193.
[27] N. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, Properties, Normalization and Correction for Chance 18 (2009).
[28] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, et al., Toward generating a new intrusion detection dataset and intrusion traffic characterization., ICISSp 1 (2018) 108–116.
[29] V. Agate, D. Felice Maria, A. De Paola, P. Ferraro, G. Lo Re, M. Morana, A behavior-based intrusion detection system using ensemble learning techniques., in: ITASEC, 2022, pp. 207–218.