REFINE: Robust Evaluation Framework for IDS under Concept Drift in Dynamic Environments

Article

Accepted version

G. N. Costa, A. De Paola, S. Drago, P. Ferraro, G. Lo Re

Proceedings of the 18th International Conference on Agents and Artificial Intelligence (ICAART 2026)

DOI: 10.5220/0014447100004052

Publisher: SCITEPRESS

REFINE: Robust Evaluation Framework for IDS under Concept Drift in Dynamic Environments

Gabriele Nicolò Costa[1][a], Alessandra De Paola[1][b], Salvatore Drago[2][c], Pierluca Ferraro[1][d] and Giuseppe Lo Re[1][e]

[1]*Department of Engineering, University of Palermo, Palermo, Italy*

[2]*IMT School for Advanced Studies Lucca, Lucca, Italy*

{*gabriele.costa03, alessandra.depaola, pierluca.ferraro, giuseppe.lore*}*@unipa.it, salvatore.drago@imtlucca.it*

Keywords: Threat Detection, Online Intrusion Detection System, Machine Learning, Concept Drift, Drift Data Generator, Evaluation Framework.

Abstract: Most machine learning-based Intrusion Detection Systems (IDSs) are designed for stationary environments, where data distributions is assumed to remain constant over time. However, modern network environments are dynamic, and this can lead to significant changes in the observed environment since the training phase, causing degradation in IDS performance. Consequently, increasing attention has been given to online learning techniques designed to address such phenomenon, known as concept drift. Designing such adaptive systems is a far from trivial task, due to a multitude of factors, such as experimental biases as well as the lack of real-world labeled datasets with precise drift annotations. Moreover, the evaluation of such systems still lacks a standardized methodology, and critical aspects are often inconsistently addressed, making comparisons between approaches particularly difficult. To address these challenges, this work proposes REFINE, a Robust Evaluation Framework for IDS under concept drift in dynamic environments. REFINE combines a Concept Drift Stream Generator (CDSG), which produces realistic datasets from real network traffic with controlled drift characteristics, and a robust online evaluation pipeline that mitigates experimental biases. Results demonstrate that REFINE enables accurate, unbiased evaluation and comparison of online IDSs, providing critical insights into their adaptation and detection capabilities across various drift scenarios.

1 INTRODUCTION

In recent years, the rapid proliferation of interconnected devices, driven in part by the widespread adoption of IoT technologies in domains such as industrial systems, smart homes, and smart cities (Agate et al., 2025; Khamesi et al., 2020), has led to a significant increase in cyber threats targeting distributed systems. This trend has intensified the focus on cybersecurity, particularly in the domain of network security, making the detection of network intrusions increasingly critical. In this context, Intrusion Detection Systems (IDSs) are a core component of network security architectures. A wide range of methodologies have been used in developing IDSs, with the most common approaches relying on machine learning (ML) and, more broadly, artificial intelligence (AI) algorithms (Saranya et al., 2020; Agate et al., 2024). The integration of ML techniques has significantly advanced the development of automated IDSs, enabling the analysis of network traffic logs to identify deviations from normal behavior and generate timely alerts, thus assisting network administrators with high detection performance (Liao et al., 2013). Traditionally, ML-based IDSs have been designed to operate in stationary environments, where the statistical properties of the data-generating process remain constant over time (Augello et al., 2024). However, this assumption does not hold in dynamic environments, such as modern complex computer networks. Although recent studies have shown that ML-based IDSs perform well under stationary conditions, their effectiveness tends to degrade over time due to concept drift (Agrahari and Singh, 2022; Agate et al., 2022). In other words, the characteristics of both legitimate and malicious network traffic evolve over time. This may oc-

[a] https://orcid.org/0009-0008-3474-8734
[b] https://orcid.org/0000-0002-7340-1847
[c] https://orcid.org/0009-0009-0367-0484
[d] https://orcid.org/0000-0003-1574-1111
[e] https://orcid.org/0000-0002-8217-2230

cur, for example, following the deployment of new web services, changes in user behavior, or the emergence of zero-day and adversarial attacks. As a result, the training data becomes obsolete, leading to a drop in performance. This scenario introduces new challenges for designing IDSs that are robust and capable of handling concept drift in dynamic environments.

Consequently, increasing attention has been devoted to online learning under concept drift (Augello et al., 2025b; Lu et al., 2018; De Paola et al., 2024), which typically involves two main operations: detecting concept drift and employing adaptive machine learning methods to update the model based on new data (Camarda et al., 2025). Nevertheless, the lack of labeled and real-world network datasets with precise temporal annotations of concept drift poses a key challenge to the development and rigorous testing of algorithms designed for drift detection and adaptation. Despite these developments, the evaluation of such systems remains neither robust nor standardized. As a result, outcomes are often affected by experimental bias, making comparisons between different state-of-the-art systems particularly challenging. A central issue concerns how IDS models are initially trained, evaluated, and subsequently retrained during the adaptation phase, as well as the nature of the data used in each step. In the context of online learning under concept drift, the test-then-train methodology has emerged as a prominent evaluation approach. This methodology evaluates system performance in batches following an initial training phase. Each instance is first tested, then used for drift detection and model adaptation. This ensures that the system is never evaluated on data it has already seen during training. It is therefore crucial to constrain certain preprocessing operations on the dataset. Common procedures such as shuffling or applying PCA to the entire dataset introduce *temporal bias* (Pendlebury et al., 2019), as they leak information from future test records into the training process, thereby inflating performance. Another major issue is class imbalance in the training data, which can cause models to overfit to the majority class distribution. This issue has been addressed in the field of malware detection by the authors of (Pendlebury et al., 2019), who highlighted how this phenomenon, referred to as *spatial bias*, significantly impacts the evaluation of system performance. Common strategies to mitigate this issue include building realistic network datasets with more balanced class distributions, or applying training-set balancing techniques. However, in the context of online learning under concept drift, it is not possible to predict how class imbalance will evolve over time, let alone assume that it will remain as observed in the initial training phase. Finally, many studies evaluate the performance of such systems using a single online metric, typically Accuracy or F1 score. This approach is not robust, as these metrics can be influenced by several factors, such as those discussed above, that may either mask or overestimate the effects of concept drift, making it difficult to assess how performance evolves following adaptation.

