Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

ADAPT-IDS: An Unsupervised Multi-Metric Framework forReliable Concept Drift Detection and Adaptive Intrusion Detectionin Evolving Network Streams

ROHIT THAKUR

thakur.rohit1997@gmail.com

Panjab Unviersity

Raj Kumari

Panjab Unviersity

Article

Keywords: Concept drift, intrusion detection, unsupervised learning, multi-metric fusion, adaptive IDS, selective labeling, network traffic streams

Posted Date: September 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10963104/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

ADAPT-IDS: An Unsupervised Multi-Metric Framework for Reliable Concept Drift Detection and Adaptive Intrusion Detection in Evolving Network Streams

Rohit Kumar Thakur[a], Raj Kumari[a]

[a]University Institute of Engineering and Technology Panjab University, Chandigarh, 160014, Chandigarh, India
[b]University Institute of Engineering and Technology Panjab University, Chandigarh, 160014, Chandigarh, India

ARTICLE INFO

Keywords:
Concept drift
intrusion detection
unsupervised learning
multi-metric fusion
adaptive IDS
selective labeling
network traffic streams

ABSTRACT

Intrusion detection systems (IDSs) operating in evolving network environments are vulnerable to concept drift, where changes in traffic and attack behavior can reduce detection performance over time. This paper proposes ADAPT-IDS, an unsupervised multi-metric framework that couples drift detection with label-efficient IDS adaptation. The framework integrates Jensen–Shannon divergence, standardized distributional difference, statistical drift significance, and residual exceedance analysis through weighted evidence fusion. Unlike approaches that rely on a single statistical measure or immediate model updates, ADAPT-IDS imposes minimum-effect constraints and persistence-based confirmation to distinguish sustained distributional changes from transient fluctuations. Confirmed drift triggers a limited oracle-labeling procedure and high-confidence benign sample selection to construct a balanced replay buffer for IDS retraining, reducing the need for continuous annotation. Experiments are conducted on a chronologically ordered network intrusion dataset containing 2,097,150 instances and 63 numerical features. After using 200,000 instances for initial model development, 1,897,150 instances are processed prequentially as a streaming sequence. ADAPT-IDS identifies 327 candidate drift events, of which 73 satisfy the persistence criterion and trigger model adaptation. The resulting IDS achieves 97.46% accuracy, 99.94% precision, 85.88% detection rate, 92.38% F1-score, 92.93% balanced accuracy, and 0.912 MCC, with a false-alarm rate of 0.0119%. Oracle labeling is required for 7.70% of evaluation-stream instances. The results support the use of persistent multi-metric drift evidence for label-efficient adaptation of IDSs under evolving network conditions.

1. Introduction

Modern networked systems generate traffic that changes continuously as new applications, communication services, cloud platforms, and connected devices are introduced. The same environment also provides attackers with new opportunities to modify their techniques and exploit changes in network behavior. As a result, Intrusion Detection Systems (IDSs) must operate on data whose characteristics may differ from those observed during model training. Machine-learning-based IDSs can provide effective traffic classification, but their performance is affected when the distribution of incoming data changes over time. This phenomenon, commonly referred to as **concept drift**, can gradually increase false alarms and missed attacks if the detection model is not able to respond to the changing data distribution [1], [2].

Concept drift refers to a change in the underlying data-generating process or in the relationship between observations and their associated target concepts over time [1]. In network traffic, such changes may result from variations in user activity, network configuration, application usage, communication protocols, traffic intensity, or attack behavior. A classifier trained using historical observations may therefore encounter future traffic with statistical properties that were not adequately represented during training. The effect can be particularly problematic in streaming environments, where observations arrive continuously and the assumption that a previously trained model will remain valid cannot be maintained indefinitely [1], [3].

A range of methods has been proposed for learning from evolving data streams. Adaptive Windowing (ADWIN), for example, monitors incoming observations and adjusts the effective data window when sufficient evidence of a change is detected [3]. Many established drift detectors, however, use prediction errors or require class labels to determine

*Corresponding author
ORCID(s):

whether the underlying concept has changed. This requirement is restrictive in applications where labels are delayed, costly to obtain, or unavailable for a large proportion of incoming traffic. In network security, expert verification of individual traffic flows is especially difficult to sustain at stream scale. This has motivated the development of **unsupervised concept drift detection**, in which changes are identified from the observed data distributions without depending on immediate ground-truth labels [4].

Statistical comparison between data windows offers a direct way of monitoring such changes. The Kolmogorov–Smirnov statistic, Jensen–Shannon divergence, and Wasserstein distance, for example, can be used to measure differences between reference and current distributions. These measures do not respond to distributional changes in exactly the same manner. A particular statistic may be more sensitive to changes in location, spread, distributional shape, or the degree of overlap between two distributions. Network traffic is also inherently multidimensional, so changes may occur in several features and may not be adequately represented by a single measure. Recent studies and surveys of unsupervised drift detection have reported a wide range of detection assumptions and statistical mechanisms, indicating that the choice and combination of monitoring measures remain important for unlabeled data streams [4], [5].

Detection of a statistical difference alone does not necessarily mean that an IDS should be retrained. Network traffic can change substantially over short periods because of temporary bursts, changes in activity levels, or other routine variations. An isolated statistical deviation may therefore disappear in subsequent observations without representing a persistent change in the traffic-generating process. Updating the IDS whenever such a deviation occurs can lead to unnecessary retraining and may introduce instability into the detection model. At the other extreme, delaying adaptation when a genuine and persistent change has occurred can leave the IDS operating with a decision boundary that no longer reflects current traffic. A practical adaptive system consequently needs a mechanism that considers the persistence of detected changes before initiating model updates.

The need for such mechanisms has also been reflected in recent IDS research. A comprehensive survey of drift-aware intrusion detection approaches reports increasing interest in monitoring changes in cyber environments and adapting detection models as network and threat characteristics evolve. The survey also points to concept drift and changing feature behavior as areas that remain insufficiently addressed in many IDS studies [6]. These findings suggest that adaptive IDS evaluation should not be limited to classification accuracy on a fixed test set. The monitoring process, the confirmation of a change, the timing of adaptation, and the subsequent behavior of the updated model all need to be considered.

The availability of labels presents a second practical difficulty. Supervised IDS models require labeled examples when they are retrained, but obtaining such labels continuously from a high-volume network stream is expensive and may require expert analysis. A fully supervised adaptation scheme can therefore place a substantial burden on annotation. A completely label-free update scheme has the opposite problem: without reliable labels, the system may incorporate incorrectly classified or unrepresentative samples into the training process. This creates a middle ground in which drift monitoring remains independent of class labels, while a limited amount of labeling is requested only after there is sufficient evidence that the model requires adaptation. The cost and availability of ground-truth information are also identified in the literature as important reasons for investigating unsupervised drift detection methods [4].

Based on these considerations, this paper proposes **ADAPT-IDS: An Unsupervised Multi-Metric Framework for Reliable Concept Drift Detection and Adaptive Intrusion Detection in Evolving Network Streams**. The framework treats drift monitoring and intrusion classification as related but separate processes. The drift-monitoring component does not use class labels and evaluates incoming traffic using four indicators: **Jensen–Shannon divergence (JS)**, which measures distributional similarity; **standardized distributional difference (ED)**, which captures changes in statistical characteristics; **statistical drift significance (DS)**, which evaluates feature-level distributional changes; and **residual exceedance rate (RS)**, which measures deviations from the learned benign traffic profile. Their outputs are combined using a weighted evidence-fusion mechanism to obtain a single drift score. The purpose of combining these measures is to obtain evidence from different aspects of distributional change rather than making the drift decision dependent on one statistic.

The framework also distinguishes between detecting a possible change and confirming that the change is persistent. A candidate drift event is generated when the fused evidence satisfies the configured detection condition. The recent sequence of candidate events is then examined using a persistence criterion. Model adaptation is performed only after this criterion is satisfied. This additional step limits the effect of isolated statistical deviations and reduces the likelihood that short-lived traffic fluctuations will repeatedly initiate IDS retraining.

After a persistent drift event has been confirmed, ADAPT-IDS uses a selective-label adaptation procedure. A limited set of samples is submitted for oracle labeling, while a benign-profile reconstruction model is used to identify observations with high confidence of belonging to the benign class. These samples are combined to form a balanced replay buffer, which is then used to update the supervised IDS classifier. The approach therefore does not require labels throughout the entire stream. Instead, labeling is associated with confirmed drift events, when additional information is needed to support model adaptation. This provides a compromise between continuous supervised retraining and unrestricted label-free model updates.

The proposed framework is evaluated on a chronologically ordered network intrusion dataset containing **2,097,150 traffic instances and 63 numerical features**. The first 200,000 observations are used for initial model development, while the remaining 1,897,150 observations form the evaluation stream. The stream is divided into 380 temporal batches and processed sequentially. For each batch, predictions are generated before the corresponding adaptation step, following a prediction-before-adaptation protocol. This prevents information from a future batch from being used to produce its preceding predictions and provides a more appropriate assessment of model behavior under changing traffic conditions. The evaluation considers drift detection, persistence-confirmed drift events, adaptive IDS performance, oracle-labeling requirements, and feature sensitivity.

The main contributions of this work are as follows:

1. **Unsupervised multi-metric drift detection:** A label-independent monitoring mechanism combines JS divergence, standardized distributional difference, statistical drift significance, and residual exceedance information to characterize changes in evolving network traffic.

2. **Persistence-aware drift confirmation:** A temporal confirmation mechanism is introduced to separate isolated candidate drift events from changes that persist across successive observations before initiating model adaptation.

3. **Selective-label adaptive IDS:** The framework combines limited oracle labeling with high-confidence benign sample selection and balanced replay-based retraining, thereby restricting the use of manual labels to confirmed adaptation events.

4. **Chronological prequential evaluation:** The proposed approach is evaluated on a large temporal network traffic stream using a prediction-before-adaptation protocol, allowing IDS performance to be assessed without incorporating information from future batches into preceding predictions.

5. **Empirical analysis of adaptation behavior:** The evaluation examines drift detections, confirmed adaptation events, classification performance, labeling requirements, and feature sensitivity to characterize the behavior of ADAPT-IDS under evolving network conditions.

ADAPT-IDS consequently combines unsupervised monitoring with supervised model adaptation without requiring continuous class labeling. Distributional evidence is first used to identify and confirm persistent changes in the incoming stream. Only then is limited labeling introduced to support the construction of an updated training buffer. The resulting design addresses two practical requirements of streaming IDSs: detecting changes without relying on continuously available labels and limiting model updates to situations where persistent evidence of change has been observed.

2. Related Work

2.1. Concept Drift Detection in Evolving Data Streams

Concept drift has been studied extensively in the context of machine-learning systems that operate on changing data streams. Gama *et al.* provided an early systematic treatment of concept-drift adaptation, covering different forms of drift and the strategies used to maintain learning performance as the data-generating process changes [1]. Lu *et al.* later reviewed the broader concept-drift literature and discussed detection, adaptation, and learning under non-stationary conditions [2]. These studies establish a basic premise for stream learning: a model trained on historical data cannot be assumed to remain suitable when the characteristics of future observations change.

Several classical methods detect drift through changes in prediction error. Drift Detection Method (DDM) monitors the error rate of a learning algorithm and signals a change when the observed error departs sufficiently from its previous behavior [6]. Early Drift Detection Method (EDDM) extends this principle by considering the distribution of

distances between classification errors, with greater emphasis on gradual changes [5]. Other approaches have focused on changes in the input stream itself. Adaptive Windowing (ADWIN), for example, dynamically adjusts the size of the monitored window and detects changes by comparing statistics calculated from different portions of the window [4]. These methods have provided important foundations for online learning, but error-based approaches can depend on timely class labels, which is restrictive when labels are delayed or unavailable.

Statistical testing provides an alternative for monitoring changes without relying directly on classifier error. Nishida and Yamauchi investigated statistical testing for concept-drift detection in data streams [12], while subsequent work developed non-parametric procedures based on Hoeffding's bounds for online drift detection [14]. Pesaranghader and Viktor proposed a fast Hoeffding-based drift detector for evolving streams, further demonstrating the use of statistical evidence for online change detection [13]. Distribution-based approaches can also compare reference and current windows using measures such as the Kolmogorov–Smirnov statistic, Jensen–Shannon divergence, and Wasserstein distance. The KS test provides a non-parametric comparison of empirical distributions [54], while divergence-based measures quantify differences between probability distributions [50], and Wasserstein distance provides a distributional distance based on optimal transport [52], [53].

The choice of statistical measure affects the type of change that can be detected effectively. A detector based on a single statistic may respond strongly to one form of distributional change while providing weaker evidence for another. This issue becomes more relevant in high-dimensional streams, where different features can change at different rates and may exhibit changes in location, dispersion, shape, or tail behavior. Recent surveys have consequently examined a broader range of monitoring approaches and their assumptions. Hinder *et al.* reviewed methods for detecting concept drift in evolving environments and discussed the different statistical and monitoring principles used by existing detectors [7]. Their complementary survey addressed methods for locating and explaining drift after a change has been detected [8].

Unsupervised drift detection has received particular attention because many real-world streams do not provide labels at the time observations arrive. Sethi and Kantardzic examined the reliable detection of concept drift from streaming unlabeled data and discussed the difficulties associated with distinguishing actual changes from normal variation [11]. More recently, Lukats *et al.* conducted a benchmark and survey of fully unsupervised concept-drift detectors using real-world data streams [9]. Their analysis illustrates that detector performance can vary considerably across streams and evaluation settings. Arora *et al.* also reviewed machine-learning-based approaches for concept-drift detection and adaptation, covering different detection and adaptation strategies used in streaming environments [10].

These studies indicate that unsupervised monitoring removes an important dependence on immediate ground-truth information, but it does not by itself solve the problem of deciding whether an observed deviation represents a change that requires model adaptation. This distinction is particularly relevant to IDSs, where short-lived changes in traffic may be common. ADAPT-IDS therefore uses several statistical indicators and applies a persistence criterion before initiating model adaptation.

