Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)

t − robust spaces with dynamic graphs metrics for mitigating concept drift in attack detection

Julien Michel[a,b,*], Pierre Parrend[a,b]

[a] *Laboratoire de Recherche de l'EPITA, 14-16 Rue Voltaire, Le Kremlin-Bicètre 94270, France*
[b] *ICube, UMR7357, Université de Strasbourg CNRS, Strasbourg 67000, France*

Abstract

Concept drift is of primary concern for the detection of attacks in Internet traffic networks. Those networks register a high diversity of heterogeneous activity which change with usage, thus nurturing an evolutive environment. Detecting attacks is the action of dissociating attacks from the other data in the environment in a discriminating classification. However, both the modification of attack behaviors through time and the evolution of the environment are disruptive of the thin boundaries drawn by learning models. For the evaluation of sustainability and trustworthiness of the detection system, concept drift detection approaches have emerged. Those approaches can necessitate a significant amount of time and resources to mitigate the effect of concept drift thus detected. In this paper we propose an end-to-end approach in the production of a concept drift robust feature space and its evaluation for learning models. We define t-robustness as a quantification of feature drift, design a feature engineering approach using initial learning conditions to mitigate the effect of concept drift on learning models and define metrics to evaluate the stability of learning models. We apply our approach on the UGR16 dataset enriched with graph community metrics. The features produced by our graph community metrics extraction approach have very few dependencies in their construction, which make them relevant for time robust attack detection. Hence, we produce a feature space which produces more stable detection models through time.

Keywords: attack detection; time robustness; concept drift; feature engineering; graph community;

1. Introduction

Detecting anomalies in evolving systems affected by **concept drift** is a complex challenge. Internet networks are such systems, and ensuring sustainable attack detection despite dynamic user behaviors presents a significant problem. We define concept drift in detection as the **evolution of detection targets** within an **ever-changing environment** over time. This phenomenon both causes and results from significant changes in data features. By nature, concept drift is

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: author@institute.xxx

closely linked to the feature space – specifically, the value space of communication packet parameters such as latency or size. Consequently, it is logical to evaluate this feature space quantitatively over a given time frame. A feature space experiencing high concept drift would not effectively represent the general behavior of the data; instead, it would only capture patterns within the specific time period in which learning occurs. The challenge, therefore, is to identify a feature space that best characterizes detection targets within the environment at any given time. Given the nature of concept drift and its impact on attack detection performance, a feature space less affected by concept drift could enable more time-robust machine learning models. The key requirement for such a feature space is its ability to represent attack behaviors in a way that remains distinct from normal environmental data over time. Ideally, the feature space should consistently maintain its discriminative power for identifying attacks, regardless of external factors.

In this study, we consider both original packet features and derived graph community metrics [28] as potential candidates for time-robust features. There are two main motivations for using graph community metrics. First, topology-based graph metrics have relatively few dependencies in their construction, making them more resilient to data evolution [18]. Fewer dependencies generally lead to greater robustness against concept drift. Second, attack behaviors influence network structures, such as communities, in a way that differs from typical heterogeneous network activity. This makes graph-based metrics more resistant to attacker strategies, as they do not rely on easily reconstructed information. To define and evaluate robust feature spaces, we structure our research around three key research questions (RQ): **RQ1:** How can concept drift be quantified in communication data? **RQ2:** How can this quantification be used to develop a feature space that remains robust over time? **RQ3:** Are graph community metrics viable candidates for constructing a long-lasting feature space for attack detection? To address these questions, we assess detection performance using feature sets enriched with graph community metrics in a concept drift context. We define metrics to quantify drift relative to each feature in the feature space and evaluate their stability within the dataset. By combining graph community metrics with feature stability metrics, we construct a time-robust feature set and evaluate its detection performance under concept drift conditions.

The remainder of this paper is structured as follows. Section 2 presents related research.Section 3 defines the metrics and requirements for the proposed robust feature spaces. Section 4 provides the evaluation methodology, and Section 5 the implementation details. Section 6 presents the results of evaluation, and Section 7 discusses their implications for validating our approach. Section 8 concludes the paper.

2. State of the art

In machine learning, concept drift refers to a change in the relationship between detection targets – *i.e.*, the important elements in the data—and their environment, which consists of data typically considered irrelevant for detection [26]. This differs from data drift, which represents a shift in distribution between a training set and a testing set [3]. However, the underlying behaviors in the data remain the same, making concept drift more challenging to detect. As a result, various **Concept Drift Detection** (CDD) approaches have emerged to characterize concept drift.

Concept drift analysis. The Parallel Histograms through Time (PHT) model [11] is defined to visualize and locate drift across all features of a dataset. It analyzes changes by comparing the distribution of each relevant feature within sliding windows centered around the mean. QuadCDD [25], as its name suggests, characterizes concept drift using four key parameters: 1) *Drift start*: The last point in the data where the current data profile remains consistent; 2) *Drift end*: The point where the detection model's accuracy falls below a chosen threshold; 3) *Drift type*: The nature of change between drift start and drift end, classified as incremental, abrupt, recurring, or gradual; 4) *Drift severity* (or intensity): The impact of drift on the detection model, quantified using accuracy rates. This approach aims to maintain model stability in data streams by adjusting the model to the detected drift. The Typicality and Eccentricity Data Analytics-based Concept Drift Detector (TEDA-CDD) [19] identifies concept drift in an unsupervised manner with low computational complexity and minimal memory usage. It employs the Jaccard Index for similarity, constructing a reference model that is then compared to the current model using Euclidean metrics. If the current model deviates significantly from the reference model, it is classified as abnormal, indicating the presence of concept drift.