To address these limitations, we propose REFINE (Robust Evaluation Framework for IDSs under coNcept drift in dynamic Environments), which enables rigorous evaluation of online IDSs across a wide range of concept drift scenarios by replicating realistic network conditions and ensuring reliable, unbiased performance measurement. REFINE consists of two main components: a Concept Drift Stream Generator (CDSG) and a robust online evaluation pipeline. The CDSG creates datasets derived from real network traffic, allowing precise control over concept drift in terms of type, intensity, width, and spatial bias. This ensures that the inherent complexity of the original data is preserved while simulating realistic drift scenarios. The online evaluation pipeline complements the CDSG by directing the entire training and testing process for IDSs in a controlled environment that accounts for spatial bias and prevents temporal bias. Experimental results on multiple datasets, generated with different drift assumptions using CDSG, show how an online IDS can be accurately evaluated with REFINE and effectively compared to other systems under the same conditions, without introducing experimental bias. Furthermore, the impact of specific drift characteristics generated by CDSG can be directly observed in the resulting model performance. This is critical for evaluating adaptation and detection techniques under different drift conditions.

The remainder of this paper is organized as follows. Section 2 provides an overview about online learning under concept drift. Section 3 describes REFINE framework in detail. Section 4 presents the experimental evaluation, in which several state-of-the-art online IDS models are tested under various concept drift conditions using the proposed framework. Finally, Section 5 concludes the paper and outlines directions for future work.

2 ONLINE LEARNING UNDER CONCEPT DRIFT

Concept drift is defined as a change in the underlying data distribution over time, which violates the assumption of stationarity typically made in machine learning (Hinder et al., 2024). This phenomenon is

typically classified according to its temporal characteristics: abrupt (or sudden) drift refers to an immediate change in the data distribution at a specific point in time; gradual drift develops slowly, with old and new concepts coexisting during the transition; incremental drift involves a progressive shift where data is drawn from both old and new distributions with changing probabilities; recurring drift describes the reappearance of previously observed distributions, often due to seasonality.

The design of online learning mechanisms should include two key operations: drift detection and model adaptation. Drift detection algorithms can be categorized based on their underlying approach, such as error-rate-based, data-distribution-based, or statistically-driven methods. Two widely adopted algorithms for concept drift detection are ADWIN (Bifet and Gavalda, 2007) and KSWIN (Raab et al., 2020). These methods compare one-dimensional data distributions, often based on error-rate analysis. On the other hand, adaptation algorithms focus on strategies for updating existing models in response to drift, a process known as *drift adaptation* or *reaction*. Drift adaptation methods are generally grouped into three categories: simple retraining, ensemble retraining, and model adjustment. Learning under concept drift requires balancing between continuous model retraining and lightweight adjustments, such as weight updates in response to detected drift. Ensemble-based adaptation techniques are widely adopted because they allow flexible update strategies by directly modifying the ensemble structure, adding, replacing, or reweighting base learners in response to concept drift. This flexibility enables different adaptation policies to be implemented while preserving the strong performance that ensemble methods, such as Random Forest (Breiman, 2001), have consistently demonstrated on high-dimensional data like network traffic under stationary conditions (Resende and Drummond, 2018; Zhang et al., 2008).

To address the challenges of adapting ensemble models in dynamic environments, the authors of (Gomes et al., 2017) proposed the Adaptive Random Forest (ARF) algorithm. This method is specifically designed to handle evolving data streams in real time by constructing an ensemble of base classifiers equipped with drift warning and detection mechanisms. When a warning is triggered, a new learner is instantiated. If the drift is confirmed, the corresponding learner in the ensemble is replaced, and a partial fit is used to update the model's weights. This mechanism corresponds to the Adaptive policy, where learners are replaced only when drift is explicitly detected. In contrast, the Incremental (or Periodic Retraining) policy updates the ensemble at regular intervals, regardless of whether drift is observed. While simpler to implement, this approach is typically more computationally demanding and less responsive to changes, making it less efficient in dynamic environments. The key difference between these two adaptation policies lies in how they handle past knowledge. Incremental approaches update model parameters continuously, without modifying the ensemble structure, which often leads to forgetting earlier concepts. In contrast, adaptive methods selectively replace individual learners only upon drift detection, allowing the model to retain relevant knowledge from previous concepts.

Nevertheless, developing robust detection and adaptation algorithms requires access to datasets that include precise temporal annotations and allow control over the characteristics of concept drift. As a result, researchers often rely on synthetic drift generators that provide control over key drift properties, such as type, duration, width, and intensity (Montiel et al., 2018). Several synthetic drift generators have been proposed to simulate different types of concept drift. For instance, *AgrawalGenerator* (Agrawal et al., 1991) simulates drift by altering the generator function; *RandomRBFGenerator* employs random centroids and radial basis functions; *LEDGenerator* (Breiman et al., 2017) induces drift by applying a 10% feature perturbation. Although useful for simulating controlled drift scenarios, these generators produce overly simplistic data that fail to capture the complexity of real-world network traffic. As a result, they often lead to inflated performance estimates and fail to provide realistic evaluation conditions for drift-aware algorithms. To simulate more realistic drift patterns, the authors of (Lin et al., 2024) proposed a generator based on real-world time series, where drift is introduced through operations such as clipping, transposing, and inserting temporal gaps. However, this generator only allows control over the temporal position of the drift and lacks the ability to manipulate other critical aspects such as drift type and intensity, which are essential for evaluating system performance and the effectiveness of incremental adaptation mechanisms.