2.2. Concept Drift in Intrusion Detection Systems

Network intrusion detection presents a particularly demanding setting for concept-drift research. Network traffic is affected by changing applications, users, configurations, workloads, and communication patterns. At the same time, attack behavior can change as adversaries adopt new tools and techniques. An IDS trained on a historical traffic distribution may therefore encounter future observations that differ from those used during model development. If the model is not updated appropriately, this distributional change can affect its ability to distinguish benign and malicious traffic.

Recent studies have examined this problem directly. Khanna investigated concept-drift-based intrusion detection for evolving data-stream classification and discussed the limitations of static IDS models when traffic characteristics change over time [21]. Jemili *et al.* combined concept-drift detection with online incremental learning to address changing network traffic in intrusion detection environments [22]. These studies demonstrate the value of connecting drift detection with model adaptation rather than treating intrusion detection as a fixed classification problem.

Other work has approached the problem through continual or self-supervised learning. Yang *et al.* proposed a self-supervised adaptation method for handling concept drift in network intrusion detection [23]. Hu *et al.* investigated dual adaptive windows for online network intrusion detection under concept drift [24]. Ying *et al.* presented METANOIA as a lifelong intrusion detection and investigation system designed to mitigate concept drift [25]. Semi-supervised and continual-learning approaches have also been investigated for network intrusion detection [26], [27]. More recent

studies have examined continual learning under evolving network threats and incremental learning for IoT intrusion detection [28], [29].

These approaches differ in how they identify change and how they update the IDS. Some modify the training window, while others use incremental or continual learning to retain information as new observations become available. Their results reinforce the need for adaptation in evolving environments, but the adaptation mechanism itself remains dependent on how reliably the system can identify a meaningful change.

A related line of work has focused specifically on drift-aware IDS architectures. García-Huitzitl *et al.* investigated an adaptive IDS combining clustering and random-forest classification with concept-drift detection [32]. Meganathan *et al.* studied adaptive sliding windows for real-time intrusion detection under concept drift [31]. The HOIDS framework proposed by De Paola *et al.* further considers concept drift within a hybrid online intrusion detection architecture [63]. These studies show that drift detection can be integrated into an operational IDS rather than being used only as an independent data-stream analysis tool.

Despite this progress, the requirements of drift detection and those of IDS adaptation are not identical. Detecting a statistical difference between two traffic windows does not necessarily mean that the classifier should be retrained immediately. An adaptive IDS must also consider the persistence of the observed change and the reliability of the data used for subsequent model updating. These considerations form the basis for the design adopted in ADAPT-IDS.

2.3. Unsupervised Drift Detection for Cybersecurity Streams

The availability of ground-truth labels is one of the practical constraints in network-stream monitoring. A high-volume traffic stream can contain a large number of flows, while determining the correct class of each flow may require additional analysis or expert verification. Continuous labeling is consequently difficult to maintain as the stream grows.

The broader concept-drift literature has addressed this limitation through unsupervised monitoring. Sethi and Kantardzic examined drift detection from unlabeled streams and emphasized the difficulty of identifying reliable changes without access to target information [11]. Lukats *et al.* specifically evaluated fully unsupervised drift detectors on real-world data streams and showed that their effectiveness depends on the characteristics of the underlying stream and the evaluation setting [9]. Hinder *et al.* also provide a detailed examination of drift-monitoring methods and their assumptions in evolving environments [7].

The same issue appears in cybersecurity applications. A detector that requires an immediate classification outcome for every observation cannot be used directly when security labels are delayed. This has encouraged the use of feature-distribution monitoring as an independent source of drift evidence. The approach is attractive because the incoming traffic itself can be examined before its final class is known.

Network traffic, however, is not statistically uniform. Different features may respond differently to changes in network activity. For example, changes in traffic volume may alter numerical distributions, whereas a change in application or attack behavior may affect only a subset of features or alter the shape of their distributions. A detector based on one statistic may therefore provide only partial evidence of the underlying change.

ADAPT-IDS addresses this issue by monitoring four complementary signals: Jensen–Shannon divergence, standardized distributional difference, statistical drift significance, and residual exceedance rate. The individual measures are not treated as independent drift decisions. Their outputs are combined to form a unified evidence score, which is subsequently subjected to threshold and persistence conditions.

2.4. Adaptive Learning and Drift-Triggered IDS Updating

Identifying a change is only one part of the adaptation problem. Once a persistent change has been detected, the system must determine which observations should be retained and how they should be incorporated into the IDS model. An update based on unreliable observations can be as problematic as failing to update the model.

Adaptive stream-learning methods provide several approaches to this problem. ADWIN modifies the effective observation window in response to detected changes, allowing older observations to be removed when they become less representative of the current stream [4]. Adaptive Random Forests extend this idea to ensemble learning by allowing classifiers to adapt as the underlying stream evolves [15]. Ensemble methods for data-stream classification have also been studied as a way of maintaining predictive performance under changing distributions [16]. Other approaches have investigated one-pass change detection and online classifier adaptation for streaming data [17], [18], [19].

Within IDS research, online and continual adaptation has become increasingly common. Jemili *et al.* combined drift detection with online incremental learning for intrusion detection [22], while Yang *et al.* investigated self-supervised adaptation under concept drift [23]. Hu *et al.* used adaptive windows for online network intrusion detection [24],

and METANOIA considers lifelong adaptation in the presence of changing attack conditions [25]. Continual-learning approaches have also been applied to address evolving intrusion patterns [26]–[29].

A common concern in adaptive learning is the quality of the observations used during an update. If every sample collected after a drift alarm is treated as suitable training data, incorrectly classified or malicious observations can enter the updated training set. This may alter the decision boundary in an undesirable direction. The problem is particularly relevant when the adaptation process is only partially supervised.

ADAPT-IDS uses a different sequence. A drift alarm does not immediately cause retraining. The candidate event must first satisfy a persistence condition. Once persistent drift is confirmed, a limited set of observations is subjected to oracle labeling, while high-confidence benign observations are identified using the benign-profile reconstruction mechanism. These samples are then used to construct a balanced replay buffer for IDS retraining. The design limits both unnecessary updates and the amount of manual labeling required by the adaptation process.

2.5. Multi-Metric Drift Detection

Different statistical measures provide different views of distributional change. The Kolmogorov–Smirnov statistic compares empirical cumulative distributions and is widely used for non-parametric distributional testing [54]. Jensen–Shannon divergence provides a symmetric and bounded divergence derived from the Kullback–Leibler divergence [50], [51]. Wasserstein distance instead measures the cost of transforming one probability distribution into another and has been extensively studied in statistical applications [52], [53]. These differences make the measures complementary rather than interchangeable.

The broader literature also shows that drift detection is sensitive to the characteristics of the data stream. The benchmark conducted by Lukats *et al.* found differences among fully unsupervised detectors across real-world streams [9]. The surveys by Hinder *et al.* similarly demonstrate that drift detectors rely on different assumptions and provide different forms of evidence about changing data distributions [7], [8]. These findings provide a basis for considering multiple statistical signals when monitoring complex streams.

In cybersecurity, the argument for multiple sources of evidence is strengthened by the heterogeneous nature of network features. A change may appear as a shift in a feature distribution, a change in its dispersion, a statistically significant difference between windows, or an increase in observations that deviate from a learned benign profile. No single signal necessarily captures all of these changes.

Recent IDS research has consequently moved toward architectures that explicitly account for evolving traffic and model adaptation [21]–[34], [63]. However, the specific combination of multi-metric statistical monitoring, persistence-based confirmation, and selective labeling considered in ADAPT-IDS addresses the adaptation process as a connected sequence rather than as independent detection and learning components.

In ADAPT-IDS, the four monitoring signals are combined through weighted evidence fusion. The resulting score is evaluated against the drift threshold, after which candidate events are accumulated for persistence analysis. Model adaptation is therefore based on both the magnitude of the observed evidence and its temporal consistency.

2.6. Research Gap

The literature indicates that concept-drift detection, unsupervised monitoring, and adaptive intrusion detection have each received substantial attention. The remaining problem is how these components can be combined without introducing unnecessary labeling or excessive model updates.

First, label dependence remains an important limitation in conventional drift detection. Error-based methods such as DDM and related approaches require information about classifier performance, which in practice depends on the availability of labels [5], [6]. Unsupervised detectors address this restriction, but their behavior can vary across data streams [9], and their integration with subsequent IDS adaptation is not always explicit.

Second, a single statistical measure may provide incomplete evidence of drift. KS-based testing, divergence measures, and transport-based distances quantify different properties of distributional change [50], [52], [54]. This makes the use of several complementary indicators reasonable when the monitored traffic is heterogeneous and high-dimensional.

Third, detection and adaptation should not necessarily occur at the same time. A statistical deviation can result from temporary traffic variation rather than a persistent change. Existing adaptive-stream and IDS studies demonstrate several ways of responding to detected change [15], [22], [24], but the timing of adaptation remains an important design consideration.

Fourth, the data selected for adaptation can affect the reliability of the updated IDS. Continual and adaptive learning methods provide mechanisms for incorporating new observations [26]–[29], but using incoming traffic without sufficient control can expose the training process to mislabeled or unrepresentative observations.

Finally, the amount of labeling required for adaptation is an important operational consideration. Fully supervised updating can become expensive when the traffic stream is large, whereas completely label-free updating may provide insufficient information for safely modifying a supervised classifier. An adaptation mechanism that requests labels only after persistent evidence of drift offers an intermediate alternative.

These observations motivate the design of ADAPT-IDS around four connected stages:

$$\text{multi-metric monitoring} \rightarrow \text{evidence fusion} \rightarrow \text{persistence confirmation} \rightarrow \text{selective adaptation}.$$

The framework does not introduce a new statistical test in isolation. Instead, it combines existing forms of distributional evidence with a persistence rule and a controlled adaptation procedure. The resulting architecture is intended for network streams in which drift monitoring can be performed without continuous labels, while a limited amount of supervised information remains available when adaptation is required.

2.7. Positioning of ADAPT-IDS

ADAPT-IDS is positioned at the intersection of unsupervised concept-drift detection and adaptive intrusion detection. Existing studies provide the individual building blocks: statistical drift tests support distributional monitoring [12], [54]; unsupervised detectors remove the immediate requirement for class labels [9], [11]; adaptive and continual-learning methods provide mechanisms for updating models under changing conditions [15], [22], [26]; and recent IDS studies explicitly address changing traffic and attack behavior [21]–[34], [63].

The distinction of the proposed framework lies in how these components are arranged. ADAPT-IDS does not use class labels during routine drift monitoring. Four complementary indicators are first combined into a drift score. Candidate events are then subjected to a persistence condition before an adaptation decision is made. When persistent drift is confirmed, only a limited subset of observations is labeled, while high-confidence benign observations are used to maintain class balance in the replay buffer. This sequence limits the role of manual labeling to confirmed adaptation events.

Table 1
Qualitative Comparison of Drift Detection and Adaptive IDS Strategies

Research Direction	Label-free Drift Detection	Multi-metric Evidence	Persistent Confirmation	Selective Labeling	Adaptive IDS
Traditional supervised drift detectors [5], [6]	×	Usually ×	Variable	×	✓
ADWIN-type approaches [4]	✓	×	Window-based	×	✓
Statistical distribution-based detectors [12], [14], [54]	✓	Usually single metric	Usually ×	×	Variable
Fully unsupervised drift detectors [9], [11]	✓	Variable	Variable	×	Usually separate
Drift-aware IDS approaches [21], [22], [31], [63]	Variable	Variable	Variable	Variable	✓
Continual / adaptive IDS approaches [23]–[29]	Variable	Variable	Variable	Variable	✓
ADAPT-IDS	✓	✓	✓	✓	✓

Abbreviations: ADWIN—Adaptive Windowing; DDM—Drift Detection Method; EDDM—Early Drift Detection Method; IDS—Intrusion Detection System; JS—Jensen–Shannon divergence; ED—standardized distributional difference; DS—statistical drift significance; RS—residual exceedance rate.

Table 2
Comparison of Selected Concept-Drift and IDS Adaptation Approaches

Study / Framework	Streaming IDS	Concept Drift	Unsupervised Monitoring	Multi-Metric Evidence	Adaptive Learning
ADWIN [4]	✓	✓	✓	×	✓
DDM / EDDM [5], [6]	✓	✓	×	×	✓
Statistical drift detection [12], [14], [54]	✓	✓	✓	Usually single metric	Optional
Fully unsupervised detectors [9], [11]	✓	✓	✓	Variable	Usually separate
Concept-drift-aware IDS [21], [22], [24]	✓	✓	Variable	Variable	✓
Continual / self-supervised IDS [23], [25]–[29]	✓	✓	Variable	Variable	✓
HOIDS [63]	✓	✓	✓ / Architecture-dependent	Hybrid	✓
ADAPT-IDS	✓	✓	✓	**JS + ED + DS + RS**	✓

Table 3
Adaptation Characteristics of Selected Approaches

Study / Framework	Retraining / Updating	Historical Information	Persistence Mechanism	Reference Updating	Label-Efficient Design
ADWIN [4]	✓	Window-based	Window comparison	✓	×
DDM / EDDM [5], [6]	✓	Limited historical statistics	Warning / drift levels	Method-dependent	×
Statistical drift detectors [12], [14], [54]	Optional	Reference/current windows	Usually ×	Variable	×
Fully unsupervised detectors [9], [11]	Usually separate	Method-dependent	Variable	Variable	✓
Adaptive / continual IDS [22]–[29]	✓	Method-dependent	Variable	Variable	Variable
HOIDS [63]	✓	Architecture-dependent	Architecture-dependent	✓	Architecture-dependent
ADAPT-IDS	✓	**Balanced replay buffer**	**Persistence confirmation**	✓	**Selective oracle labeling**

3. Background on Concept Drift Management

Concept drift management is the process of detecting, assessing, and responding to changes in the statistical properties of an evolving data stream. In conventional machine-learning settings, training and test data are often assumed to follow a relatively stable distribution. This assumption is difficult to maintain in streaming environments, where changes may occur in both the distribution of input variables and the relationship between observations and their class labels. As a result, a model that performs well during its initial deployment can gradually lose effectiveness when the underlying data-generating process changes [1–3].