When detection systems encounter concept drift, their efficiency gradually declines, leading to a significant drop in detection performance. This deterioration occurs due to changes in both the detection target and its environment [12]. To address this issue, various methods have been developed to mitigate performance loss after detecting drift. Concept drift necessitates changes in how models learn and adapt [16]. However, there is no universal adaptation strat-

egy, as the optimal approach depends on the model, the type of drift, and its intensity. The most common adaptation strategy is performance-driven, which updates the learning model based on performance loss – typically measured by accuracy degradation. However, this method is highly dependent on the detection model. Another widely used approach is distribution-driven, which measures the distance between two data profiles relative to the type of drift. While it is possible to develop tailored solutions for specific drift scenarios and learning models, such approaches require identifying both the moment of drift and the underlying drift process. In real-world data streams, this is computationally expensive and often unsustainable. This challenge highlights the need for robust, long-lasting, and generalizable solutions. Adaptive learning [29], which leverages multiple data streams as input for concept drift mitigation, has demonstrated promising results. By using correlations among different streams as a robust parameter, this approach has outperformed baseline models in six out of eight evaluated datasets under concept drift conditions. Additionally, concept drift is closely linked to the statistical evolution of features over time, as features serve as the primary interface between data and learning models. Different feature spaces for the same dataset can exhibit varying degrees of concept drift severity [14]. When the statistical evolution of features reaches a point where it negatively impacts detection performance, feature drift occurs within a subset of the feature space relevant to the learning model [4].

Concept drift in attack detection. Attack detection environments are particularly susceptible to concept drift because attackers continuously adapt in real time to new tools and attack vectors as soon as they become available [15]. Additionally, attackers modify their tactics as they are tracked by Security Operations Centers (SOCs) [22]. Moreover, both new attacks and benign behaviors emerge dynamically throughout the lifecycle of network environments [1]. As a result, detecting and mitigating concept drift is a critical security concern.

In their review of concept drift detection in intrusion detection systems (IDSs), Shyaa et al. [23] highlight that detecting and categorizing concept drift remains a significant challenge due to key issues such as high data dimensionality, class imbalance, and learning model optimization. They also emphasize the lack of research addressing the intersection of concept drift and feature drift. Chen et al. [8] investigate the impact of feature drift on detection performance in data streams using DroidEvolver [27], an online learning model for malware detection. Their approach dynamically adjusts the feature set based on real-time data stream analysis. When data drift is detected, the initially selected features are reevaluated, and a new feature-space selection is performed using LinearSVM to better align with the evolving data profile. To create features resistant to concept drift, the FeSAD framework [10] proposes selecting features that extend the lifespan of a learning-based ransomware detector. This approach quantifies drift severity using the Heterogeneous Euclidean Overlap metric and subsequently retrains the model using a core feature set. Whenever drift intensity surpasses a predefined threshold, a genetic algorithm is used to generate a new set of features [9].

Most current research focuses on detecting concept drift based on its severity and type, which then necessitates frequent model updates – a process that requires substantial time and resources [2]. However, the role of time-robust features in ensuring the long-term effectiveness of machine learning models should not be overlooked [24]. The core limitation lies in the approach to concept drift detection, which often results in outdated models that require updates. Since concept drift is inherently tied to the feature space, different features exhibit varying degrees of resistance to drift in the context of attack detection. Attacks, by their very nature, exhibit inherent behaviors linked to their objectives.

Thus, we hypothesize that it is possible to construct a stable feature space by analyzing the statistical evolution of features, or feature drift. Our objective is to identify features that consistently represent attack behaviors over time, ensuring robust and long-lasting detection performance.

3. Tackling concept drift

Whereas concept drift is quite well characterized in the literature, actual metrics to measure it prove to be insufficient. We therefore elicit the requirements for evaluation and propose a novel metric *t-robustness* and related distance properties to address this issue.

Quantifying concept drift. The analysis of concept drift follows two main steps: extraction of feature states for the dataset under analysis, and extraction of stability metrics to quantify the concept drift itself. To quantify concept drift, we introduce four key metrics: *median-centered cut*, *State Distance*, *t-equivalency* and *t-robustness*.

Feature states represent the statistical state of a feature during a time interval δ_t. 2 features states are considered: standard deviation σ and median-centered cut.

Median-centered cut is a clustering of feature values during δ_t. It is parametrized by n the number of clusters and p the percentage difference from the median.