3 PROPOSED FRAMEWORK

REFINE is a modular framework designed to address the challenges of evaluating machine learning-based IDSs under concept drift, by combining realistic stream generation from real network traffic with configurable drift characteristics and a robust, repro-

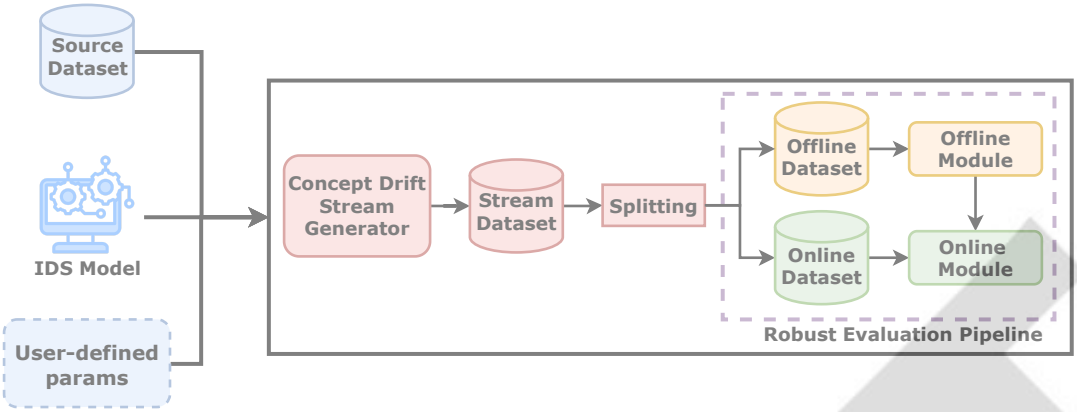


Figure 1: REFINE framework.

ducible evaluation pipeline. The framework operates on three inputs: a source dataset, an ML-based IDS to be evaluated, and a set of user-defined parameters that control the experimental setup. The overall architecture of the framework is shown in Figure 1.

REFINE is organized into two main subsystems:

1. *Concept Drift Stream Generator* (CDSG), which generates a dataset with concept drift and spatial bias, based on user-defined parameters, while avoiding experimental biases related to time. The code of CDSG is publicly available on GitHub[1].
2. *Robust Evaluation Pipeline*, consisting of an *Offline Module* for training the IDS on static data, and an *Online Module* for evaluating its performance on streaming data using fixed-size windows. The online phase also integrates an error-rate-based concept drift detector to monitor performance degradation and assess whether the CDSG introduces drift with the desired characteristics.

The dataset generated by the CDSG is split into two parts: an offline part, used for initial training, and an online part, used for streaming evaluation during the online phase. User-defined parameters play a central role in guiding dataset generation and shaping the evaluation process across both offline and online phases. The following subsections describe the architecture of REFINE, outlining the design and functionality of each subsystem.

3.1 Concept Drift Stream Generator

Unlike traditional approaches that rely on feature perturbation or sample interpolation, CDSG constructs concept drift directly from the structure of the original dataset, and streams the resulting data to simulate online scenarios with fine-grained and realistic control. As shown in Figure 2, the CDSG consists of two main components: the *Concept Generator*, which constructs the concepts from the original dataset by identifying clusters of similar concepts, and the *Drift Streamer*, which builds the final data stream with parameterizable drift characteristics. Parameters for concept generation include the number of clusters and the strategy for building concepts from them. Drift-related parameters include the type, intensity, and temporal annotations (start, end, duration, etc.), as well as the spatial bias (defined as the ratio between the number of benign and malicious samples), which can be specified separately for both the main and drift concepts. Although the clustering parameters are directly related to the traffic classes, they can be specified differently to produce various behaviors. The type, intensity and temporal annotations of concept drift jointly determine when it occurs, how it manifests over time (e.g. sudden or recurrent) and its severity. Higher intensity values induce more pronounced shifts in distribution. The intensity parameter quantifies the severity of drift in terms of deviation from the original concept clusters. Hence, higher values emphasize more anomalous samples and cause them to be selected earlier by the drift streamer. In binary classification, spatial bias affects the distribution of benign and malicious samples in the generated output stream. Higher spatial bias values correspond to increasingly imbalanced datasets, which are characterized by a significant difference between classes. This causes the model to overfit the majority class.

Tab. 1 summarizes the main drift control parameters of CDSG compared with other SOTA drift generators and streamers (introduced in Sec. 2).

[1] https://github.com/ndslab-group/REFINE-CDSG.

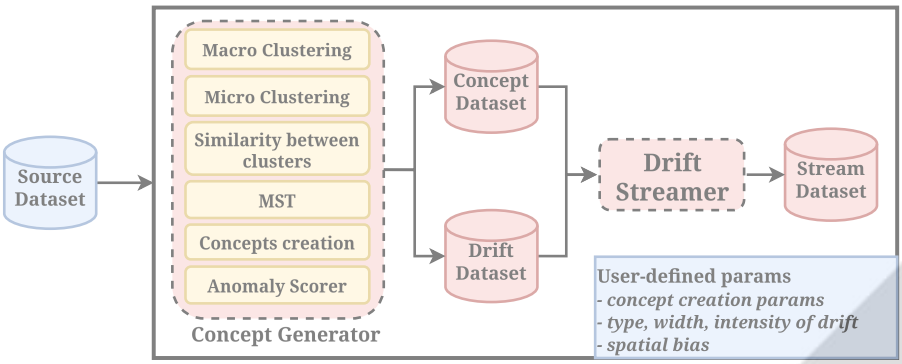


Figure 2: Concept Drift Stream Generator (CDSG) components.

Table 1: Comparison of REFINE-CDSG and SOTA drift generators in terms of domain applicability and drift-control parameters.

Generator and Streamer	Domain Applicability	Drift-Control Parameters
Agrawal	*Synthetic*	Generator functions
RandomRBF	*Synthetic*	Random centroids and radial basis functions
LEDGenerator	*Synthetic*	Feature perturbation
RealDrift	*Realistic*	Clipping, transposing and inserting temporal gaps
REFINE-CDSG	*Realistic*	Drift type and intensity, temporal annotations, spatial bias

3.1.1 Concept Generator

This component splits the original dataset into two distinct subsets: one representing the main concept, and the other the drift concept. The concept generation process consists of multiple phases, which produce the datasets later used by the Drift Streamer. An overview of these phases is shown in Figure 2.