3.1. Concept Drift in Data Streams

Let a data stream be represented as a sequence of observations

$$S = \left\{ \left(x_t, y_t \right) \right\}_{t=1}^{T},$$

where x_t denotes the feature vector observed at time t, and y_t represents its corresponding class label. Under a stationary learning environment, the joint probability distribution $P(X, Y)$ remains approximately unchanged over time. Concept drift occurs when this distribution changes:

$$P_t(X, Y) \neq P_{t+\Delta}(X, Y).$$

The change may affect the marginal feature distribution $P(X)$, the conditional distribution $P(Y \mid X)$, or both [1, 3]. In network intrusion detection, such changes can arise from modifications in normal user behavior, newly deployed applications, changing communication patterns, variations in traffic intensity, emerging attack techniques, and changes in attack frequency.

Concept drift should not be treated as equivalent to random variation. A temporary fluctuation may disappear after a short period without changing the underlying operating environment, whereas genuine drift can persist and alter the conditions under which a model makes predictions. Distinguishing persistent changes from short-lived fluctuations is therefore important for reliable adaptation [1, 7].

3.2. Types of Concept Drift

Concept drift can be characterized according to the temporal manner in which the underlying distribution changes. Common categories include **abrupt, gradual, incremental, and recurring drift** [1, 2].

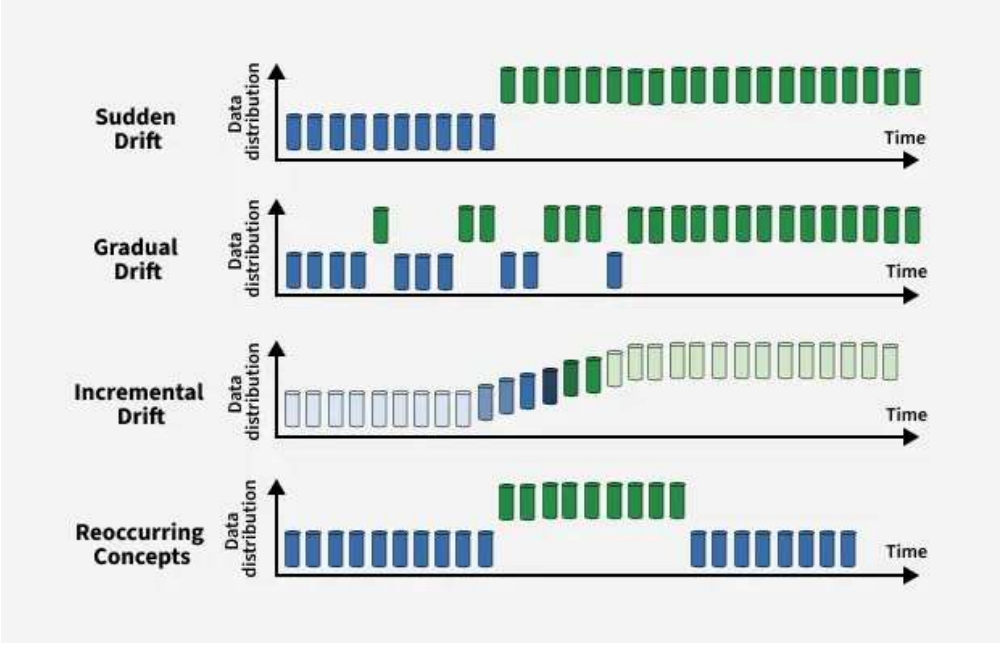
Abrupt drift occurs when the data-generating process changes within a relatively short period. In a network environment, the sudden deployment of a new attack mechanism may produce an immediate change in traffic characteristics. Such changes can be temporally localized, but delayed adaptation may rapidly reduce detection performance.

Gradual drift occurs when the previous and emerging concepts coexist during a transition period. The probability of observing the new concept progressively increases while the previous concept remains present. This situation may arise when a new application, protocol, or attack technique is gradually adopted across a network.

Incremental drift describes a continuous transition between concepts through a sequence of intermediate states. Rather than moving directly from one distribution to another, the stream evolves through several closely related distributions. Individual changes may therefore be small and difficult to distinguish from ordinary traffic variability.

Recurring drift occurs when a previously observed concept reappears after a period of absence. Seasonal traffic patterns, recurring workloads, or periodically repeated attack campaigns can produce this behavior. An adaptive system that completely discards historical information may consequently need to relearn a concept that has already been encountered.

These drift patterns have different implications for detection and adaptation. A strategy suitable for an abrupt change may not be appropriate for gradual or recurring changes. Practical drift-management systems should therefore respond to sustained environmental changes while limiting unnecessary adaptation caused by short-term fluctuations [1, 4].



center

center

Figure 1: Types of concept drift in evolving data streams.

3.3. Drift Detection

Drift detection constitutes the first operational stage of concept drift management. Its purpose is to determine whether the statistical characteristics of a recent observation window differ sufficiently from those of a reference window to justify further investigation or adaptation.

Let X_R denote the reference window and X_C denote the current observation window. A generic drift detector evaluates a discrepancy function

$$D\left(X_R, X_C\right),$$

and generates a candidate drift indication when

$$D\left(X_R, X_C\right) > \tau,$$

where τ denotes a decision threshold.

The definition of D determines the type of change that can be identified. Error-based methods monitor changes in prediction errors and therefore generally require access to class labels or reliable feedback. Distribution-based methods instead compare observations collected from different temporal windows and can be applied when labels are unavailable or delayed [5, 6, 11].

For unsupervised network monitoring, the reference window represents previously observed traffic behavior, whereas the current window contains newly arriving traffic. A substantial discrepancy between the two windows provides evidence that the operating environment may have changed. Statistical significance, however, does not necessarily indicate an operationally important change. With sufficiently large samples, even a small distributional difference can become statistically significant. Practical drift detection should therefore consider both the statistical evidence and the magnitude of the observed change [7, 11].

3.4. Statistical Measures for Distributional Change

Several statistical measures can be used to characterize differences between reference and current distributions. Each measure captures a different aspect of distributional change.

The **Kolmogorov–Smirnov (KS) statistic** measures the maximum absolute difference between two empirical cumulative distribution functions:

$$D_{KS} = \sup_x \left|F_R(x) - F_C(x)\right|.$$

The KS statistic is non-parametric and can identify differences between empirical distributions without assuming a particular parametric form [54]. It is therefore useful for detecting changes in individual feature distributions. Its response, however, depends on the type of distributional change being observed.

The **Jensen–Shannon (JS) divergence** provides a symmetric measure of distributional discrepancy. Given two probability distributions P and Q, it is defined as

$$JS(P, Q) = \frac{1}{2}KL(P \parallel M) + \frac{1}{2}KL(Q \parallel M),$$

where

$$M = \frac{P + Q}{2}.$$

The JS divergence is derived from the Kullback–Leibler divergence and provides a bounded symmetric measure that can be used to compare empirical feature distributions [50, 51].

The **Wasserstein distance** provides another perspective on distributional change by measuring the minimum transportation cost required to transform one probability distribution into another. Unlike measures based only on pointwise discrepancies, Wasserstein distance can reflect changes in the location and overall shape of distributions [52, 53].

These measures are not equivalent. A distributional change that produces a weak response under one metric may be more clearly reflected by another. This complementarity provides a motivation for combining multiple statistical indicators when monitoring heterogeneous and evolving network traffic.

3.5. Drift Confirmation and False Alarm Control

A practical challenge in drift management is controlling false alarms. Network traffic naturally exhibits temporal variability, and individual batches may differ from a reference distribution even when no persistent environmental change has occurred. Triggering model retraining after every isolated deviation can therefore increase computational cost and may lead to unstable adaptation.

One way to address this problem is to separate **drift detection** from **drift confirmation**. The detection stage identifies a candidate change, while the confirmation stage evaluates whether the evidence persists sufficiently to justify an adaptation event. This distinction is consistent with the broader need to distinguish transient deviations from sustained distributional changes in streaming environments [1, 7, 11].

For a sequence of binary candidate decisions c_t, a persistence mechanism can examine the most recent W batches and confirm a drift event when the number of positive decisions reaches a predefined requirement K:

$$\sum_{i=t-W+1}^{t} c_i \geq K.$$

Here, W represents the persistence window and K represents the minimum number of candidate drift indications required for confirmation. The mechanism reduces the probability that an isolated statistical deviation will immediately trigger model adaptation.

In ADAPT-IDS, this distinction is central to the adaptation process. A candidate drift indication does not directly cause model retraining. Instead, adaptation is initiated only after the observed evidence satisfies the persistence criterion.

3.6. Drift Adaptation

Once a persistent change has been confirmed, the learning system must determine how the model should be updated. Common adaptation strategies include retraining with recent observations, incremental model updating, sliding-window learning, and replay-based learning [1, 2, 15, 59, 60].

A **sliding-window strategy** continuously replaces older observations with newer ones. This allows the model to emphasize recent traffic conditions but can also remove historical observations that remain useful for recognizing previously observed behavior.

Incremental learning updates an existing model using newly available observations rather than rebuilding the complete model from scratch. This can reduce computational requirements, although the effectiveness of the approach depends on whether the underlying learning algorithm supports reliable incremental updates [15, 59].

Replay-based adaptation maintains a controlled collection of historical observations and combines them with newly acquired samples during retraining. Such a strategy can preserve previously learned patterns while providing greater control over the composition and class balance of the adaptation data [59, 60].

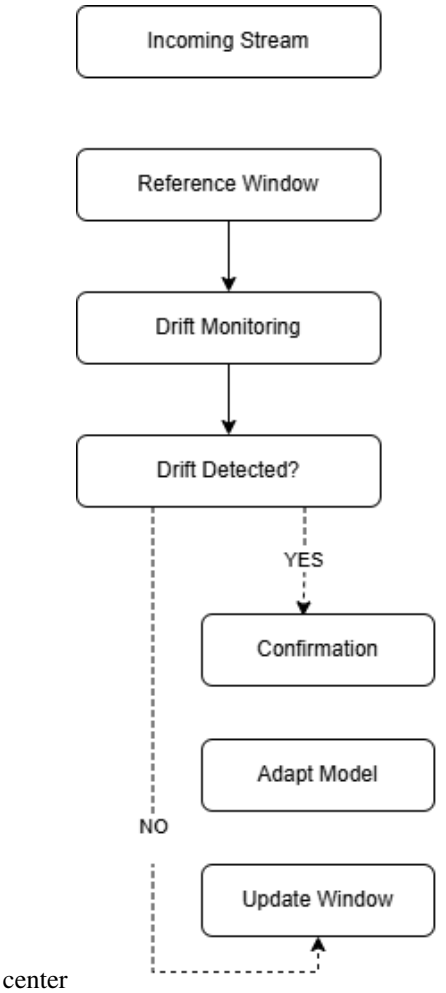
Adaptation in intrusion detection introduces an additional data-quality problem because newly arriving traffic can contain both benign and malicious observations. Incorporating all newly observed samples without any selection mechanism may introduce noisy, uncertain, or incorrectly characterized examples into the retraining set. Data selection is therefore an important component of an adaptive IDS.

3.7. Labeling Requirements in Adaptive Systems

Label availability is an important practical constraint in adaptive intrusion detection. Supervised drift detectors can monitor changes in prediction errors, but this requires timely access to ground-truth labels. In operational network environments, obtaining reliable labels for individual traffic flows can be time-consuming and may require expert analysis.

An alternative is to perform drift monitoring in feature space and request labels only when adaptation is considered necessary. This separates the continuous monitoring process from the more expensive labeling process. The detector can therefore operate without labels during routine monitoring, while a limited number of observations can be labeled after a persistent change has been confirmed [11].

This principle is incorporated into ADAPT-IDS. The drift-monitoring component operates without class labels, while a limited oracle-labeling step is introduced only after confirmed adaptation events. High-confidence benign observations identified from the learned benign traffic profile can supplement the labeled samples. These observations are then organized through a balanced replay mechanism to construct the data used for subsequent IDS retraining.



center

center

Figure 2: Generic drift detection and adaptation workflow.

The objective is not to eliminate labeling completely, but to restrict labeling to adaptation events where additional information is most useful. This provides a practical compromise between fully supervised adaptation and continuous manual annotation.

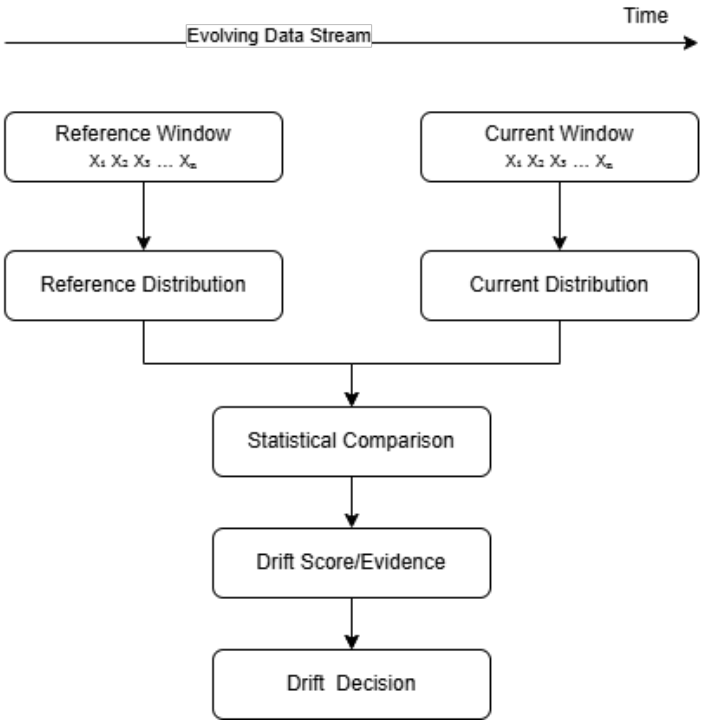
3.8. Reference Window Management

The reference window is another important component of drift management. Since the detector compares current observations against this reference, both the composition and age of the reference influence detection behavior.