State Distanceis the proportional distance between two feature states. State distance is a stability metric. With the $pt1_k$ the proportion of bins at time $t1$ and the $pt2_k$ the proportion of bins at time $t2$ and n the number of bins:

$$Sd = \sum_{k=1}^{n} |pt2_k - pt1_k| \tag{1}$$

t-equivalency or time-equivalency, is average proportional difference of the median-centered cut over n time δ_t. t-equivalency is a stability metric. With Sd_k and $1 < k \leq n$ the state distance between states k and 1:

$$equiT = \overline{|Sd_k|} \tag{2}$$

t-robustness or time-robustness, is the minimum between average state distance averaged with average standard deviation difference and average t-equivalency over a distribution reduced to the [0,1] interval. t-robustness is a stability metric. With $1 \leq k < n$, Sd_k the state distance and $Stdd_k$ the standard deviation difference reduced to the [0,1] interval between state k and $k+1$ and $equiT_{k'}$ the t-equivalency from state k' with $1 \leq k' < (n-t)$:

$$robusT = \min\left(\frac{1 - \overline{|Stdd_k|}}{2} + \frac{1 - \overline{|Sd_k|}}{2}, 1 - \overline{equiT_{k'}}\right) \tag{3}$$

Requirements to evaluation of robustness to concept drift. We identify the following requirements for evaluating robustness to concept drift: *;Quantifying features drift:* For each feature in the dataset, we require a stability measure to evaluate its robustness over time. *;Performance of feature space through concept drift:* We need to evaluate how attack detection models perform using a given feature space when concept drift occurs. *;Feature quality relative to concept drift:* For each feature, within a detection environment, we must determine which features should be included in a robust feature set.

We also define the following conditions for the dataset:
Continuity: The data must cover a sufficiently long time period and contain enough data within this period.
Time distance: The start and end times must be spaced far enough to reveal significant changes in data behavior.
Feature drift: The original feature space must exhibit some degree of drift, even if only partially.
These requirements are essential for a reproducible evaluation of concept drift mitigation measures.

Community metrics for attack detection. As initial candidates for robust features, we evaluate graph community metrics, which have minimal dependencies and require only the network's end-to-end topology and time [13]. Moreover, community structures within network topology are closely linked to attack behaviors [21]. Therefore, we expect graph community metrics to serve as stable features for detecting attacks. To construct these metrics, we generate two types of graphs from flow data, here NetFlow captures. The first type consists of graphs where nodes represent IP addresses, and edges correspond to individual communications (each NetFlow record). The second type extends this by defining nodes as pairs of IP addresses and ports. We use the Louvain algorithm [6] to partition these graphs into communities. From these partitions, we compute various graph community metrics, including the *average degree* of communities, the *number of nodes*, *density*, *expansion* [28], and the *NEDIndex* [20]. To capture temporal variations, we apply time windowing of 5 and 20 minutes to produce dynamic community metrics, including changes in average degree ($\Delta degree$) and density ($\Delta density$) between time t and $t+1$ for a same community. Additionnaly, we compute $Stability = \frac{|V_t \cap V_{t+1}| - |(V_t \cap \bar{V}_{t+1}) \cup (V_{t+1} \cap \bar{V}_t)|}{|V_t \cup V_{t+1}|}$ as the ratio of similarity between two consecutive states of the same community.

4. Methodology

As highlighted in the literature, existing approaches to mitigating concept drift primarily focus on detecting its type and severity, followed by retraining the learning models. A key limitation of these methods is the substantial time and computational resources required to perform such operations effectively. To address our research questions, we propose a comprehensive methodology for the extraction of robust features, as illustrated in Figure 2 in Annexes. The process begins with enriching the base dataset by generating derived features that are potential candidates for improving robustness. Next, feature states are extracted from this enriched dataset, and their temporal robustness is assessed. Based on this evaluation, a robust feature set is selected – excluding features found to degrade robustness

in the presence of concept drift. We then assess the performance of learning models trained for a limited time period using these robust features, comparing the results to baseline models trained on the original feature set.

To explore the impact of concept drift on model performance, we construct multiple scenarios, each involving detection models subjected to varying levels of concept drift. Within each scenario, we compare models using different feature sets to observe variations in behavior under identical learning and evaluation conditions. This performance-driven evaluation enables both the validation of concept drift presence in the data and the identification of features that contribute to improved temporal robustness. Subsequently, we conduct a distribution-driven feature drift evaluation across the entire dataset to quantify the extent of concept drift affecting each feature. *Feature states* are computed at regular time intervals, and drift is quantified via *state distance* and *t-equivalency* metrics between consecutive intervals. To construct a time-robust feature set, we integrate (i) the magnitude of drift for each feature – termed *t-robustness* – and (ii) its relevance to detection performance in the base scenario, measured via information gain. Features satisfying both criteria are selected as candidates for building models with sustained performance over time. These selected features are then used to retrain models across all scenarios. We compare their performance against prior models to evaluate improvements resulting from the use of time-robust features.

The primary goal of this methodology is to circumvent the need for real-time concept drift detection and continual model retraining in streaming environments. By extracting features that are inherently more robust to temporal drift within given attack scenarios, detection models can maintain reliable performance over extended periods. We evaluate the effectiveness of our approach using the following metrics:

Best performance at worst: For each detection model across all learning scenarios, we compute the Matthew Correlation Coefficient (MCC) over each time period and report the worst-case performance.

Retained expectancy: For each model, we track the ratio of performance at each time interval to the initial performance, capturing the model's ability to retain its detection capability over time.

Detection rate difference: For each prediction, we calculate the difference to initial detection rates across intervals.