The source dataset is first partitioned into macro clusters, such as benign vs. malicious traffic in binary classification scenarios. Each macro cluster is then subdivided into micro clusters, which may capture, for example, different attack types within the malicious class or distinct user behavior within the benign class. These two phases can be performed using standard clustering algorithms (Xu and Wunsch, 2005), or manually by allowing users to define their own partitions.

The next step is the construction of the main concept, which involves selecting and merging the most similar micro clusters to ensure conceptual consistency. To estimate the similarity between micro clusters, a separate classifier is trained on each macro cluster, treating each micro cluster as a distinct class. The classification scores are then used to quantify how easily the model confuses one micro cluster with another. This methodology relies on a measure called *Cluster Similarity*, whose goal is to quantify the overlap between micro clusters within each macro cluster. The similarity score balances the probability of misclassification (i.e., confusion between clusters) against the probability of correct classification:

$$sim(i,j) = \frac{p(cluster_{pred} = j | cluster_{true} = i)}{p(cluster_{pred} = i)}. \quad (1)$$

This score expresses how easily the classifier confuses cluster i with cluster j, normalized by how well it correctly identifies cluster i. The numerator captures the probability that a sample from cluster i is misclassified as belonging to cluster j, while the denominator reflects the probability of correctly classifying a sample from cluster i. This ratio provides a normalized measure of confusion between clusters, taking into account both classification accuracy and cluster size.

Next, a graph is built for each macro cluster, where nodes represent the micro clusters and edge weights correspond to the similarity scores defined above. From each graph, a Minimum Spanning Tree (MST) is extracted to capture the closest relationships among micro clusters. The *main concept* is then constructed by selecting a subset of connected micro clusters from each MST. The remaining micro clusters are grouped into the *drift concept* and assigned an anomaly score relative to the main concept. This score guides the Drift Streamer in selecting appropriate samples for the drift phase.

3.1.2 Drift Streamer

The Drift Streamer then builds the final data stream by combining these two datasets based on user-defined drift parameters. It controls the type and width of the drift, while also maintaining a specified spatial bias, that is, the proportion of benign to malicious samples, within both concepts. Drift samples are ranked by anomaly score, so that the most anomalous ones

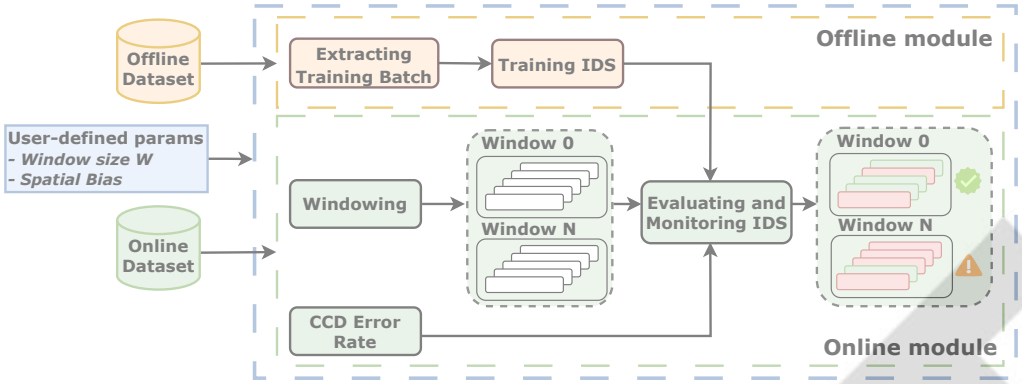


Figure 3: Robust Evaluation Pipeline: Offline and Online modules.

appear first. This ranking allows the streamer to control the intensity of the drift by selecting samples that deviate more sharply from the main concept, while respecting the specified drift width and type. The final stream is generated by combining the main and drift concepts according to the user-defined drift schedule. Samples are initially drawn from the main concept until the drift onset (or a recurrence point, in the case of recurring drift), after which sampling switches to the drift concept for the specified duration. The resulting stream reflects the user-defined concept drift configuration and preserves experimental validity by avoiding operations that could introduce bias.

The effectiveness of this approach depends on the characteristics of the original dataset. In particular, if the dataset contains few samples from the minority class, it may be impossible to generate streams with the desired spatial bias and drift intensity. Moreover, drift duration and intensity are inherently related: longer drift durations typically result in smoother transitions with less pronounced anomalies, while shorter durations produce sharper and more abrupt changes.

3.2 Robust Evaluation Pipeline

This module is designed to address two main goals: training the IDS model on offline data while controlling spatial bias, and evaluating its performance in an online setting using appropriate metrics that capture robustness over time, as illustrated in Figure 3. The pipeline operates on two inputs: the offline and online datasets. It also requires a user-defined window size W, which is used to extract training batches from the offline data and to segment the online stream during evaluation. Larger window sizes generally result in more stable performance trends over time; however, they also tend to amplify performance degradation when concept drift occurs, as the window progressively accumulates a higher proportion of drifted samples. The overall goal is to ensure reliable evaluation of IDS models across both offline training and real-time online monitoring phases.

3.2.1 Offline Module

To mitigate the effects of spatial bias, the model is not trained on the entire offline dataset. Instead, a training window of size W is extracted from it. The extracted window preserves the same spatial bias as the streamed dataset, ensuring consistency between training and evaluation phases and improving the reliability of performance measurements. To ensure that the choice of training window does not affect performance, the model is cross-evaluated on multiple subsets of the offline dataset, each extracted using different windows of size W. The model's performance on each windowed subset is then compared to that obtained by training on the entire offline dataset, using metrics such as Accuracy, Macro F1, F1-score, True Positive Rate (TPR), and True Negative Rate (TNR). Once the windowing strategy has been validated, the final model is trained on a separate window of size W, sampled from the offline dataset according to the user-defined spatial bias. The offline module then returns the trained model, which is passed to the online module for evaluation.