If the reference remains unchanged for an extended period, it can become increasingly different from the current operating environment even after the model has successfully adapted. Subsequent observations may then continue to appear anomalous relative to an obsolete baseline, resulting in repeated drift indications. Conversely, updating the reference too aggressively can cause genuine drift to be incorporated into the baseline before it has been sufficiently confirmed, reducing the sensitivity of subsequent detection.

Reference management therefore requires a balance between **historical stability** and **environmental responsiveness**. The reference should provide sufficient continuity for meaningful comparison while remaining representative of the current operating environment.

In ADAPT-IDS, the reference distribution is updated following confirmed adaptation events using recent traffic batches. This allows subsequent monitoring to operate relative to a more contemporary representation of network behavior rather than relying indefinitely on the initial baseline.



center

center

Figure 3: Concept drift management process in ADAPT-IDS.

3.9. Concept Drift Management in ADAPT-IDS

The concept-drift management process adopted in ADAPT-IDS can be viewed as a sequence of connected operations. Incoming traffic is first monitored against a reference distribution using multiple distributional indicators. Candidate changes are then evaluated using a persistence mechanism. Only confirmed persistent changes initiate the adaptation stage, where selected labeled and high-confidence benign observations are used to update the IDS.

The principal design principle is that **detection and adaptation are treated as separate decisions**. Distributional evidence first identifies a potential change. Persistence analysis then determines whether the evidence is sufficiently sustained. Only after this confirmation is the supervised IDS permitted to adapt.

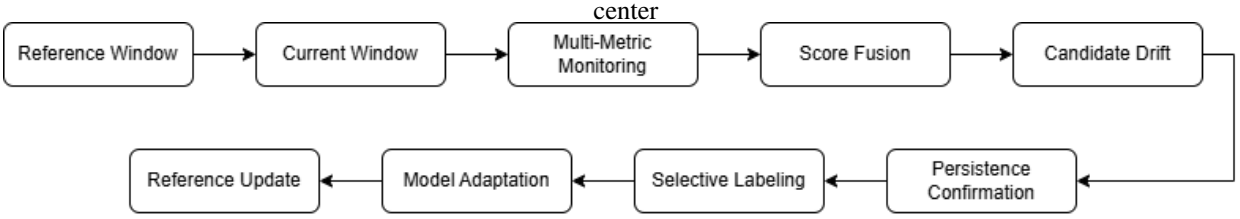
This separation is particularly relevant to evolving network streams because it combines three practical requirements: **label-independent monitoring, controlled model adaptation, and efficient use of labeled data**. The resulting management process provides the conceptual basis for the ADAPT-IDS framework described in the following section.

4. ADAPT-IDS Architecture

4.1. Architectural Overview

The ADAPT-IDS architecture is designed for intrusion detection in network environments where traffic characteristics and attack behavior change over time. Conventional IDS models are commonly developed under the assumption that the relationship between input observations and target classes remains sufficiently stable after training. In an evolving network environment, this assumption may no longer hold. Changes in user behavior, applications, communication patterns, and attack activity can alter the underlying data distribution and gradually reduce the effectiveness of a previously trained detector [1, 2].

center



center

Figure 4: Event-driven drift confirmation and adaptation loop.

ADAPT-IDS separates two operations that are often coupled in adaptive learning: *drift monitoring* and *IDS adaptation*. The drift-monitoring layer operates without using ground-truth labels from incoming traffic. This is important in network environments where labels may be unavailable, delayed, or expensive to obtain [11]. Once sufficient evidence of persistent distributional change has been observed, a limited labeling procedure is activated and the supervised IDS is updated.

The architecture consists of seven principal stages:

1. chronological stream construction and preprocessing;
2. reference-window initialization;
3. multi-metric unsupervised drift monitoring;
4. weighted drift-score fusion and threshold evaluation;
5. persistence-based drift confirmation;
6. selective-label adaptive IDS retraining; and
7. dynamic reference-window rebaselining.

The resulting processing cycle follows the sequence *monitor* → *detect* → *confirm* → *adapt* → *rebaseline*. This separation between detecting a change and adapting the predictive model is consistent with the broader concept-drift management literature [1, 2].

Let the complete network stream be represented by

$$S = \left\{ \left(x_t, y_t \right) \right\}_{t=1}^{N},$$

where $x_t \in \mathbb{R}^d$ denotes the feature vector associated with the t-th network observation and y_t denotes its class label. During routine drift monitoring, y_t is excluded from the detector. The drift-monitoring layer therefore operates only on feature observations X, whereas class labels are introduced only when a confirmed adaptation event requires supervised model updating.

This separation is a central characteristic of ADAPT-IDS. It allows the system to monitor the stream continuously without requiring continuous ground-truth annotation, while retaining the ability to use labeled information when model adaptation becomes necessary.

4.2. Chronological Stream Construction

Network observations are first arranged according to their available timestamps to form a temporally ordered stream. Chronological ordering is essential because concept drift concerns changes in the data distribution over time. Randomly shuffling observations would destroy this temporal relationship and could allow information from later periods to influence earlier model-development stages [1, 2].

In the experimental implementation, the timestamp is used to establish the temporal order but is not supplied to the IDS as a predictive feature. After preprocessing, the dataset contains 63 numerical features.

The dataset is organized into three logical components:

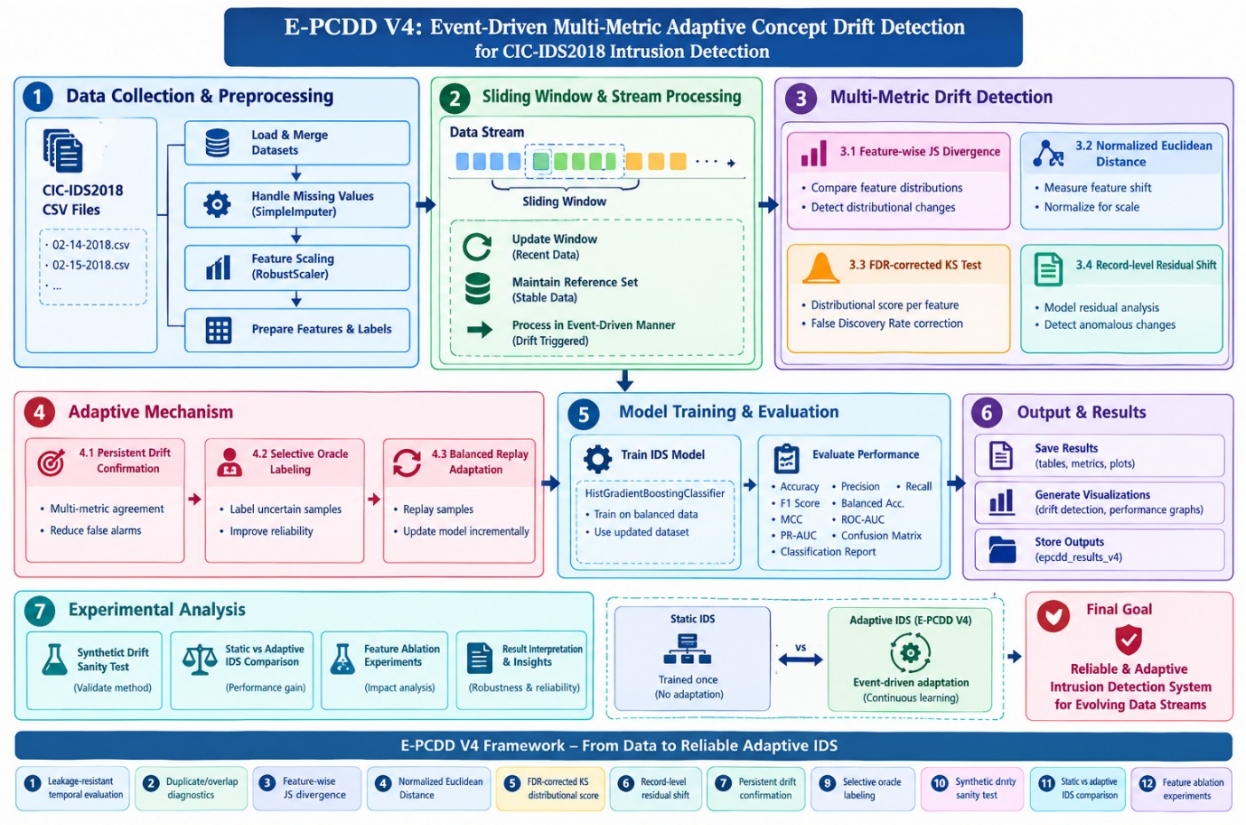


Figure 5: Event-driven multi-metric adaptive concept-drift detection workflow.

$$D = D_{init} \cup R_0 \cup S,$$

where D_{init} denotes the initial model-development data, R_0 denotes the initial reference representation, and S denotes the subsequent streaming evaluation sequence.

The initial development segment is used to construct the first IDS model and the benign behavioral profile. A portion of this period is retained to establish the initial reference distribution. The remaining observations are then processed sequentially as a streaming evaluation sequence.

This organization supports prequential evaluation. Each incoming batch is first processed by the currently deployed IDS, and any adaptation triggered by that batch is performed only afterward. Thus, information from the current batch does not contribute to the model used to generate its own predictions. This ordering is important when evaluating adaptive systems because adaptation before prediction can produce overly optimistic performance estimates [1, 59].

4.3. Data Preprocessing Module

The preprocessing module converts raw network-flow records into a numerical representation suitable for both drift analysis and IDS classification. The procedure includes timestamp handling, numerical feature selection, missing-value treatment, removal of non-informative attributes, and robust scaling.

Missing values are replaced using median values estimated exclusively from the initial training partition. The scaling parameters are also learned only from the initial training data. The resulting transformation can be represented as

$$\widetilde{X} = f\left(X; \theta_{init}\right),$$

where $f(\cdot)$ denotes the preprocessing transformation and θ_{init} represents the parameters estimated from D_{init}.

The same fitted transformation is subsequently applied to the reference data and to all streaming batches. No future batch is used to update the imputation or scaling parameters during evaluation.

This restriction prevents information from future traffic regimes from entering the preprocessing stage before those regimes are actually observed. Such leakage control is particularly important in chronological stream evaluation because the purpose is to measure model performance on traffic that was not available when the corresponding predictions were generated.

4.4. Initial IDS Learning Module

The initial IDS establishes the first classification model using labeled observations from the initial development period. In the present implementation, a HistGradientBoosting classifier is used to learn nonlinear relationships between network-flow features and intrusion classes.

The initial model is represented as

$$M_0 = \mathcal{L}\left(X_{init}, Y_{init}\right),$$

where M_0 denotes the initial IDS and $\mathcal{L}$ denotes the supervised learning procedure.

The trained classifier is subsequently used to predict incoming traffic batches. The classifier is not retrained after every incoming batch. Instead, the adaptation module remains inactive unless the drift management layer identifies sufficient persistent evidence of a change.

This event-driven strategy limits unnecessary model updates and keeps the predictive model unchanged during periods in which the observed traffic remains within the established operating regime.

4.5. Reference-Window Management

The reference window represents the traffic regime against which newly arriving observations are evaluated. Window-based comparison is a common strategy in streaming drift detection because it allows recent observations to be compared with a representative historical distribution without retaining the complete stream [1, 4].

Let the reference window at time t be

$$R_t = \left\{x_{t-r+1}, \ldots, x_t\right\},$$

and let the incoming batch be

$$C_t = \left\{x_{t+1}, \ldots, x_{t+b}\right\},$$

where r denotes the reference-window size and b denotes the current batch size.

In the present implementation,

$$r = 50{,}000, \qquad b = 5{,}000.$$

A fixed historical reference can become increasingly unrepresentative as the operating environment changes. Adaptive-window methods such as ADWIN address this issue by modifying the effective observation window in response to changes in the stream [4]. ADAPT-IDS follows the same general motivation but uses an event-driven reference rebaselining mechanism rather than directly implementing ADWIN.

The reference update can be expressed generally as

$$R_{t+1} = U\left(R_t, C_t\right),$$

where $U(\cdot)$ denotes the reference-update operation.

The reference-management component therefore allows subsequent drift measurements to be made against a more recent representation of network behavior after a confirmed adaptation event.

4.6. Multi-Metric Unsupervised Drift Detection

The central drift-monitoring component of ADAPT-IDS is a multi-metric detector. Fully or partially label-independent drift detection is important for streaming applications in which ground-truth labels are not continuously available [9, 11]. At the same time, different detectors respond differently to changes in distribution, making the choice of a single statistic potentially restrictive [7, 10].

ADAPT-IDS combines four complementary indicators:

- Jensen–Shannon divergence (JS);
- standardized distributional difference (ED);
- statistical drift significance (DS); and
- residual exceedance rate (RS).

The four indicators are intended to capture different manifestations of change in network traffic. A new traffic regime may produce a broad distributional shift, a change in feature location relative to historical variability, statistically significant changes in individual features, or an increase in deviation from the learned benign profile.

The indicators are therefore treated as complementary evidence rather than interchangeable alternatives. Their combination forms the basis of the drift score used for subsequent threshold and persistence decisions.

4.6.1. Jensen–Shannon Divergence

The Jensen–Shannon divergence measures the discrepancy between two probability distributions. It is derived from the Kullback–Leibler divergence introduced by Kullback and Leibler [50, 51].

For probability distributions P and Q,

$$JS(P, Q) = \frac{1}{2} KL(P \parallel M) + \frac{1}{2} KL(Q \parallel M),$$

where

$$M = \frac{P + Q}{2}.$$

Unlike the directional KL divergence, JS divergence is symmetric with respect to its two input distributions. Larger values indicate greater distributional discrepancy.

In ADAPT-IDS, histogram-based empirical distributions are constructed for the reference and current observations before calculating the JS component. The resulting value is used as one of the inputs to the multi-metric fusion stage.

4.6.2. Standardized Distributional Difference

The ED component captures changes in feature-level location relative to the variability observed in the reference data. For feature j, the standardized mean difference is defined as

$$ED_j = \frac{\left|\mu_{C,j} - \mu_{R,j}\right|}{\sigma_{R,j} + \epsilon},$$

where $\mu_{R,j}$ and $\mu_{C,j}$ are the reference and current means, respectively, $\sigma_{R,j}$ is the reference standard deviation, and ϵ is a small numerical constant used to avoid division by zero.