We define **MCC rate difference**, akin to accuracy rate difference [16] as an absolute measure of learning stability:

$$\Delta MCC = \frac{MCC_1 - MCC_2}{MCC_1} \tag{4}$$

The **retained expectancy rate difference** is defined as a relative measure:

$$\Delta Retained\ expectancy = 1 - \frac{\sum_1^k MCC_k}{MCC_1} \tag{5}$$

These metrics quantify the degradation of detection performance under concept drift, providing a basis for evaluating the long-term robustness of the selected feature sets.

5. Implementation

The evaluation of the proposed framework for quantifying and mitigating concept drift is conducted across three learning scenarios and a control scenario. The learning scenarios differ in their choice of reference periods for initial training, after which the detection algorithms are exposed to concept drift without any model updates. In contrast, the control scenario involves training and testing over the entire time span of the dataset, with the reference data partitioned into training and test sets. For this evaluation, we use the UGR'16 dataset [17], which is known to exhibit pronounced concept drift behavior, making it well-suited for assessing the robustness of detection models over time.

Analysis scenarios. We evaluate detection performance across different scenarios using an XGBoost model [7], comparing the results obtained from both the base dataset and an enriched feature set incorporating graph community metrics.

Learning Scenario 1: The two first days the UGR16 dataset are assumed to be known. The model is trained on this data and then applied to the remainder of the dataset without updates.

Learning Scenario 2: The first five days (corresponding to the fifth week of July in the dataset) are used for training. The model is then applied to the rest of the data, following the same procedure as in Scenario 1.

Learning Scenario 3: The model is tested on each segment of the UGR'16 test dataset, with training data always drawn from the immediately preceding time frame. This scenario assumes full knowledge of prior periods.

Control Learning Scenario: This scenario represents a setting free from concept drift. Each data segment (excluding the first) is split into 80% training and 20% testing sets using 5-fold cross-validation, solidifying results.

This process, illustrated in Figure 2 in the Annexes, includes the following steps: 1) Feature state extraction across all time intervals; 2) Computation of state distances and t-equivalency between consecutive intervals; 3) Calculation of t-robustness for each feature, combining both state distance and t-equivalency measures; 4) Feature ranking and selection: features with t-robustness below a predefined threshold are discarded, yielding a reduced, time-robust feature set. Using this refined feature space, we reapply Scenarios 1–3 (and include the control scenario as Scenario 4) to assess detection performance. The results obtained from models trained on time-robust features are then compared with those of the baseline models. This comparison enables us to analyze how detection performance evolves across scenarios and to quantify the performance gains achieved through the proposed robustness-driven feature selection.

Concept drift in UGR16 dataset. The UGR'16 test dataset contains network traffic data collected between the fifth week of July 2016 and the fourth week of August 2016. This dataset was chosen for two main reasons. First, it offers a continuous time series over its period without missing intervals, making it well-suited for evaluating detection performance across different time segments. A dataset with sufficient temporal continuity and duration is essential for assessing the effects of concept drift, and UGR'16 fulfills this requirement. Second, meaningful evaluation of concept drift necessitates an evolving background distribution. In the case of UGR'16, the background traffic originates from real-world network traces collected by an Internet security provider, ensuring the data is both realistic and representative of the kinds of drift encountered in practical scenarios. For the purposes of our experiments, each weekly file in the UGR'16 test dataset is split into two segments along the time axis. All segments—except the first chronological one—are used as test frames in our evaluation.

6. Evaluation

The evaluation is structured into four key steps, aimed at assessing the relative performance of robust feature sets, as defined by our methodology, in a classification task involving imbalanced data under the influence of concept drift. The classification task focuses on the detection of cyberattacks within core network traffic. The evaluation proceeds as follows : 1) Baseline Performance Assessment: We first evaluate the detection performance of models trained under the control scenario, serving as a baseline; 2) Robust Feature Selection: We apply our methodology to identify features that exhibit temporal robustness with respect to concept drift; 3) Robustness Evaluation under Concept Drift: Using the selected robust features, we assess the learning performance under the three learning scenarios defined in Section 6.1, in addition to the control scenario; 4) Analysis of Feature Space Properties: Finally, we compute and analyze the statistical properties of the resulting robust feature spaces to complement and validate the performance evaluation.

Baseline evaluation on control learning scenario. In the *control learning scenario* 3.d in Annexes, where models are not exposed to concept drift, we observe a *performance ceiling* indicative of optimal detection capability under stable conditions. In this setup, detection is evaluated using 5-fold cross-validation applied independently to each time segment. Under these conditions, the feature set enriched with graph community metrics (referred to as the GC set) achieves the highest overall performance, with an **average MCC** of approximately 0.9843 across the dataset. In comparison, the base feature set yields a lower **average MCC** of 0.9336. The time-robust (t-robust) feature set also demonstrates strong performance, achieving an average MCC of 0.9750, indicating that it retains high classification quality even when optimized for temporal robustness.