3.2.2 Online Module

The online module simulates the deployment of the trained IDS in a real-time setting, using the online portion of the stream generated by the CDSG. To support step-wise evaluation over time, the online dataset is partitioned into windows of size W, matching the window size used in the offline module. To monitor model performance over time, the framework uses Accuracy, F1-score, and the cumulative Area Under Time (AUT) of F1, a metric that summarizes the sys-

tem's predictive quality across the evaluation period. Originally proposed in (Pendlebury et al., 2019), AUT captures the evolution of model performance, making it useful for assessing robustness and identifying when retraining is required in adaptive scenarios, particularly in the presence of concept drift.

To identify when concept drift occurs during the evaluation, the pipeline integrates a Concept Drift Detector (CDD) module. This module provides temporal indicators of concept drift, validating that the dataset generated by the CDSG contains the expected changes in distribution. The experimental evaluation employs a concept drift detector based on the KSWIN statistical test (introduced in Sec. 1); however, the framework is compatible with any alternative drift detection method.

4 EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of different IDS models, selected to represent distinct adaptation strategies, using the proposed REFINE framework. The goal is to assess how each model performs under controlled concept drift scenarios, using a shared source dataset and consistent configuration for both drift generation and evaluation. The Concept Drift Stream Generator (CDSG) produces streamed datasets that vary in spatial bias, while using the same concept generation configuration across experiments. Drift intensity is automatically adjusted to the maximum feasible level based on the input data. Two different window sizes are used during the streaming phase to assess how the granularity of observation impacts model performance over time. The results show that IDS performance degrades over time in the presence of concept drift, and that specific drift parameters significantly influence this degradation. These findings underline the importance of a robust evaluation framework that allows testing a single IDS model under varying drift scenarios and user-controlled configurations, in order to better understand its behavior in dynamic environments.

Several IDS models with different adaptation strategies are evaluated. Specifically, experimental evaluation includes two static models (Random Forest and XGBoost (Chen et al., 2015)), which are trained once on offline data and applied without any adaptation during the online phase, as well as three adaptive versions. Among the adaptive approaches, two Adaptive Random Forests (ARFs) are configured with distinct adaptation policies: ARF-Adaptive (ARF-A), which updates its ensemble only upon drift detection, and ARF-Periodic Retrain (ARF-P), which retrains at every window regardless of drift detection. In addition, an Adaptive XGBoost (Montiel et al., 2020) (AXGB) configured with a push-based update strategy is also evaluated.

The remainder of this section is organized as follows: Section 4.1 describes the initial experimental setup, including dataset generation via the CDSG and offline training of the IDS models. Section 4.2 presents the results obtained in the online evaluation phase.

4.1 Experimental Setup

Experiments were conducted using CIC-IDS2017 (Panigrahi and Borah, 2018) and BCCC-cPacket-Cloud-DDoS-2024 (Shafi et al., 2024) as source datasets. These datasets provide a comprehensive view of modern network activity, capturing both benign and malicious behaviors. Although these datasets do not exhibit real concept drift, they include various user behaviors in benign traffic and a wide range of attack types in malicious traffic, represented through flow-based features. A preprocessing step is applied to remove NaN (Not-a-Number) and Inf (Infinity) values, ensuring clean input for concept generation (Sharafaldin et al., 2018). In the experiments, the CDSG module is used to generate streamed datasets with fixed concept generation parameters. The drift intensity is automatically adjusted to be as high as possible, in order to maximize the observable effect of concept changes. Table 2 and Table 3 summarize key experimental parameters for both source datasets, including the level of spatial bias, the type of drift (sudden and recurrent sudden), the drift width (start, end, and recurrence points), and the offline/online splitting ratio. The source datasets are first split into two macro clusters representing benign and malicious traffic, respectively. These clusters are then divided into three and six micro-clusters, respectively, using the Hierarchical Clustering algorithm in a completely unsupervised manner that does not rely purely on class type labels. This step ensures that the data is grouped into micro-clusters that are distinct from each other and not according to the original labels. To construct the main and drift concepts, the process selects one micro cluster from the benign macro cluster to represent the main concept, and two micro clusters from the malicious macro cluster to represent the drift concept. These concept datasets are then passed to the Drift Streamer, which generates data streams with configurable spatial bias and drift characteristics based on the user-defined parameters.

The resulting datasets, along with the selected

Table 2: Overview of the generated datasets with concept drift, using CIC-IDS2017 as source dataset and exhibiting Recurrent Sudden Drift. For each level of spatial bias (K), the table reports the total size, the offline/online split, the expected number of windows for two different window sizes ($|W_1| = 5000$, $|W_2| = 20000$), and the temporal indicators of drift (start and recurrence), both as sample indices and as expected window positions.

		Splitting (40%-60%)		**Expected windows**		**Drift start-recurrent in Online Dataset**		**Expected sudden drift at win**		**Expected recurrent drift at win**	
Bias K	**Dataset Size**	**Offline**	**Online**	W_1	W_2	**Start Drift**	**Recurrent Drift**	W_1	W_2	W_1	W_2
0.7	500000	200000	300000	60	15	100000	260000	20	5	52	13
0.9	700000	280000	420000	84	21	120000	320000	24	6	64	16

Table 3: Overview of the generated datasets with concept drift, using BCCC-cPacket-Cloud-DDoS-2024 as source dataset and exhibiting Sudden Drift. For the selected level of spatial bias (K), the table reports the offline/online dataset size, the expected number of windows for two different window sizes ($|W_1| = 5000$, $|W_2| = 20000$), and the temporal indicators of drift (start), both as sample indices and as expected window positions.