An aggregated ED score is subsequently obtained across the monitored features according to the implementation. The ED component provides information about changes in feature location relative to the reference variability and therefore complements the broader distributional information captured by JS.

4.6.3. Statistical Drift Significance

The DS component evaluates whether observed differences between reference and current feature distributions are statistically significant. The Kolmogorov–Smirnov (KS) test is used because it provides a non-parametric comparison of empirical distributions [54].

The KS statistic is

$$D_{KS} = \sup_x \left| F_R(x) - F_C(x) \right|,$$

where $F_R(x)$ and $F_C(x)$ denote the empirical cumulative distribution functions of the reference and current samples, respectively.

ADAPT-IDS applies the KS procedure to the selected monitored features and uses multiple-testing correction when aggregating feature-level significance. This reduces the risk of treating a large number of individual hypothesis tests as significant simply because many tests are performed.

The DS component consequently provides a statistical-significance view of drift that differs from the magnitude-oriented information supplied by JS and ED.

4.6.4. Residual Exceedance Analysis

The RS component measures the proportion of current observations whose reconstruction errors exceed a threshold learned from benign traffic. A reconstruction model is trained using benign observations from the initial model-development period.

For an input vector x, let $\hat{x}$ denote its reconstructed representation. The reconstruction error is defined as

$$e(x) = \left\| x - \hat{x} \right\|_2^2 .$$

A reconstruction threshold τ_{AE} is estimated from the empirical distribution of benign training reconstruction errors. In the present implementation, the threshold corresponds to the 95th percentile.

For current batch C_t, the residual exceedance rate is

$$RS_t = \frac{1}{|C_t|} \sum_{x \in C_t} \mathbb{I}\left[e(x) > \tau_{AE} \right].$$

The RS component is not used as an independent attack classifier. Instead, it provides an additional behavioral signal indicating that an increasing proportion of incoming observations differs from the learned benign profile. Such feature-space monitoring is relevant to evolving cybersecurity environments, where both legitimate traffic and malicious behavior can change over time [21, 34].

4.7. Weighted Multi-Metric Score Fusion

The four component indicators are combined into a single operational drift score:

$$D_t = w_{JS} JS_t + w_{ED} ED_t + w_{DS} DS_t + w_{RS} RS_t.$$

The weights used in the experiments are

$$w_{JS} = 0.40, \qquad w_{ED} = 0.25, \qquad w_{DS} = 0.20, \qquad w_{RS} = 0.15.$$

Therefore,

$$D_t = 0.40 JS_t + 0.25 ED_t + 0.20 DS_t + 0.15 RS_t.$$

The largest weight is assigned to JS because it provides the primary global distributional discrepancy signal. ED, DS, and RS contribute additional information concerning feature-level magnitude, statistical significance, and deviation from the benign profile, respectively.

The particular weighting scheme is an ADAPT-IDS design choice. It is not intended to reproduce the weighting of an existing drift detector. Its purpose is to combine heterogeneous evidence into a single score that can subsequently be evaluated using calibrated decision constraints.

4.8. Threshold Calibration

The drift threshold is calibrated using reference-only data rather than being selected from the evaluation stream. Reference subsets are compared with other subsets drawn from the same reference regime to estimate the level of fused-score variation that can occur without an externally imposed temporal change.

The resulting reference-derived scores are used to establish a high-quantile operational threshold:

$$\tau_D = Q_{0.995}\left(D_{ref}\right),$$

where $Q_{0.995}$ denotes the 99.5th percentile of the fused scores obtained from the reference data.

A current batch satisfies the fused-score condition when

$$D_t \geq \tau_D.$$

ADAPT-IDS additionally imposes two minimum-effect constraints:

$$JS_t \geq 0.01$$

and

$$ED_t \geq 0.05.$$

The three conditions jointly prevent a very small distributional difference from becoming an operational drift event solely because the fused score exceeds its calibrated threshold.

4.9. Persistence-Based Drift Confirmation

A threshold crossing does not immediately trigger IDS retraining. Network traffic can exhibit short-lived deviations caused by temporary workloads, isolated attacks, service changes, or sampling variability. Treating every such deviation as a persistent environmental change can result in unnecessary model updates.

ADAPT-IDS therefore maintains a distinction between a *candidate drift* and a *confirmed persistent drift*. Let

$$z_t = \begin{cases} 1, & \text{if batch } t \text{ satisfies the candidate-drift conditions,} \\ 0, & \text{otherwise.} \end{cases}$$

A persistent drift event is confirmed when

$$\sum_{i=t-W+1}^{t} z_i \geq K.$$

The experimental configuration uses

$$W = 5, \qquad K = 3.$$

Thus, at least three candidate drift indications must occur within five consecutive batches before the adaptation stage is activated.

The persistence mechanism is designed to reduce reactions to isolated statistical deviations. It should be interpreted as a detector-level confirmation rule rather than as ground-truth temporal annotation. Unless independent drift labels are available, a confirmed event indicates that the ADAPT-IDS criteria have been satisfied; it does not establish that the underlying data-generating process changed at that exact timestamp.

4.10. Selective Labeling and Adaptive IDS Retraining

After persistent drift has been confirmed, ADAPT-IDS activates the adaptation module. The objective is to obtain sufficient supervised information for model updating without requiring continuous labeling of the entire network stream.

In the experimental configuration, up to 2,000 observations from a confirmed drift event are subjected to oracle labeling. In parallel, high-confidence benign observations are selected using the reconstruction-error profile. Up to 500 low-error observations are retained as benign candidates.

The adaptation data are represented as

$$A_t = A_t^{oracle} \cup A_t^{benign}.$$

These observations are incorporated into the replay buffer before the IDS is retrained.

The updated model can be expressed as

$$M_{t+1} = \mathcal{L}\left(B_t\right),$$

where B_t denotes the balanced replay buffer.

This design maintains an important distinction between the two layers of the architecture. Drift monitoring remains label-independent during routine stream processing, while supervised labels are introduced only after the detector has confirmed persistent change. The approach is therefore related to label-efficient adaptation strategies for evolving streams and adaptive intrusion detection [11, 21, 22].

4.11. Balanced Replay Buffer

Retraining an IDS exclusively on newly labeled observations can make the updated model overly specialized to the most recent traffic regime. It can also reduce the representation of previously observed attack patterns. Maintaining historical information during continual adaptation is therefore an important consideration in evolving learning systems [59, 60].

ADAPT-IDS uses a class-balanced replay buffer with separate capacity limits. The implemented configuration retains up to

10,000

benign observations and

10,000

attack observations.

The replay memory can be represented as

$$B_t = B_{t-1} \cup A_t,$$

followed by class-wise capacity control.

The replay mechanism has two primary purposes. First, previously acquired observations remain available during subsequent retraining. Second, class-wise capacity control prevents the adaptation set from being determined solely by the most recent class distribution.

4.12. Reference Rebaselining

Following model adaptation, continuing to compare incoming traffic against the original historical reference would progressively increase the discrepancy between the reference and the current operating environment. ADAPT-IDS therefore rebaselines the reference following a confirmed adaptation event.

The new reference is constructed from recent batches:

$$R_{t+1} = \text{Select}_{R_{\max}}\left(C_{t-k+1} \cup \cdots \cup C_t\right),$$

where $R_{\max}$ denotes the maximum reference size.

The experimental configuration uses the most recent 10 batches for rebaselining and limits the resulting reference representation to 50,000 observations.

This mechanism is conceptually related to adaptive-window approaches that modify the effective historical context as stream behavior changes [4]. The specific event-driven rebaselining procedure used in ADAPT-IDS, however, is part of the proposed architecture.

4.13. Prequential Processing Sequence

ADAPT-IDS follows a fixed chronological sequence for each incoming batch:

1. Receive the next network batch.
2. Apply the preprocessing transformation learned from the initial training data.
3. Generate IDS predictions using the currently deployed classifier.
4. Compare the incoming feature distribution with the current reference.
5. Calculate JS_t, ED_t, DS_t, and RS_t.
6. Calculate the weighted fused drift score D_t.
7. Apply the calibrated threshold and minimum-effect constraints.
8. Record a candidate drift when all candidate conditions are satisfied.
9. Update the persistence history.
10. Confirm persistent drift when the persistence criterion is satisfied.
11. If persistent drift is confirmed, obtain the required selective oracle labels.
12. Select high-confidence benign observations.
13. Update the balanced replay buffer.
14. Retrain the IDS using the replay buffer.
15. Rebaseline the reference distribution.
16. Continue processing the next batch.

The ordering is deliberate. Predictions for the current batch are generated before any adaptation triggered by that same batch is performed. Consequently, the reported prequential performance corresponds to the state of the deployed IDS when the observations actually arrive.

4.14. End-to-End ADAPT-IDS Workflow

The complete processing pipeline can be represented as

$$\boxed{\text{Network Stream} \rightarrow \text{Preprocessing} \rightarrow \text{Reference/Current Windows} \rightarrow \text{JS+ED+DS+RS}}$$

followed by

$$\boxed{\text{Weighted Fusion} \rightarrow \text{Threshold and Effect Tests} \rightarrow \text{Persistence Confirmation}}$$

and, when persistent drift is confirmed,

$$\boxed{\text{Selective Labeling} \rightarrow \text{Balanced Replay} \rightarrow \text{IDS Retraining} \rightarrow \text{Reference Rebaselining} \rightarrow \text{Continued Monitoring.}}$$

The resulting process forms a closed adaptive loop. The IDS continues to monitor incoming traffic using the currently deployed model, while model updating is activated only when the drift-management layer identifies persistent evidence that satisfies the predefined decision criteria.

4.15. Architectural Characteristics and Design Rationale

The principal characteristics of ADAPT-IDS are summarized as follows.

Unsupervised drift monitoring: Routine drift monitoring does not require ground-truth labels from the incoming stream. This is important in applications where labels are delayed or unavailable and is consistent with research on drift detection from unlabeled streaming data [9, 11].

Multi-metric evidence: Four complementary indicators are combined to represent different forms of distributional and behavioral change. The fused score is therefore not dependent on a single statistical signal.

Persistence-aware confirmation: Candidate drift detections must satisfy a temporal persistence rule before they can trigger adaptation. This reduces the influence of isolated fluctuations on model-update decisions.

Selective labeling: Oracle labeling is activated only following confirmed persistent drift. The design therefore reduces the need to continuously label the entire evaluation stream.

Replay-based adaptation: Historical observations are retained together with newly acquired samples during retraining. This provides a mechanism for preserving previously observed traffic and attack patterns during successive model updates [59, 60].

Dynamic reference management: The reference distribution is reconstructed after confirmed adaptation, allowing subsequent drift measurements to be performed relative to a more recent traffic regime.

Leakage-controlled prequential evaluation: The current batch is predicted before any adaptation caused by that batch. This preserves the chronological nature of the evaluation and prevents the adapted model from influencing its own batch-level performance measurement.

Overall, ADAPT-IDS is a hybrid adaptive intrusion-detection architecture with an unsupervised drift-monitoring layer and a selectively supervised adaptation layer. Its contribution is not a claim that JS divergence, KS testing, replay learning, or adaptive windows are individually new. Rather, the framework integrates multi-metric distribution monitoring, minimum-effect constraints, persistence-based confirmation, selective labeling, balanced replay, and event-driven reference rebaselining into a single chronological IDS pipeline. These design choices are motivated by established work on concept drift, unlabeled-stream detection, and adaptive intrusion detection [1, 9, 11, 21, 22, 25].

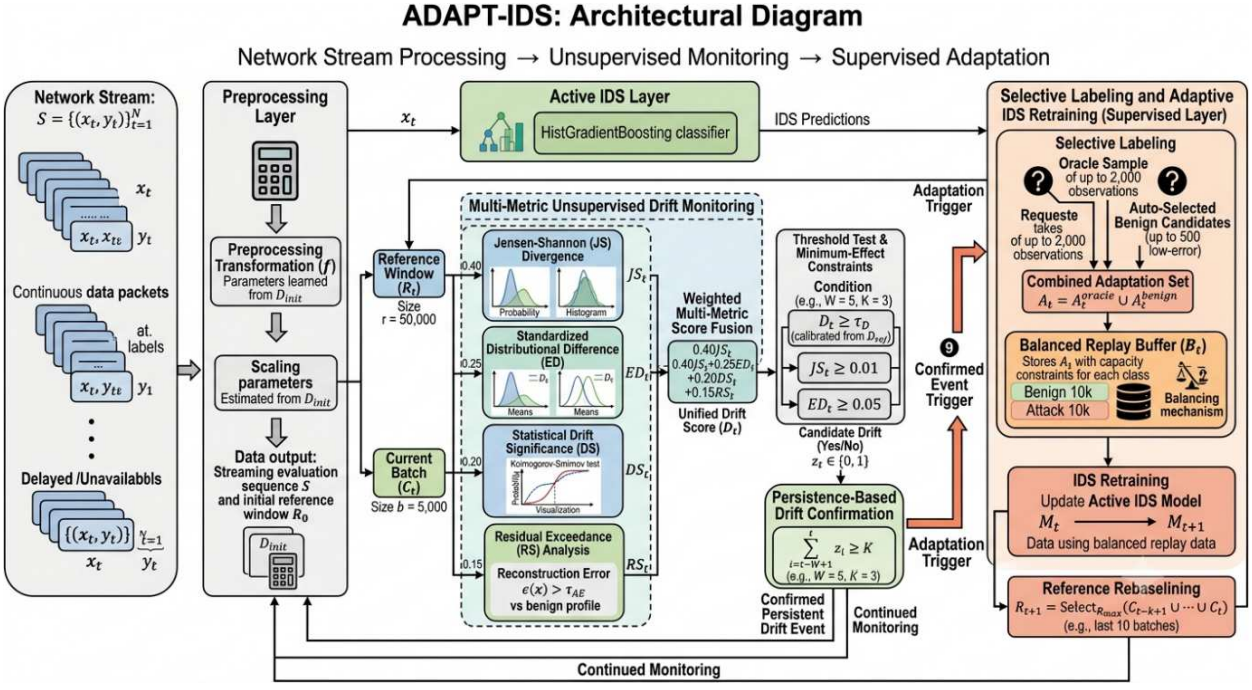


Figure 6: ADAPT-IDS architectural diagram.