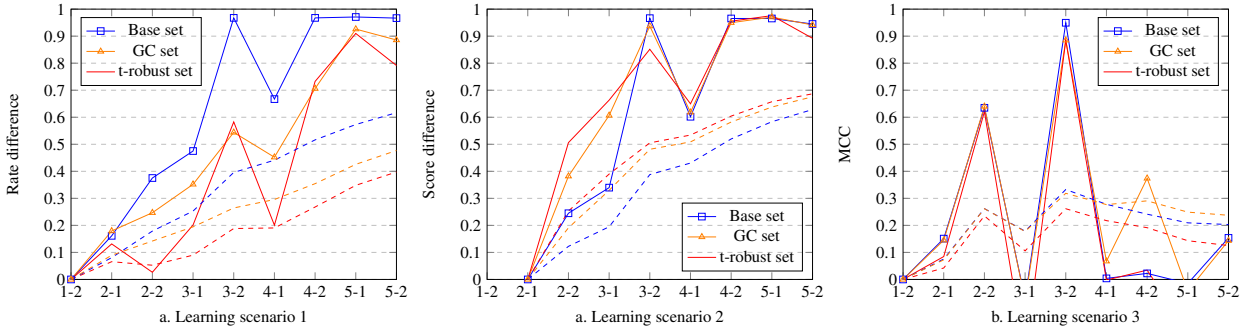
Selection of robust features. To construct a robust dataset, a limited set of features must be selected. The process for feature selection is outlined in Table 1, based on two key criteria: t-robustness of each feature and its coverage within the XGBoost model trained on the 1-1 time interval. The selection process follows these steps: 1) Pre-selection: Features with a t-robustness greater than 0.75 are initially chosen; 2) Coverage filtering: Features that exhibit a coverage greater than 100 are retained; 3) Correlation check: Features are removed if they are correlated above 90% with another feature that has higher coverage, or with a feature that has already been discarded. It is noteworthy that both graph community-based features and features from the original set are capable of achieving sufficient t-robustness for selection while maintaining high coverage values. Specifically, among the graph community metrics, those derived from graphs where IP addresses and ports are treated as nodes demonstrate higher t-robustness compared to other metrics.

Features	1-1 Coverage	t-robustness	Selected
ToS	1157	0.99984	Yes
Source Port	10544	0.96945	Yes
Nb_of_nodes_c_ipport5	70	0.93705	Too low coverage
Nb_of_edges_c_ipport5	167	0.93680	Too correlated
delta_Node_ipport20	102	0.92512	Yes
Duration	458	0.90899	Yes
delta_expansion_ipport5	0	0.89568	Too low coverage
delta_Node_ipport5	17017	0.89492	Yes
edges_dist_ipport5	290	0.88725	Yes
stability_ipport5	203	0.88429	Yes

Table 1. Highest t-robustness features in UGR16 dataset with coverage from XGBoost training on 1-1 time interval and logic for dropping columns: If the feature has coverage value under 100 in training or too much correlation to a feature of higher coverage or dropped then it is dropped

Robustness to concept drift with graphs and robust features. We compare detection performance across *Scenarios 1 to 3* for the original dataset and the dataset enriched with derived features, specifically graph community metrics. As shown in Figure 3 in Annexes, which displays the MCC of the XGBoost models for each scenario, we observe significant differences in model behavior. Learning Scenarios 1 and 2 share a similar structure, with the only distinction being the amount of labeled data at the start. In Scenario 1, only the first two days are labeled, corresponding to the 1-1 time interval, while in Scenario 2, the entire fifth week of July (1-1 and 1-2) is labeled. For models trained on the base feature set, performance remains almost identical across both scenarios. However, when the graph community metrics feature set is used, performance diverges considerably, with an average MCC difference of 0.2227 in favor of Scenario 1. Learning Scenario 3, where models are trained on the time interval immediately preceding detection, exhibits more pronounced short-term drift in the data, highlighting periods where data behavior changes significantly. For instance, a sharp performance drop occurs for both models between time intervals 3-1 and 3-2. While both models display similar performance trends in this scenario, the GC set model experiences a notable performance drop at 4-2, compared to the base set model. An interesting observation across all models in Scenarios 1 to 3 is a significant degradation in detection performance at 3-2, except for the Scenario 1 GC set model and the t-robust model. These models retain MCC scores of 0.4165 and 0.3164, respectively, while the base set model drops to 0.0297, indicating the robustness provided by the graph community features and temporal robustness metrics in handling concept drift.

Fig. 1. MCC rate difference (plain lines) and retained expectancy rate difference (dashed lines) over learning scenario 1-3. Lower value indicate performance closer to initial detection.



Properties of robust feature spaces. We evaluate specific properties of performance-driven concept drift robustness in feature spaces, focusing on the following metrics:

Best Performance at Worst. : For each model and learning scenario, as illustrated in Figure 3 in Annexes, we assess its worst-case performance. Notably, in Scenario 1, the base set model experiences a significant drop to an MCC of

0.0267 at 5-1, whereas the t-robust feature set model at the same time interval achieves a higher MCC of 0.0682, which represents its worst performance in this scenario. A similar pattern is observed in Scenario 3, at time period 3-2, where the base set model drops to an MCC of 0.0464, while the t-robust set model maintains an MCC of 0.0875.

Retained Expectancy. : For each model in Learning Scenarios 1 to 3, we evaluate the proportion of the current performance relative to the initial performance at 1-2 or 1-3 for scenario 2. This is represented by the dashed lines in Figure 3 in Annexes. For Scenarios 1 and 2, the tendency across all models is a gradual decline in retained expectancy over time.

Detection Rate Difference. : We assess the *MCC rate difference* and *Retained Expectancy rate difference* for each model across Scenarios 1 to 3. As shown in Figure 1, the MCC rate difference for the t-robust feature set is consistently lower than that of the other models, particularly in later time intervals.