	Dataset Size		**Expected Windows**		**Drift in Online Dataset**	**Expected sudden at win**	
Bias K	**Offline**	**Online**	W_1	W_2	**Start Drift**	W_1	W_2
0.9	40000	160000	32	8	80000	16	4

IDS models, are the input of the Robust Evaluation Pipeline. The process begins with the Offline Module, which selects a training window from each offline dataset. This window is sampled to preserve the same spatial bias used during drift generation, ensuring consistency between training and evaluation. The experiments use two different training window sizes, as reported in Table 2 and Table 3. Models trained on these subsets are then evaluated on the remaining portion of the offline dataset and compared against a baseline approach where the entire offline dataset is used for training without any resampling. Offline results showed no statistically significant difference between models trained on the full dataset and those trained on the reduced subsets in both cases. This is explained by the fact that both the adaptive and periodically retrained models rely on the static model as their base classifier, before applying their respective adaptation strategies. Moreover, both approaches achieved comparable performance, with an average F1-score of approximately 0.98 and a standard deviation of ±0.01.

4.2 Online evaluation

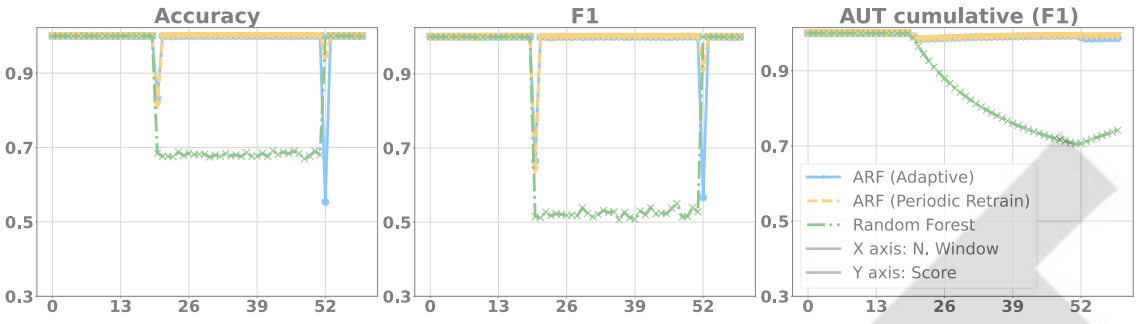
Once trained, the models are evaluated within the Online Module using the same window sizes adopted during the offline phase for the windowing step. Figure 4 shows the online evaluation of all models across four configurations, combining different spatial bias levels and window sizes. Each subplot reports Accuracy, F1-score, and cumulative AUT(F1) over time, considering generated datasets with concept drift, using CIC-IDS2017 as source dataset (Tab. 2) and three representative models.

All models maintain strong performance while the stream reflects the main concept, even when it reappears as part of the recurrent drift pattern. However, their performance drops noticeably at the expected drift points, as detected by the KSWIN Concept Drift Detector (CDD). Table 4 confirms that the KSWIN detector successfully identified drift intervals that align with those configured by the framework, as detailed in Table 2. This validates the ability of the CDSG to generate controlled and reproducible concept drift.

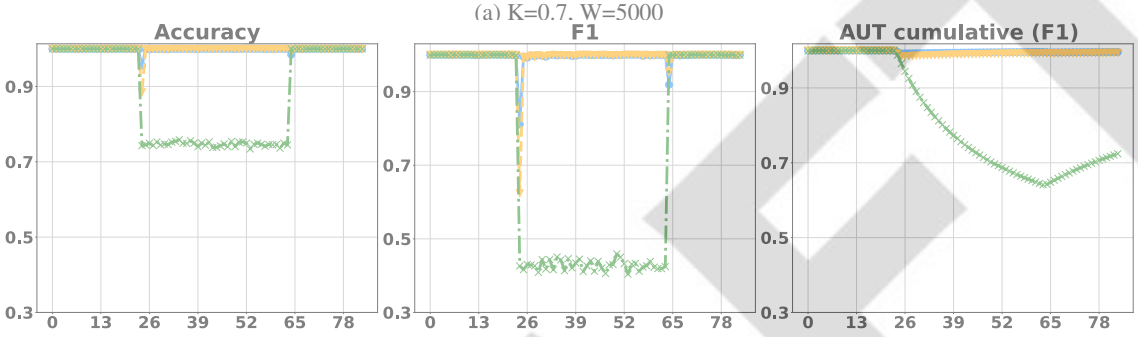
To analyze the impact of individual parameters on model performance across different metrics, the behavior of the *static model* is analyzed. The experimental evaluation show that a high spatial bias results in a more pronounced drop in performance, regardless of the window size configuration. Higher spatial bias (Fig. 4b, 4d) increases the risk of overfitting and amplifies the apparent abruptness of concept drift, effectively increasing its intensity. In contrast, lower bias (Fig. 4a, 4c) leads to smoother transitions. These results highlight the importance of frameworks like REFINE for assessing the impact of spatial bias in dynamic settings.

The results also show that relying only on accuracy is not sufficient, as it can be misleading in the presence of class imbalance: models may appear to perform well by overfitting to the majority class. Metrics like F1-score and cumulative AUT(F1) provide a more reliable assessment by capturing both model effectiveness and how performance evolves over time in response to concept drift. Spatial bias impacts and plays a crucial role in evaluating the performance of adaptive models over time; however, thanks to their adaptation mechanisms, these models are able to recover quickly after drift events. Window size also plays a critical role during drift events: smaller windows tend to experience less performance degradation, while larger ones include more samples from the new concept, amplifying the effects of drift.