Algorithm 1 ADAPT-IDS: Unsupervised Multi-Metric Drift Detection and Adaptive IDS

Require: Network stream S, initial data D_{init}, reference size R, batch size B, persistence parameters (W, K), oracle budget L

Ensure: Predictions $\widehat{Y}$, drift events $\mathcal{D}$, adapted models $\mathcal{M}$

Sort S chronologically and split into D_{init}, reference window R, and streaming batches X_C^t
Remove timestamp/label attributes; fit median imputation and robust scaling on D_{init}
Train initial IDS M_0 and benign reconstruction model A; set τ_{AE} at the 95th percentile
Initialize reference statistics, replay buffer Q, persistence history H, and fused threshold τ_D
for each batch X_C^t **do**
    Predict $\widehat{Y}_t = M(X_C^t)$
    Compute JS_t, ED_t, DS_t, and RS_t
    Compute $D_t = 0.40JS_t + 0.25ED_t + 0.20DS_t + 0.15RS_t$
    if $D_t \geq \tau_D$, $JS_t \geq 0.01$, and $ED_t \geq 0.05$ **then**
        Append candidate alarm to H and retain the last $W = 5$ decisions
        if $\sum H \geq K = 3$ **then**
            Confirm persistent drift and obtain oracle samples within budget L
            Add oracle-labeled and high-confidence benign samples to balanced buffer Q
            Retrain M using Q
            Rebaseline reference distributions
            Reset H
        end if
    end if
end for
return $\widehat{Y}$, $\mathcal{D}$, $\mathcal{M}$, and evaluation statistics

The parameterization used in the present study is

$$w_{JS} = 0.40, \qquad w_{ED} = 0.25, \qquad w_{DS} = 0.20, \qquad w_{RS} = 0.15,$$

$$W = 5, \qquad K = 3, \qquad JS_{\min} = 0.01, \qquad ED_{\min} = 0.05,$$

with a drift-threshold quantile of 99.5% and a reconstruction threshold quantile of 95%. The reference window contains up to 50,000 observations, each incoming batch contains 5,000 observations, up to 2,000 observations may be subjected to oracle labeling after a confirmed drift event, and up to 500 high-confidence benign observations are selected for adaptation.

‘‘‘latex

5. Experimental Evaluation

This section evaluates ADAPT-IDS under evolving network traffic using a chronological prequential evaluation protocol. The evaluation is designed to examine drift detection, persistence-based confirmation, drift-triggered model adaptation, selective labeling, and computational overhead. For each incoming batch, the current IDS first generates predictions. The batch is then analyzed by the unsupervised drift detector, and model adaptation is performed only when the persistence criterion confirms a drift event.

5.1. Experimental Datasets

5.1.1. CIC-IDS-2018

CIC-IDS-2018 is used as the primary dataset for evaluating ADAPT-IDS under a chronological streaming setting. The dataset contains benign and malicious network-flow records generated from controlled network scenarios. CIC intrusion-detection datasets are widely used for evaluating machine-learning-based network intrusion detection systems [41–43].

The records were retained in temporal order rather than randomly shuffled. The timestamp was used only to establish the chronological order of observations and was not included as an input feature. After cleaning and preprocessing,

the resulting dataset contains 2,097,150 instances represented by 63 numerical features. The first 200,000 observations are used for initial model development. The remaining 1,897,150 observations form the evaluation stream. A reference window of 50,000 observations and a batch size of 5,000 observations are used during streaming evaluation, resulting in 380 sequential evaluation batches.

5.1.2. CIC-IDS-2017

CIC-IDS-2017 is used as an independent dataset to examine whether the proposed detection and adaptation procedure remains applicable under a different network-traffic distribution. The dataset is processed using the same general chronological and prequential methodology. A total of 312,212 observations are evaluated over 63 sequential batches.

The purpose of this experiment is independent verification rather than a claim of universal superiority over existing IDS approaches.

5.2. Chronological Data Preparation and Preprocessing

Let the initial labeled dataset be

$$D_{\text{init}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{init}}}, \qquad N_{\text{init}} = 200,000, \tag{1}$$

where x_i denotes the numerical feature vector and y_i denotes the corresponding intrusion label. The remaining observations form the streaming sequence

$$S = \{x_t\}_{t=N_{\text{init}}+1}^{N}. \tag{2}$$

All observations are sorted according to their timestamps before streaming evaluation. No random shuffling is applied. Missing numerical values are imputed using statistics estimated from the initial training data. Robust scaling parameters are also estimated only from the initial training portion and subsequently applied to the streaming observations.

For CIC-IDS-2018, the final feature representation contains 63 numerical variables. The timestamp is retained only for chronological ordering, while the class label is excluded from the drift detector. Consequently, the unsupervised monitoring component receives only the feature observations X_t and does not access the corresponding labels Y_t during routine drift detection.

5.3. Initial IDS Model

The initial IDS model is trained using the labeled initial dataset. A HistGradientBoosting classifier is used because the network-flow features can exhibit nonlinear relationships and interactions while remaining suitable for large tabular datasets.

The principal classifier parameters are:

- learning rate: 0.08;
- maximum iterations: 180;
- maximum leaf nodes: 31;
- minimum samples per leaf: 50;
- L_2 regularization: 1.0;
- random seed: 42.

Let the initial classifier be denoted by M_0. During streaming evaluation,

$$M_t = \begin{cases} M_{t-1}, & \text{if no persistent drift is confirmed,} \\ \mathcal{T}(Q_t), & \text{if persistent drift is confirmed,} \end{cases} \tag{3}$$

where Q_t denotes the replay-based adaptation dataset and $\mathcal{T}(\cdot)$ represents the classifier training procedure.

5.4. Benign Reconstruction Model

A separate reconstruction model is trained using benign observations from the initial training data. Its purpose is to provide an additional signal based on deviations from the learned benign traffic representation.

For an observation x_i, let $\hat{x}_i$ denote its reconstructed representation. The reconstruction error is calculated as

$$e_i = \frac{1}{d} \sum_{j=1}^{d} \left(x_{ij} - \hat{x}_{ij} \right)^2 , \tag{4}$$

where d is the number of input features.

The reconstruction threshold is estimated as

$$\tau_{RS} = Q_{0.95} \left(\{ e_i : x_i \in D_{\text{benign}} \} \right) . \tag{5}$$

For an incoming batch X_t, the residual exceedance rate is

$$RS_t = \frac{1}{|X_t|} \sum_{x_i \in X_t} \mathbb{I} \left(e_i > \tau_{RS} \right) . \tag{6}$$

5.5. Unsupervised Drift Detection

For each incoming batch X_t, ADAPT-IDS compares the current observations with the reference window R_t. The detector does not use intrusion labels during this stage. Four complementary signals are calculated: Jensen–Shannon divergence, standardized distributional difference, statistical drift significance, and residual exceedance.

5.5.1. Jensen–Shannon Divergence

For probability distributions P and Q, the mixture distribution is

$$M = \frac{1}{2}(P + Q), \tag{7}$$

and the Jensen–Shannon divergence is

$$JS(P, Q) = \frac{1}{2} D_{\text{KL}}(P \| M) + \frac{1}{2} D_{\text{KL}}(Q \| M). \tag{8}$$

The divergence is calculated from empirical feature distributions [50, 51].

5.5.2. Standardized Distributional Difference

For feature j,

$$ED_j = \frac{|\mu_{C,j} - \mu_{R,j}|}{\sigma_{R,j} + \epsilon}, \tag{9}$$

where $\mu_{C,j}$ and $\mu_{R,j}$ are the current and reference means, respectively, and $\sigma_{R,j}$ is the reference standard deviation.

5.5.3. Statistical Drift Significance

The Kolmogorov–Smirnov statistic is defined as

$$D_{\text{KS}} = \sup_x \left| F_R(x) - F_C(x) \right| , \tag{10}$$

where $F_R(x)$ and $F_C(x)$ are the empirical cumulative distribution functions of the reference and current windows [54].

Multiple feature-level tests are corrected using the Benjamini–Hochberg false-discovery-rate procedure [55]. The resulting aggregated signal is denoted by DS_t.

5.5.4. Residual Exceedance

The residual signal RS_t is obtained from the benign reconstruction model. A high proportion of observations whose reconstruction errors exceed τ_{RS} provides additional evidence of a change in the current traffic distribution.

5.5.5. Weighted Evidence Fusion

The four signals are combined as

$$D_t = 0.40JS_t + 0.25ED_t + 0.20DS_t + 0.15RS_t. \tag{11}$$

The drift threshold is

$$\tau_D = Q_{0.995}(D_{\text{ref}}). \tag{12}$$

A candidate drift is generated when

$$D_t > \tau_D, \qquad JS_t \geq 0.01, \qquad ED_t \geq 0.05. \tag{13}$$

5.6. Persistence-Based Drift Confirmation

Let

$$C_t \in \{0, 1\} \tag{14}$$

denote the candidate-drift indicator for batch t. Persistent drift is confirmed when at least K candidate alarms occur within the most recent W batches:

$$\sum_{i=t-W+1}^{t} C_i \geq K. \tag{15}$$

The experimental configuration uses

$$W = 5, \qquad K = 3. \tag{16}$$

Candidate alarms are intermediate detector outputs and should not be interpreted as ground-truth drift annotations. Only confirmed events are allowed to trigger model adaptation.

5.7. Prequential Evaluation Protocol

The streaming evaluation follows the order:

1. receive the next chronological batch;
2. apply preprocessing;
3. generate IDS predictions;
4. calculate the drift indicators;
5. fuse the drift evidence;
6. apply the threshold and minimum-effect constraints;
7. update the persistence state;

8. confirm persistent drift;
9. obtain selective oracle labels;
10. select high-confidence benign observations;
11. construct the replay dataset;
12. retrain the IDS;
13. update the reference window.

Predictions are therefore generated before adaptation based on the current batch. This avoids using post-detection information to produce predictions for the same observations [1, 59].

5.8. Selective Labeling and Adaptive Retraining

When persistent drift is confirmed, up to 2,000 observations are selected for oracle labeling for each confirmed event in the CIC-IDS-2018 experiment. In addition, up to 500 high-confidence benign observations are automatically selected.

The replay buffer contains up to 10,000 benign and 10,000 attack observations. The newly labeled observations are combined with retained historical examples before retraining the HistGradientBoosting classifier.

5.9. Reference-Window Rebaselining

Following adaptation, the reference window is updated using recent stream observations. The most recent 10 batches are considered, subject to a maximum reference size of 50,000 observations.

$$R_{t+1} = \text{RecentWindow}\left(X_{t-k+1}, \ldots, X_t\right). \tag{17}$$

The drift statistics are recalculated using the updated reference so that subsequent monitoring focuses on changes relative to the current traffic regime.

5.10. Evaluation Metrics

Accuracy is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{18}$$

Precision is

$$Precision = \frac{TP}{TP + FP}. \tag{19}$$

Detection rate is

$$DR = \frac{TP}{TP + FN}. \tag{20}$$

The F1-score is

$$F1 = 2\frac{Precision \times DR}{Precision + DR}. \tag{21}$$

Balanced accuracy is

Table 4
Prequential IDS performance on CIC-IDS-2018.

Metric	ADAPT-IDS
Accuracy	97.4587%
Precision	99.9367%
Detection Rate	85.8773%
F1-score	92.3751%
Balanced Accuracy	92.9327%
MCC	0.9124
False-Alarm Rate	0.0119%

Table 5
Confusion matrix for CIC-IDS-2018.

	Predicted Benign	Predicted Attack
Actual Benign	1,556,895	185
Actual Attack	48,027	292,043

$$BA = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{22}$$

The Matthews correlation coefficient is

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}. \tag{23}$$

The false-alarm rate is

$$FAR = \frac{FP}{FP+TN}. \tag{24}$$

These metrics provide complementary information about classification performance, particularly under class imbalance [61].

5.11. Primary Results on CIC-IDS-2018

The CIC-IDS-2018 evaluation contains 1,897,150 sequential observations processed in 380 batches. ADAPT-IDS generated 327 candidate drift detections, of which 73 satisfied the persistence criterion and triggered model adaptation.

The corresponding confusion matrix is

Only 185 benign observations were incorrectly classified as attacks. The resulting precision is 99.9367%, while the false-alarm rate is 0.0119%. The 48,027 false-negative attack observations account for the lower detection rate of 85.8773%.

5.12. Drift Detection and Adaptation Results

The persistence mechanism therefore reduces the number of candidate events that directly trigger adaptation. Candidate detections represent intermediate evidence, whereas only confirmed persistent events initiate retraining.

5.13. Labeling Efficiency

A maximum of 2,000 oracle-labeled observations is requested for each of the 73 confirmed events, resulting in 146,000 oracle-labeled observations.

Table 6
Drift detection and adaptation statistics for CIC-IDS-2018.

Measure	Value
Evaluation batches	380
Candidate drift batches	327
Candidate drift rate	86.05%
Confirmed persistent drift events	73
Confirmed event rate	19.21%
Candidate-to-confirmed ratio	22.32%
Adaptation events	73
Models retrained	73
Reference updates	73
Oracle samples	146,000
Automatically selected benign samples	36,500
Mean adaptation time	9.74 s

Table 7
Prequential IDS performance on CIC-IDS-2017.

Metric	ADAPT-IDS
Accuracy	94.4576%
Precision	99.9711%
Detection Rate	89.1380%
F1-score	94.2442%
Balanced Accuracy	94.5556%
MCC	0.8948
False-Alarm Rate	0.0267%

Table 8
Confusion matrix for CIC-IDS-2017.

	Predicted Benign	Predicted Attack
Actual Benign	153,241	41
Actual Attack	17,263	141,667

$$\frac{146,000}{1,897,150} \times 100 = 7.70\%. \tag{25}$$

Thus, 92.30% of the evaluation-stream observations were processed without oracle labeling. In addition, 36,500 benign observations were selected automatically.