These results validate the proposed methodology, demonstrating that the robust feature spaces significantly enhance detection performance in the presence of concept drift. Furthermore, the advantage of incorporating topological information, such as graph community metrics, is clearly confirmed.

7. Discussion

We now evaluate the contribution of the proposed robust feature spaces, first *wrt.* their performance in detection tasks, then *wrt.* their capability to actually handle the concept drift challenges. The evaluation of current experiments and results leads us to the elicitation of novel research challenges, which will pave the way to a still better understanding of these issues, which are key to building lasting security detectors.

Detection performance. The proposed framework brings improvements *wrt.* the evaluation of detection performance in the presence of concept drift through an explicit quantification for feature drift and a significant improvement of detection performance in the presence of characterized concept drift. It defines a novel approach for evaluating feature quality in this context.

Quantifying features drift: While concept drift cannot be fully characterized solely through feature drift, the latter is a direct manifestation of the former. To quantify drift at the level of individual features, we employ both state distance and t-equivalency measures. These metrics allow for the independent assessment of drift per feature. From this, we derive a stability score, t-robustness, which captures the resilience of each feature within the feature space. t-robustness is bounded in the interval [0,1], enabling an interpretable comparison and ranking of features according to their drift behavior (see Table 1).

Performance of feature space through concept drift: THe performance evaluation (Figure 3 in Annexes) indicates that the UGR16 dataset presents an environment subject to concept drift. Across multiple learning scenarios, we observe that the XGBoost detection model—despite initially achieving similar performance—exhibits divergent behavior over time on identical data subsets. This is particularly evident in Scenario 1, where models trained on the base set and GC set yield comparable MCC scores at time step 1–2, but diverge significantly by time step 3–2. In this case, the GC set exhibits greater temporal stability. In contrast, Scenario 2, which extends the training data of Scenario 1, produces notably different results for the GC set model during the same evaluation window. Given that Scenario 2 builds directly on Scenario 1, one would expect similar performance from the GC set model. This divergence motivates the introduction of the t-robust set, evaluated using two criteria: Best Performance at Worst and Detection Rate Difference. These metrics highlight a key property—feature space stability—under sufficient training conditions.

Feature quality relative to concept drift: We evaluate feature quality with respect to a defined detection objective. While drift resistance is necessary, it is not sufficient on its own. As a counterexample, a feature with a constant value over time is inherently stable but contributes nothing to detection performance. Therefore, effective feature selection for concept drift robustness must balance stability and predictive utility. To this end, we leverage feature coverage metrics from XGBoost during an initial training phase on the complete feature space of a dynamic graph community-enriched dataset. This dataset satisfies the initial conditions across all evaluated learning scenarios, making it an appropriate foundation for our analysis.

Evaluation of concept drift robustness. The characterization of robust feature spaces through $t-robustness$ provides a principled approach for identifying and understanding concept drift, as well as evaluating the learning capacity of detection algorithms over time. Below, we summarize how this methodology supports the three research questions that guide this study.

RQ1: How to quantify concept drift in data? To quantify concept drift, we employ a series of learning scenarios with UGR16 dataset and conduct a performance-driven evaluation of the XGBoost model. We then compute the feature drift of each feature by combining two complementary metrics: *statedistance* and $t-equivalency$. These measures enable us to assign a drift value to each feature, providing a data-driven basis for evaluating temporal stability.

RQ2: How to use this quantity to set up a new feature space, more robust through time? Using the computed *statedistance* and $t-equivalency$ values, we derive a $t-robustness$ score for each feature. This score is then combined with initial training-phase insights – specifically, the information coverage value from the XGBoost model – to assess each feature's relevance under original detection conditions. By jointly considering both robustness and importance, we construct a t-robust feature space, which is designed to maintain high detection performance over time.

RQ3: Are graph community metrics relevant candidates to build such a feature space long-lasting for attack detection? Experimental results from our t-robust feature space analysis confirm the relevance of graph community-based features in long-term detection scenarios. Specifically, in the context of internet traffic data (e.g., network flow records), features derived from graph communities formed using combinations of IP addresses and port numbers (as node identifiers) exhibit higher $t-robustness$ scores than analogous metrics derived from full-graph structures or those using only IP addresses as nodes. This highlights the potential of dynamic graph community metrics for enhancing the resilience and longevity of detection models in evolving network environments.

Open questions. Beyond the validation of the proposed approach, performed evaluations lead to the elicitation of further research questions, which draw a research roadmap for the following of our study and for the community:

Quantification of concept drift in data: 1) How to have an absolute quantification of concept drift which is more than based on feature drift quantification?; 2) How to optimize this approach parametrization by fine-tuning?

Extraction of robust feature spaces: 1) How the learning approach parametrization process can be included in concept drift mitigation?; 2) How to add new dimension in the datasets while gaining robustness to concept drift?

Identification of relevant derived features as candidates for robust feature spaces: How to adapt the methology to be suitable for unsupervised approaches?

8. Conclusion

In this work, we proposed $t-robust$ spaces, that represent a modified data learning problem through properly selected features as well as implicit features that are derived from the original traffic traces. This selection process aims to craft data that support a learning prediction more robust through the time, and less fragile to concept drift. Previous approaches from the literature focus on enhancing the learning model themselves, whereas we demonstrate that building time-lasting detection models strongly depends on the stability of the data features that are considered rather than in the models themselves. Indeed, as shown by the adversarial problem of machine learning, learning robustness heavily relies on the underlying data more than on the actual learning model [5].