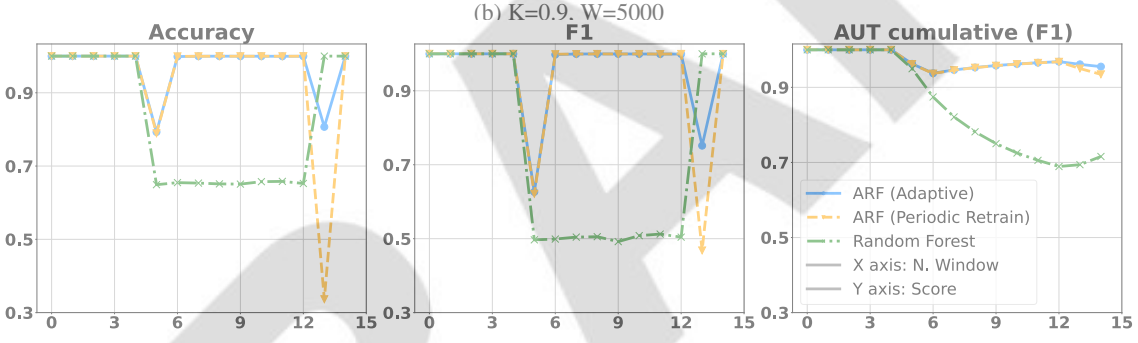
As a result, evaluations with narrower windows



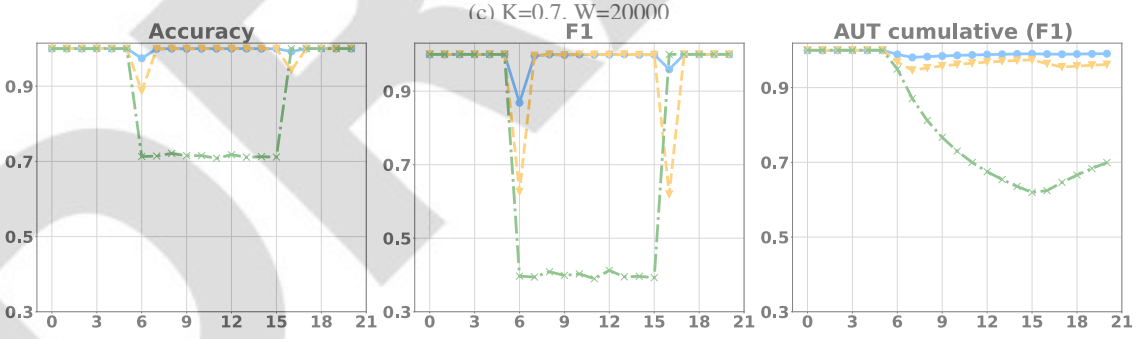
(a) K=0.7, W=5000



(b) K=0.9, W=5000



(c) K=0.7, W=20000



(d) K=0.9, W=20000

Figure 4: Performance of the evaluated IDS models over time across different concept drift scenarios based on generated datasets with concept drift, using CIC-IDS2017 as source dataset (Tab. 2). Each subplot reports Accuracy, F1-score, and cumulative AUT(F1) as a function of the window number. The X-axis shows the evaluation window index, and the Y-axis shows the corresponding metric values for each configuration.

(Fig. 4a, 4b) exhibit smaller drops in AUT compared to those with wider windows (Fig. 4c, 4d). Finally, comparing the AUT curves reveals that frequently retrained adaptive models can become less reliable over time. Continuous weight updates may lead the model to forget the original concept, reducing its long-term effectiveness. This effect is evident in Fig. 4d, where the AUT of the continuously adaptive ARF surpasses that of the periodically retrained variant. These observations show that the quality of adaptation depends on accurate drift detection, class balance, and representative samples. Moreover, continuous adaptation often performs worse than adaptation triggered after drift detection, particularly in recurrent drift scenarios. As expected, hyperparameter variations consistently affect performance and temporal behavior.

Table 4: First and last windows where the KSWIN detector identifies concept drift in datasets with different spatial bias levels (Tab. 2).

	Spatial Bias K=0.7		**Spatial Bias K=0.9**	
Window Size	**5000**	**20000**	**5000**	**20000**
First Drift Detected	20-21	5-6	24-25	5-6
Last Drift Detected	52-53	13-14	64-65	16-17

In addition to the experiments described earlier, which involved only the static Random Forest model and its adaptive variants (applied to datasets generated from CIC-IDS2017), we extended the evaluation to include XGBoost and its adaptive version, AXGBoost. These additional experiments were conducted on synthetic datasets with concept drift generated from two different sources: CIC-IDS2017 (Table 2) and BCCC-cPacket-Cloud-DDoS-2024 (Table 3).

Table 5 reports the average Accuracy and F1-score of each model across three phases: before the drift, during the drift, and after the drift in the case of recurrent drift.

As shown, both static models suffer a significant drop in performance during drift. In particular, with the BCCC-cPacket dataset, the F1-score degradation is more pronounced. This is due to the presence of different types of benign traffic compared to CIC-IDS2017, which enables CDSG to induce stronger drift in both benign and malicious traffic distributions. In contrast, the adaptive versions of the models demonstrate the expected behavior of performance recovery following retraining phases.

These results confirm that CDSG is capable of generating datasets with diverse types of drift from different source datasets, and that the proposed framework can be applied across different types of detection systems.

Table 5: Mean Accuracy and F1-scores for pre-drift (Before), during-drift (During), and recurrent-drift (Recurrent) phases, across both source datasets, with parameters K=0.9 and W=20,000.

		Before		***During***		***Recurrent***		***Total***	
	Model	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
CIC-IDS2017	RF	100.0	100.0	71.40	39.89	100.0	100.0	86.38	71.37
	ARF-A	100.0	99.98	99.72	98.58	99.84	99.18	99.83	99.13
	ARF-P	100.0	100.0	98.85	96.16	98.86	92.38	99.18	96.36
	XGB	100.0	100.0	79.61	48.20	100.0	100.0	90.29	75.33
	AXGB	98.33	83.31	93.71	86.54	98.56	91.66	96.18	86.83
BCCC-cPacket	RF	99.31	96.41	89.99	0.0	-	-	94.65	48.2
	ARF-A	99.01	94.75	97.43	74.97	-	-	98.22	84.86
	ARF-P	99.07	95.07	97.43	74.99	-	-	98.25	85.03
	XGB	99.36	96.68	89.98	0.0	-	-	94.67	48.34
	AXGB	96.41	84.72	97.64	92.1	-	-	97.02	88.41

5 CONCLUSION AND FUTURE WORK

This paper proposed REFINE, a Robust Evaluation Framework for Intrusion Detection Systems (IDS) under concept drift in dynamic environments. REFINE enables the generation of realistic data streams with controlled concept drift, derived from real network traffic, without relying on synthetic perturbations or sample interpolation. The Concept Drift Stream Generator (CDSG) constructs distinct concepts from the structure of the original dataset, while the evaluation pipeline supports systematic offline training and online testing under user-defined drift scenarios.