5.14. Independent Verification on CIC-IDS-2017

The CIC-IDS-2017 experiment evaluates 312,212 observations over 63 sequential batches. ADAPT-IDS achieves 94.4576% accuracy, 99.9711% precision, 89.1380% detection rate, and 94.2442% F1-score. Balanced accuracy is 94.5556%, MCC is 0.8948, and the false-alarm rate is 0.0267%.

The class-wise performance is given in Table 9.

The drift-monitoring procedure generated 63 candidate drift batches and 13 confirmed persistent drift events. All 13 confirmed events resulted in model adaptation.

Table 9
Class-wise performance on CIC-IDS-2017.

Class	Precision	Recall	F1-score	Support
Benign	89.8753%	99.9733%	94.6557%	153,282
Attack	99.9711%	89.1380%	94.2442%	158,930

Table 10
Drift detection and adaptation statistics for CIC-IDS-2017.

Measure	Value
Evaluated samples	312,212
Evaluated batches	63
Candidate drift batches	63
Confirmed persistent drift events	13
Adaptation events	13
Models retrained	13
Reference updates	13
Oracle samples	26,000
Oracle attack labels	10,387
Automatically selected benign samples	6,200
Mean adaptation time	8.04 s

Table 11
Cross-dataset comparison of ADAPT-IDS.

Measure	CIC-IDS-2018	CIC-IDS-2017
Evaluated instances	1,897,150	312,212
Evaluation batches	380	63
Accuracy	97.46%	94.46%
Precision	99.94%	99.97%
Detection Rate	85.88%	89.14%
F1-score	92.38%	94.24%
Balanced Accuracy	92.93%	94.56%
MCC	0.9124	0.8948
False-Alarm Rate	0.0119%	0.0267%
Candidate drift batches	327	63
Confirmed persistent events	73	13
Adaptation events	73	13
Oracle samples	146,000	26,000
Oracle-label fraction	7.70%	8.33%
Mean adaptation time	9.74 s	8.04 s

5.15. Cross-Dataset Comparison

Precision remains above 99.9% on both datasets. CIC-IDS-2017 produces a higher detection rate and F1-score, whereas CIC-IDS-2018 produces a lower false-alarm rate and a higher MCC. These differences reflect the different traffic and attack distributions of the two datasets.

5.16. Computational Cost of Adaptation

ADAPT-IDS performs model retraining only after persistent drift rather than after every incoming batch. On CIC-IDS-2018, 73 adaptation events were observed, with a mean adaptation time of 9.74 s. On CIC-IDS-2017, 13 adaptation events were observed, with a mean adaptation time of 8.04 s.

The adaptation time can be expressed as

Table 12
Feature-level ablation results.

Configuration	Features	Accuracy	Precision	DR	F1	BA	MCC	ROC-AUC	PR-AUC	FAR
All features	63	0.972296	0.999791	0.845626	0.916269	0.922794	0.904340	0.998623	0.993421	0.000039
No DstPort	62	0.973910	0.998839	0.855444	0.921597	0.927614	0.909975	0.999253	0.997261	0.000217
No Protocol	62	0.972296	0.999791	0.845626	0.916269	0.922794	0.904340	0.998623	0.993421	0.000039
No DstPort + Protocol	61	0.973910	0.998839	0.855444	0.921597	0.927614	0.909975	0.999253	0.997261	0.000217

$$T_{\text{adapt}} = T_{\text{label}} + T_{\text{selection}} + T_{\text{replay}} + T_{\text{training}} + T_{\text{reference}}. \tag{26}$$

The measured time is implementation-dependent and should therefore be interpreted together with the experimental hardware and software environment.

5.17. Feature Ablation Analysis

Removing *DstPort* increases the detection rate from 0.845626 to 0.855444 and the F1-score from 0.916269 to 0.921597. ROC-AUC and PR-AUC also increase. However, the FAR increases from 0.000039 to 0.000217. Removing *Protocol* alone produces the same reported results as the full feature configuration.

5.18. Validity and Leakage Control

The following controls were applied during the experiments:

- observations were processed chronologically without random shuffling;
- preprocessing parameters were estimated from the initial training data only;
- the initial IDS was trained before streaming evaluation;
- intrusion labels were excluded from routine drift monitoring;
- predictions were generated before adaptation;
- oracle labels were requested only after persistent drift confirmation;
- replay data combined newly acquired observations with retained historical samples;
- CIC-IDS-2017 was used as an independent verification dataset.

These controls reduce temporal leakage and ensure that the reported predictions correspond to the model available at each point in the stream.

5.19. Experimental Summary

The CIC-IDS-2018 experiment processes more than 1.89 million stream observations across 380 chronological batches. ADAPT-IDS generates 327 candidate drift alarms, of which 73 satisfy the persistence criterion and trigger adaptation. The resulting IDS achieves 97.46% accuracy, 99.94% precision, 85.88% detection rate, 92.38% F1-score, 92.93% balanced accuracy, and an MCC of 0.9124, with a false-alarm rate of 0.0119%.

On CIC-IDS-2017, the framework achieves 94.46% accuracy, 99.97% precision, 89.14% detection rate, 94.24% F1-score, 94.56% balanced accuracy, and an MCC of 0.8948. The false-alarm rate is 0.0267%. Oracle labeling accounts for 7.70% of the CIC-IDS-2018 evaluation stream and 8.33% of the CIC-IDS-2017 stream.

Overall, the experiments show that ADAPT-IDS can combine unsupervised drift monitoring, persistence-based confirmation, and selective model adaptation while maintaining high attack precision and low false-alarm rates on the evaluated datasets. These results should not be interpreted as universal superiority over existing drift detectors or adaptive IDS methods. Direct comparison under identical chronological protocols is required for such a claim. “‘

6. Conclusion and Future Work

6.1. Conclusion

This study presented **ADAPT-IDS**, an unsupervised multi-metric framework for reliable concept drift detection and adaptive intrusion detection in evolving network streams. The framework addresses a fundamental limitation of conventional intrusion detection systems, which often assume that the statistical characteristics of network traffic remain relatively stable after model training. In operational environments, however, traffic behavior, application usage, attack strategies, and feature distributions can change over time. Such changes can gradually reduce the effectiveness of a previously trained IDS.

ADAPT-IDS addresses this problem by combining four complementary sources of drift evidence: **Jensen–Shannon divergence (JS), standardized distributional difference (ED), statistical drift significance (DS), and residual exceedance (RS)**. The four signals are integrated through a weighted evidence-fusion mechanism rather than relying on a single statistical measure. Minimum-effect constraints are additionally applied to prevent statistically detectable but practically weak changes from immediately triggering adaptation. A persistence-based confirmation mechanism then distinguishes sustained changes from isolated candidate alarms by requiring at least three candidate drift decisions within a five-batch observation window.

The framework also introduces an event-driven adaptation strategy that limits the need for continuous manual labeling. Ground-truth labels are not used during routine drift monitoring. Instead, labels are introduced only after persistent drift has been confirmed. Selective oracle samples are combined with high-confidence benign observations obtained from the reconstruction model and stored through a balanced replay strategy. The IDS is subsequently retrained using the resulting adaptation data, while the reference distribution is updated using recent traffic observations. This allows the model to incorporate information from a new traffic regime while retaining previously learned information.

The primary evaluation on the chronologically ordered CIC-IDS-2018 stream processed **1,897,150 prequential evaluation instances across 380 batches**. ADAPT-IDS identified **327 candidate drift batches** and confirmed **73 persistent drift events**, resulting in 73 adaptive model updates. The resulting IDS achieved **97.46% accuracy, 99.94% precision, 85.88% detection rate, 92.38% F1-score, 92.93% balanced accuracy, and an MCC of 0.9124**, with a false-alarm rate of **0.0119%**. The results show that the proposed framework can maintain high attack-prediction precision while processing a long chronological stream containing evolving traffic patterns.

An independent evaluation on CIC-IDS-2017 provided additional evidence of the applicability of the proposed architecture. Across **312,212 observations and 63 sequential batches**, ADAPT-IDS achieved **94.46% accuracy, 99.97% precision, 89.14% detection rate, 94.24% F1-score, 94.56% balanced accuracy, and an MCC of 0.8948**, with a false-alarm rate of **0.0267%**. The framework generated 63 candidate drift batches and confirmed 13 persistent drift events, resulting in 13 adaptive model updates. The results obtained on the two datasets show that the proposed procedure is not restricted to a single traffic distribution, although broader evaluation is required before making stronger generalization claims.

The experiments also demonstrate the labeling efficiency of the event-driven adaptation strategy. On CIC-IDS-2018, **146,000 oracle-labeled observations**, corresponding to approximately **7.70% of the evaluation stream**, were used for adaptation. On CIC-IDS-2017, the corresponding oracle-labeling fraction was **8.33%**. Consequently, most evaluation-stream observations were processed without direct oracle annotation. This result supports the use of selective labeling when continuous labeling of network traffic is impractical.

The feature-level ablation experiment further showed that the contribution of individual network-flow attributes is not uniform. Removing the destination-port feature increased the F1-score from 0.9163 to 0.9216 and improved the detection rate from 84.56% to 85.54%, although the false-alarm rate also increased. Removing the protocol feature alone produced no measurable difference under the evaluated configuration. These findings indicate that feature redundancy and feature-specific effects should be considered when designing IDS models for evolving network environments.

Overall, the experimental findings support the central design premise of ADAPT-IDS: **adaptive intrusion detection under concept drift requires both reliable drift evidence and a controlled mechanism for deciding when and how model adaptation should occur**. The combination of heterogeneous drift signals, minimum-effect constraints, temporal persistence, selective labeling, historical replay, and dynamic reference management provides a structured adaptation process without requiring continuous labeling of the complete network stream.

The reported results should not, however, be interpreted as proof of universal superiority over existing concept-drift-aware IDS frameworks. In particular, direct comparisons with closely related approaches such as HOIDS and

established drift detectors require matched datasets, preprocessing procedures, feature representations, stream construction, and evaluation protocols. The present experiments therefore provide empirical evidence for the effectiveness of ADAPT-IDS under the evaluated settings while also identifying several areas that require further validation.

6.2. Future Work

Despite the encouraging results, several limitations and research opportunities remain.

6.2.1. Component-Level Drift Detector Ablation

A systematic ablation of the proposed drift detector is an important next step. Separate configurations should evaluate the progressive addition of the individual drift signals:

$$JS \rightarrow JS + ED \rightarrow JS + ED + DS \rightarrow JS + ED + DS + RS \rightarrow JS + ED + DS + RS + \text{Persistence}. \quad (27)$$

Such experiments would quantify the individual and cumulative contribution of each component and determine whether the observed behavior results from complementary evidence or is dominated by a particular metric. This analysis should report both drift-detection measures and downstream IDS performance.

6.2.2. Comparative Evaluation with Established Drift Detectors

Future experiments should compare ADAPT-IDS with established drift-detection methods, including adaptive-window, statistical, sequential, and fully unsupervised approaches. Suitable baselines include **ADWIN, DDM/EDDM where labels are permissible, KS-based detection, Jensen–Shannon divergence, Wasserstein distance, Page–Hinkley, and contemporary unsupervised concept-drift detectors** [1, 4–6, 9].

The comparison should use identical chronological streams and preprocessing procedures. In addition to IDS performance, the evaluation should report false-alarm frequency, drift-detection delay, number of confirmed events, number of model adaptations, computational cost, and labeling requirements.

6.2.3. Direct Comparison with HOIDS

HOIDS represents a closely related concept-drift-aware hybrid IDS architecture. A controlled head-to-head comparison between HOIDS and ADAPT-IDS would therefore provide a more meaningful assessment of their relative strengths.

The comparison should use the same datasets, temporal ordering, feature preprocessing, initial training partitions, stream construction, and evaluation metrics. The analysis should consider not only F1-score but also **drift-detection behavior, adaptation frequency, labeling requirements, computational overhead, false alarms, and robustness to evolving attack distributions**. Such an evaluation would provide stronger evidence for determining the relative advantages of the two architectural strategies.

6.2.4. Multi-Dataset and Real-Time Validation

Although CIC-IDS-2018 and CIC-IDS-2017 provide useful experimental settings, future work should extend the evaluation to additional intrusion-detection datasets and, where possible, real network traffic. Datasets representing heterogeneous enterprise, IoT, industrial, cloud, and other network environments would provide a broader basis for evaluating generalization [42, 44–46, 48, 49].

Real-time deployment should also be investigated using a streaming infrastructure such as Apache Kafka or an equivalent event-processing platform. Such experiments could measure throughput, processing latency, memory consumption, drift-detection overhead, and adaptation time under realistic workloads.

6.2.5. Improved Reference-Window Management

The current reference-management strategy uses recent traffic to represent the evolving operating environment. Although this allows the detector to follow long-term changes, the recent window may contain a substantial amount of previously unseen malicious traffic. This creates a potential risk of **reference contamination**.

Future versions should investigate benign-filtered reference construction, confidence-weighted reference updates, multiple historical reference windows, and attack-aware reference validation. Maintaining both short-term and long-term reference distributions may also help distinguish gradual distributional changes from abrupt changes.

6.2.6. Drift Detection Delay and Drift Localization

The current evaluation primarily examines whether persistent drift is detected and whether it subsequently triggers model adaptation. Future research should additionally quantify **detection delay, drift duration, drift magnitude, and drift localization**. These measures are important because identifying a change is not sufficient if the detector responds substantially after the underlying change has already affected IDS performance [7, 8].

For datasets containing known temporal attack transitions, detected drift timestamps can be compared with externally defined event boundaries. This would allow evaluation using measures such as detection delay, missed-drift rate, false-drift rate, and drift-event localization accuracy.

6.2.7. Adaptive Weight Optimization

The current fusion mechanism uses fixed weights:

$$w_{JS} = 0.40, \qquad w_{ED} = 0.25, \qquad w_{DS} = 0.20, \qquad w_{RS} = 0.15. \tag{28}$$

Future research can investigate whether these weights should remain fixed or adapt according to the current traffic regime. Online optimization, Bayesian optimization, meta-learning, or reinforcement-learning-based weight selection could be explored to determine whether the detector can dynamically emphasize the most informative evidence under different forms of distributional change.