These $t-robust$ spaces are supported by a feature engineering approach consisting of following steps: extraction of derived features to complement base features from the target dataset, extraction of the feature states, computation of proposed $t-robustness$ from intermediate metrics *state distances* and $t-equivalency$, and finally detection of target properties, in our case cyberattacks against core network traffic. Through comparison with a baseline detection pipeline, we demonstrate 1) a strong improvement in robutness in time of the detection; 2) that dynamic graph metrics provide a significant improvement against base netflow features.

For validating our proposal, we compared in different learning scenarios, three feature spaces associated with an XGboost model: the base set, the graph community set and the t-robust set. The models using the t-robust set are on average more stable through time, in particular for learning scenario 1 where the *retained expectancy* at the last time interval of test remains at **0.6025** for evaluation with the UGR16 dataset, while the GC set and Base set are respectively at **0.5230** and **0.3831**. This is an important property toward the trustworthiness of a detection system. However, there is a necessity in understanding better the parameters for selection as the results are very heterogeneous depending on the scenario. The t-robust set model has better ceiling performance than the base set model as observed in the control scenario. Therefore, prospective works in fine tuning criteria for selection parameter and cementing distances between feature is a prime concern. Thus, we need to set up more controlled environments in which we strengthen characterization of the relations between concept drift, feature drift and detection model.

References

[1] Adepu, S., Mathur, A., 2016. Generalized attacker and attack models for cyber physical systems, in: 2016 IEEE 40th annual computer software and applications conference (COMPSAC), IEEE. pp. 283–292.

[2] Agrahari, S., Singh, A.K., 2022. Concept drift detection in data stream mining: A literature review. Journal of King Saud University-Computer and Information Sciences 34, 9523–9540.

[3] Ali Abdu, N.A., Basulaim, K.O., 2024. Machine learning in concept drift detection using statistical measures. International Journal of Computers and Applications 46, 281–291.

[4] Barddal, J.P., Gomes, H.M., Enembreck, F., Pfahringer, B., 2017. A survey on feature drift adaptation: Definition, benchmark, challenges and future directions. Journal of Systems and Software 127, 278–294.

[5] Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D., 2006. Can machine learning be secure?, in: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pp. 16–25.

[6] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008, P10008. URL: https://dx.doi.org/10.1088/1742-5468/2008/10/P10008, doi:10.1088/1742-5468/2008/10/P10008.

[7] Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

[8] Chen, Z., Zhang, Z., Kan, Z., Cortellazzi, J., Pendlebury, F., Pierazzi, F., Cavallaro, L., Wang, G., 2023. Is it overkill? analyzing feature-space concept drift in malware detectors (2023 ed.).

[9] Fernando, D.W., Komninos, N., 2022. Fesa: Feature selection architecture for ransomware detection under concept drift. Computers & Security 116, 102659.

[10] Fernando, D.W., Komninos, N., 2024. Fesad ransomware detection framework with machine learning using adaption to concept drift. Computers & Security 137, 103629.

[11] Gâlmeanu, H., Andonie, R., 2024. Concept drift visualization of svm with shifting window, in: 2024 28th International Conference Information Visualisation (IV), IEEE. pp. 1–7.

[12] Guerra-Manzanares, A., Luckner, M., Bahsi, H., 2022. Android malware concept drift using system calls: detection, characterization and challenges. Expert Systems with Applications 206, 117200.

[13] Han, D., Wang, Z., Zhong, Y., Chen, W., Yang, J., Lu, S., Shi, X., Yin, X., 2021. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. IEEE Journal on Selected Areas in Communications 39, 2632–2647.

[14] Hinder, F., Vaquet, V., Hammer, B., 2024. Feature-based analyses of concept drift. Neurocomputing 600, 127968.

[15] Liao, N., Wang, J., Guan, J., Fan, H., 2024. A multi-step attack identification and correlation method based on multi-information fusion. Computers and Electrical Engineering 117, 109249.

[16] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G., 2018. Learning under concept drift: A review. IEEE transactions on knowledge and data engineering 31, 2346–2363.

[17] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., Therón, R., . Ugr'16 dataset. URL: https://nesg.ugr.es/nesg-ugr16/.

[18] Navruzov, E., Kabulov, A., 2022. Detection and analysis types of ddos attack, in: 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE. pp. 1–7.

[19] Nunes, Y.T.P., Guedes, L.A., 2024. Concept drift detection based on typicality and eccentricity. IEEE Access 12, 13795–13808.

[20] Rahman, M.K., 2015. Nedindex: A new metric for community structure in networks, in: 2015 18th International Conference on Computer and Information Technology (ICCIT), IEEE. pp. 76–81.

[21] Rossetti, G., Cazabet, R., 2018. Community discovery in dynamic networks: a survey. ACM computing surveys (CSUR) 51, 1–37.

[22] Sgandurra, D., Lupu, E., 2016. Evolution of attacks, threat models, and solutions for virtualized systems. ACM Computing Surveys (CSUR) 48, 1–38.

[23] Shyaa, M.A., Ibrahim, N.F., Zainol, Z., Abdullah, R., Anbar, M., Alzubaidi, L., 2024. Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. Engineering Applications of Artificial Intelligence 137, 109143.