Experimental results show that REFINE effectively simulates realistic drift conditions while avoiding common experimental biases. The framework allows a detailed analysis of how drift-related parameters (e.g., window size, drift width, and class imbalance) affect model robustness over time. These results highlight the importance of a reproducible and parameterized evaluation strategy in non-stationary environments. REFINE thus provides a valuable tool for assessing the performance of IDSs, enabling researchers to explore adaptation capabilities under a wide range of drift configurations.

The experimental evaluation includes variations in two hyperparameters (spatial bias and window size), both of which are closely tied to practical application requirements. In domains such as threat detec-

tion, real-world data often exhibit imbalanced distributions of malicious activity, which can be effectively simulated through spatial bias. Similarly, the choice of window size plays a crucial role in balancing responsiveness to concept drift against overall system performance, a trade-off that the framework explicitly captures.

Future work will explore additional combinations of hyperparameters, with particular focus on their impact on development and implementation in real-world settings. In this regard, the proposed framework provides a valuable foundation for testing and simulating scenarios that reflect realistic operating conditions.

Moreover, future work will also focus on extending REFINE with additional configurable parameters and evaluation strategies to support a broader range of drift scenarios. We plan to integrate and compare multiple adaptive learning techniques and drift detection methods within the same framework, along with drift-aware algorithms and sensitivity analyses of clustering techniques used in concept formation. This will enable a systematic evaluation of their relative effectiveness prior to deployment, enhancing the empirical robustness of the overall approach and improving the scalability of REFINE.

Moreover, although the experiments focused on two benchmark datasets for IDSs, we aim to explore the applicability of REFINE in other domains affected by concept drift, such as, distributed reputation systems (Agate et al., 2021), social networks security (Concone et al., 2019), malware detection (Augello et al., 2025a), increasing the general applicability of the framework.

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.

REFERENCES

Agate, V., Batool, F., Bordonaro, A., De Paola, A., Ferraro, P., Lo Re, G., Morana, M., and Virga, A. (2025). Population Protocols for Adaptive Event Dissemination with Autonomous Agents in Vehicular Networks. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 640–647.

Agate, V., De Paola, A., Drago, S., Ferraro, P., and Lo Re, G. (2024). Enhancing IoT Network Security with Concept Drift-Aware Unsupervised Threat Detection. In *2024 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.

Agate, V., De Paola, A., Lo Re, G., and Morana, M. (2021). A simulation software for the evaluation of vulnerabilities in reputation management systems. *ACM Transactions on Computer Systems*, 37(1-4).

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

Agrahari, S. and Singh, A. K. (2022). Concept drift detection in data stream mining : A literature review. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part B):9523–9540.

Agrawal, R., Imienski, T., and Swamy, A. (1991). Database mining: A performance perspective, ieee tran. *On Knowledge and Data Engg*.

Augello, A., De Paola, A., and Lo Re, G. (2025a). Hybrid multilevel detection of mobile devices malware under concept drift. *Journal of Network and Systems Management*, 33(2).

Augello, A., De Paola, A., and Lo Re, G. (2025b). M2fd: Mobile malware federated detection under concept drift. *Computers & Security*, page 104361.

Augello, A., Lo Re, G., Peri, D., and Thiyagalingam, P. (2024). Nep-ids: a network intrusion detection system based on entropy prediction error.

Bifet, A. and Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Camarda, F., De Paola, A., Drago, S., Ferraro, P., and Lo Re, G. (2025). Managing concept drift in online intrusion detection systems with active learning. In *CEUR Workshop Proceedings*, volume 3962. CEUR-WS.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Concone, F., Re, G. L., Morana, M., and Ruocco, C. (2019). Twitter spam account detection by effective labeling. volume 2315.

De Paola, A., Drago, S., Ferraro, P., and Lo Re, G. (2024). Detecting zero-day attacks under concept drift: An online unsupervised threat detection system. In *CEUR Workshop Proceedings, 8th Italian Conference on Cybersecurity, ITASEC 2024*.

Gomes, H. M., Bifet, A., Read, J., Barddal, J.-P., Enembreck, F., Pfharinger, B., and Holmes, G. (2017).

Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9):1469–1495.

Hinder, F., Vaquet, V., and Hammer, B. (2024). 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*, 7:1330257.

Khamesi, A. R., Silvestri, S., Baker, D., and De Paola, A. (2020). Perceived-value-driven optimization of energy consumption in smart homes. *ACM Transactions on Internet of Things*, 1(2).

Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., and Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24.

Lin, B., Huang, C., Zhu, X., and Jin, N. (2024). Realdrift-generator: A novel approach to generate concept drift in real world scenario. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1124–1129. IEEE.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363.

Montiel, J., Mitchell, R., Frank, E., Pfahringer, B., Abdessalem, T., and Bifet, A. (2020). Adaptive xgboost for evolving data streams. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Montiel, J., Read, J., Bifet, A., and Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5.

Panigrahi, R. and Borah, S. (2018). A detailed analysis of cicids2017 dataset for designing intrusion detection systems. *International Journal of Engineering & Technology*, 7:479–482.

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., and Cavallaro, L. (2019). {TESSERACT}: Eliminating experimental bias in malware classification across space and time. In *28th USENIX security symposium (USENIX Security 19)*, pages 729–746.

Raab, C., Heusinger, M., and Schleif, F.-M. (2020). Reactive soft prototype computing for concept drift streams. *Neurocomputing*, 416:340–351.

Resende, P. A. A. and Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Comput. Surv.*, 51(3).

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

Shafi, M., Lashkari, A. H., Rodriguez, V., and Nevo, R. (2024). Toward generating a new cloud-based distributed denial of service (ddos) dataset and cloud intrusion traffic characterization. *Information*, 15(4):195.

Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A., et al. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1(2018):108–116.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

Zhang, J., Zulkernine, M., and Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5):649–659.