Any adaptive weighting strategy should be evaluated carefully to ensure that the optimization process does not introduce label leakage or overfit to a particular stream.

6.2.8. More Efficient Selective Labeling

Although ADAPT-IDS reduces the fraction of observations requiring oracle labels, the labeling strategy can be further optimized. Instead of using a fixed oracle-sample budget, future versions could prioritize observations according to predictive uncertainty, reconstruction error, disagreement between IDS predictions and reconstruction-based decisions, or estimated information gain.

An uncertainty-aware oracle-selection strategy could reduce the number of labels required for adaptation while retaining sufficient information about the new traffic regime. This direction is particularly relevant for operational environments where expert labeling is expensive or slow.

6.2.9. Robustness Against Adversarial and Stealthy Drift

Future investigations should examine whether an adversary can deliberately manipulate network-flow distributions to evade drift detection or gradually contaminate the reference window. Slow, stealthy, and adversarial concept drift represents a particularly difficult case because gradual changes may remain below the detection threshold for extended periods.

Future versions of ADAPT-IDS could therefore investigate trusted reference anchors, drift-rate constraints, independent validation of adaptation samples, and mechanisms for detecting suspicious changes in the reference distribution itself.

6.2.10. Statistical Significance and Repeated Evaluation

Future experiments should use multiple temporal splits, repeated runs with different random seeds, confidence intervals, and statistical significance tests. A single experimental run may not be sufficient to determine whether an observed performance difference is systematic or dependent on a particular temporal partition [61, 62].

Where the dataset structure permits repeated evaluation, results should therefore be reported in the form

$$\text{mean} \pm \text{standard deviation}. \tag{29}$$

Statistical comparison should also be performed when evaluating ADAPT-IDS against competing methods under matched experimental conditions.

6.2.11. Explainable Drift and IDS Adaptation

Another important direction is the integration of explainability into both drift detection and IDS adaptation. Rather than reporting only a numerical drift score, future versions could identify the features that contributed most strongly to the detected change and determine whether the observed changes are associated with particular attack behaviors.

Feature-level drift explanations could help security analysts understand why adaptation was triggered and whether the detected change corresponds to a meaningful alteration in network behavior. This would make the framework more useful in operational security environments where model decisions need to be interpreted by analysts.

6.3. Final Perspective

ADAPT-IDS provides a foundation for **label-efficient, drift-aware, and adaptive intrusion detection** in evolving network streams. Its main contribution is the integration of drift monitoring and model adaptation into a single operational process consisting of multi-metric evidence generation, minimum-effect filtering, temporal confirmation, selective labeling, adaptive learning, historical replay, and reference rebaselining.

The results obtained on CIC-IDS-2018 and independently evaluated on CIC-IDS-2017 demonstrate the feasibility of this approach under the experimental conditions considered in this study. The findings also motivate further investigation into component-level ablation, standardized benchmarking, real-time deployment, reference contamination, adaptive drift weighting, and controlled comparison with existing concept-drift-aware IDS frameworks.

Future validation under broader datasets, repeated temporal evaluations, real network streams, and matched comparisons with established methods will be necessary to determine how well ADAPT-IDS transfers to different operational environments. Such studies can help establish whether the proposed architecture can progress from a dataset-based research framework toward a robust and practically deployable adaptive intrusion-detection system.

CRediT Authorship Contribution Statement

Rohit Kumar Thakur: Conceptualization, Methodology, Software, Data curation, Formal analysis, Investigation, Validation, Visualization, Writing – original draft, Writing – review & editing.

Raj Kumari: Supervision, Conceptualization, Validation, Writing – review & editing.

All authors have read and approved the final version of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the support and guidance provided during the development, implementation, experimentation, and preparation of this research work.

Data Availability

The CIC-IDS-2017 and CIC-IDS-2018 datasets used in this study are publicly available intrusion-detection datasets. The datasets were used for experimental evaluation and independent verification of the proposed ADAPT-IDS framework. The source datasets are subject to the terms and conditions of their respective providers. The code, processed data, and experimental results generated in this study can be made available by the corresponding author upon reasonable request.

References

[1] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, Art. 44, pp. 1–37, 2014, doi: 10.1145/2523813.

[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019, doi: 10.1109/TKDE.2018.2876857.

[3] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015, doi: 10.1109/MCI.2015.2471196.

[4] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448, 2007, doi: 10.1137/1.9781611972771.42.

[5] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams*, 2006.

[6] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence – SBIA 2004*, Springer, pp. 286–295, 2004, doi: 10.1007/978-3-540-28645-5_29.

[7] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: Detecting concept drift," *Frontiers in Artificial Intelligence*, vol. 7, Art. 1330257, 2024, doi: 10.3389/frai.2024.1330257.

[8] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part B: Locating and explaining concept drift," *Frontiers in Artificial Intelligence*, vol. 7, Art. 1330258, 2024, doi: 10.3389/frai.2024.1330258.

[9] D. Lukats, O. Zielinski, A. Hahn, and F. T. Stahl, "A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams," *International Journal of Data Science and Analytics*, vol. 19, pp. 1–31, 2025, doi: 10.1007/s41060-024-00620-y.

[10] S. Arora, R. Rani, and N. Saxena, "A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques," *WIREs Data Mining and Knowledge Discovery*, vol. 14, no. 4, e1536, 2024, doi: 10.1002/widm.1536.

[11] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," *Expert Systems with Applications*, vol. 82, pp. 77–99, 2017, doi: 10.1016/j.eswa.2017.04.008.

[12] K. Nishida and K. Yamauchi, "Detecting concept drift using statistical testing," in *Discovery Science*, Springer, pp. 264–269, 2007, doi: 10.1007/978-3-540-75488-6_27.

[13] A. Pesaranghader and H. L. Viktor, "Fast Hoeffding drift detection method for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 96–111, 2016, doi: 10.1007/978-3-319-46227-1_7.

[14] I. Frias-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 810–823, 2015.

[15] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, pp. 1469–1495, 2017, doi: 10.1007/s10994-017-5642-8.

[16] B. Krawczyk, L. M. G. Silva, G. A. T. de Carvalho, and G. P. Coelho, "Ensemble learning for data stream classification," *Information Fusion*, vol. 37, pp. 132–156, 2017, doi: 10.1016/j.inffus.2017.02.004.

[17] R. Pears, S. Sakthithasan, and Y. S. Koh, "Detecting concept change in dynamic data streams," *Machine Learning*, vol. 97, pp. 259–293, 2014.

[18] S. Sakthithasan, R. Pears, and Y. S. Koh, "One pass concept change detection for data streams," in *PAKDD*, Springer, pp. 461–472, 2013, doi: 10.1007/978-3-642-37456-2_39.

[19] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time ensemble of classifiers," in *International Joint Conference on Neural Networks*, pp. 1–8, 2012, doi: 10.1109/IJCNN.2012.6252540.

[20] J. Lu et al., "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019, doi: 10.1109/TKDE.2018.2876857.

[21] S. Khanna, "Concept drift-based intrusion detection for evolving data stream classification in IDS: Approaches and comparative study," *The Computer Journal*, vol. 67, no. 7, pp. 2529–2547, 2024, doi: 10.1093/comjnl/bxae023.

[22] F. Jemili, K. Jouini, and O. Korbaa, "Intrusion detection based on concept drift detection and online incremental learning," *International Journal of Pervasive Computing and Communications*, vol. 21, no. 1, pp. 81–115, 2025, doi: 10.1108/IJPCC-12-2023-0358.

[23] S. Yang, X. Zheng, J. Li, J. Xu, X. Zhang, and E. C. H. Ngai, "Self-supervised adaptation method to concept drift for network intrusion detection," *IEEE Transactions on Dependable and Secure Computing*, 2025, doi: 10.1109/TDSC.2025.3599321.

[24] X. Hu, D. Ma, W. Wang, and F. Liu, "Dual adaptive windows toward concept-drift in online network intrusion detection," in *International Conference on Computational Science*, pp. 210–224, 2025, doi: 10.1007/978-3-031-97626-1_15.

[25] J. Ying, T. Zhu, A. Zheng, T. Chen, M. Lv, and Y. Chen, "METANOIA: A lifelong intrusion detection and investigation system for mitigating concept drift," *arXiv preprint arXiv:2501.00438*, 2025, doi: 10.48550/arXiv.2501.00438.

[26] S. K. Amalapuram, B. R. Tamma, and S. Channappayya, "SPIDER: A semi-supervised continual learning-based network intrusion detection system," in *IEEE INFOCOM 2024*, 2024, doi: 10.1109/INFOCOM52122.2024.10621428.

[27] J. Talpini, F. Sartori, and M. Savi, "Hierarchical multiclass continual learning for network intrusion detection," in *IEEE NetSoft 2024*, pp. 263–267, 2024, doi: 10.1109/NETSOFT60951.2024.10588909.

[28] C. Guo, X. Li, J. Cheng, S. Yang, and H. Gong, "Continual learning for intrusion detection under evolving network threats," *Future Internet*, vol. 17, no. 10, Art. 456, 2025, doi: 10.3390/fi17100456.

[29] Z. Yin, H. Chen, H. Ma, T. Hu, and L. Bai, "CAEAID: An incremental contrast learning-based intrusion detection framework for IoT networks," *Computer Networks*, vol. 262, Art. 111161, 2025, doi: 10.1016/j.comnet.2025.111161.

[30] "Incremental hybrid intrusion detection for 6LoWPAN," *Computers & Security*, vol. 135, Art. 103447, 2023, doi: 10.1016/j.cose.2023.103447.

[31] S. Meganathan, A. Sumathi, S. S. M. Shah, and R. Rajakumar, "Real-time intrusion detection using adaptive sliding windows for concept drift," *International Journal of Electronics and Communication Engineering*, vol. 12, no. 8, pp. 123–137, 2025, doi: 10.14445/23488549/IJECE-V12I8P111.

[32] E. García-Huitzitl, L. Bustio-Martínez, and R. Cumplido, "Adaptive intrusion detection system: Hybrid K-means and random forest approach with concept drift detection," *Computación y Sistemas*, vol. 29, no. 1, 2025, doi: 10.13053/cys-29-1-5528.

[33] A. Yagik and N. Tiwari, "Unveiling drift in machine learning models integrating with cybersecurity," *International Journal of Applied Mathematics*, vol. 38, no. 10S, 2025, doi: 10.12732/ijam.v38i10s.1089.

[34] M. A. Shyaa et al., "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*, vol. 137, Art. 109143, 2024, doi: 10.1016/j.engappai.2024.109143.

[35] A. Momand, S. U. Jan, and N. Ramzan, "A systematic and comprehensive survey of recent advances in intrusion detection systems using machine learning: Deep learning, datasets, and attack taxonomy," *Journal of Sensors*, 2023, Art. 6048087, doi: 10.1155/2023/6048087.

[36] V. G. da Silva Ruffo et al., "Anomaly and intrusion detection using deep learning for software-defined networks: A survey," *Expert Systems with Applications*, vol. 256, Art. 124982, 2024, doi: 10.1016/j.eswa.2024.124982.

[37] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure," *Sensors*, vol. 23, no. 5, Art. 2415, 2023, doi: 10.3390/s23052415.

[38] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, Art. 102767, 2020, doi: 10.1016/j.jnca.2020.102767.

[39] B. A. Manjunatha, K. A. Shastry, E. Naresh, et al., "A network intrusion detection framework on sparse deep denoising auto-encoder for dimensionality reduction," *Soft Computing*, vol. 28, pp. 4503–4517, 2024, doi: 10.1007/s00500-023-09408-x.

[40] B. Thiyam and S. Dey, "An improved deep autoencoder-based network intrusion detection system with enhanced performance," *International Journal of Internet Technology and Secured Transactions*, pp. 270–290, 2024, doi: 10.1504/IJITST.2024.136658.

[41] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[42] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.

[43] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, pp. 357–370, 2022, doi: 10.1007/s11036-021-01843-0.

[44] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *2015 Military Communications and Information Systems Conference*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[45] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. P. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[46] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[47] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[48] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[49] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, Art. 5941, 2023, doi: 10.3390/s23135941.

[50] S. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991, doi: 10.1109/18.61115.

[51] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951, doi: 10.1214/aoms/1177729694.

[52] V. M. Panaretos and Y. Zemel, "Statistical aspects of Wasserstein distances," *Annual Review of Statistics and Its Application*, vol. 6, pp. 405–431, 2019, doi: 10.1146/annurev-statistics-030718-104938.

[53] C. Villani, *Optimal Transport: Old and New*. Springer, Berlin, 2009, doi: 10.1007/978-3-540-71050-9.

[54] F. J. Massey Jr., "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951, doi: 10.1080/01621459.1951.10500769.

[55] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society: Series B*, vol. 57, no. 1, pp. 289–300, 1995, doi: 10.1111/j.2517-6161.1995.tb02031.x.

[56] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[57] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of ICML*, pp. 1096–1103, 2008.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015, doi: 10.48550/arXiv.1412.6980.

[59] N. Gunasekara, B. Pfahringer, H. M. Gomes, and A. Bifet, "Survey on online streaming continual learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6628–6637, 2023, doi: 10.24963/ijcai.2023/743.

[60] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *arXiv preprint arXiv:2302.00487*, 2023.

[61] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[62] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006, doi: 10.1007/s10618-006-8326-9.

[63] A. De Paola, S. Drago, P. Ferraro, and G. Lo Re, "HOIDS: Concept drift aware hybrid online intrusion detection system," *Journal of Network and Computer Applications*, vol. 254, Art. 104556, 2026, doi: 10.1016/j.jnca.2026.104556.

Authors and Affiliations

Rohit Kumar Thakur[1,*], **Raj Kumari**[1]
[1]University Institute of Engineering and Technology, Panjab University, Chandigarh, India
*Corresponding author: thakur.rohit1997@gmail.com
Email: rajkumari@pu.ac.in

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ASD.zip