[24] van Timmeren, J.E., Leijenaar, R.T., van Elmpt, W., Wang, J., Zhang, Z., Dekker, A., Lambin, P., 2016. Test–retest data for radiomics feature stability analysis: generalizable or study-specific? Tomography 2, 361.

[25] Wang, P., Yu, H., Jin, N., Davies, D., Woo, W.L., 2024. Quadcdd: A quadruple-based approach for understanding concept drift in data streams. Expert Systems with Applications 238, 122114.

[26] Webb, G.I., Hyde, R., Cao, H., Nguyen, H.L., Petitjean, F., 2016. Characterizing concept drift. Data Mining and Knowledge Discovery 30, 964–994.

[27] Xu, K., Li, Y., Deng, R., Chen, K., Xu, J., 2019. Droidevolver: Self-evolving android malware detection system, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE. pp. 47–62.

[28] Yang, J., Leskovec, J., 2012. Defining and evaluating network communities based on ground-truth, in: Proceedings of the ACM SIGKDD workshop on mining data semantics, pp. 1–8.

[29] Yu, E., Lu, J., Zhang, B., Zhang, G., 2024. Online boosting adaptive learning under concept drift for multistream classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 16522–16530.

9. Annexes

9.1. Methodology

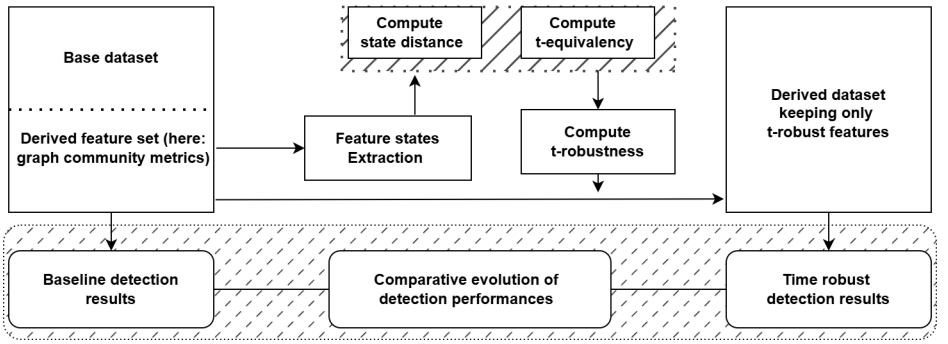
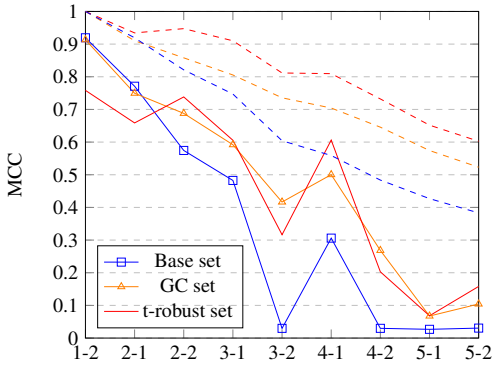


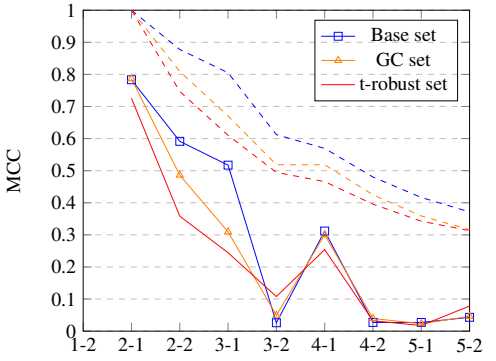
Fig. 2. Methodology and interaction between the models and the data

9.2. Detection performance in concept drift scenarios

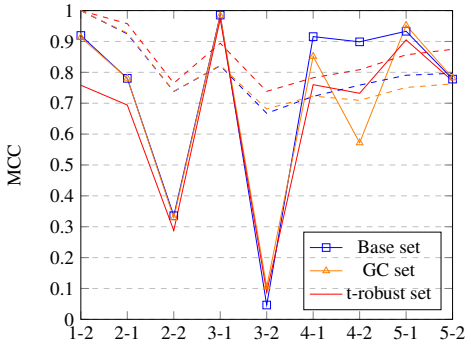
Fig. 3. MCC evolution over the different scenario depending on the XGboost model used. The abscissis denotes the capture period, written as N-M according to scenarios in 5 with N the capture week and $M \in 1, 2$ the week period. Dashed lines are retained expectancies



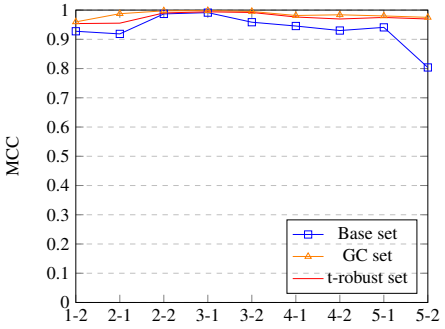
a. Learning scenario 1: 2 first days labeled



b. Learning scenario 2: 2 first days and 1-2 are labeled



c. Learning scenario 3: Training on previous period labeled



d. Control learning scenario : classification 5-folds on same period