International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Unveiling Drift in Machine Learning Models Integrating with Cybersecurity

Anjali Yagik[1] and Namita Tiwari[2]
[1]Research Scholar, [2]Associate Professor
Department of Computer Application, UIET, CSJM University Kanpur
Email: Anjaliyagik8@gmail.com, Namitatiwari@csjmu.ac.in

Abstract

Machine Learning models are often trained on historical data with the assumption that the future data will follow similar pattern. However, in real world applications data shifts over time due to factors such as change in market conditions and user behaviour, resulting in drift. Drift leads to the degradation of model performance, making its management critical for reliable predictions in dynamic environments. This study offers a thorough examination of types of drift and existing detection techniques, including statistical and machine learning based approach. Furthermore, the research will examine the significance drift in cybersecurity, where adaptive strategies introduce both data and concept drift. The aim of this paper is to bridge the gap between machine learning theories with practical security applications.

Keywords: Machine learning (ML), data streaming, drift, concept drift, cybersecurity, Intrusion Detection System (IDS), model drift.

Significance: This review paper is organized into six sections, each addressing a distinct dimension of drift in machine learning. To begin with, *Section 1* introduces terminology, review methodology and streaming data, where information must be processed in real time, and explains drift as the gradual deterioration of model performance. Moreover, it outlines the major types of drift, including concept drift, covariate drift, prior probability shift and many more. Subsequently, *Section 2* elaborates on the key concepts of drift, its impact on model accuracy, and various methods of classification. In addition, it examines drift segmentation, localization, and strategies for identifying specific instances in dynamic environments. Furthermore, *Section 3* reviews established drift detection approaches, such as statistical methods, window-based techniques, and ensemble learning. Collectively, these approaches provide a systematic understanding of effective drift identification in evolving data streams. In continuation, *Section 4* highlights application areas where drift significantly influences ML models. For instance, it discusses challenges faced in finance, healthcare, e-commerce, and cybersecurity. Following this, *Section 5* focuses on drift in cybersecurity, particularly its relationship with intrusion detection systems (IDS). It emphasizes how changing attack patterns degrade IDS performance and demonstrates the importance of adaptive algorithms in addressing evolving threats. Finally, *Section 6* outlines the future scope of research in drift management. Specifically, it underscores advancements in adaptive algorithms and stresses the necessity of domain-specific knowledge for effective solutions.

1.Introduction

The digital world produces data that grows quickly from a variety of uses. As technology advances, the world is evolving. Data processing in machine learning is contingent upon certain

Received: August 05, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

features that are determined by the availability and accessibility of the data. Data streams are distinct from other types of data in that they are characterized by the continuous and sequential arrival of instances [33]. Data streams are distinct from other types of data because of their particular qualities. The term *batch data* is frequently used to describe the data in typical machine learning scenarios. In other words, all of the data is kept in memory and is instantly accessible in its entirety. This differs strongly with stream mining, in which data streams produce items in a sequential, continuous fashion that may also be temporary. This implies that the availability of stream data can be limited [17]. A data stream is any collection of data in a series that has a timestamp. It is not known how much data in the series is still there as the data stream is infinite [11]. Depending on the state of their key distribution, data streams are categorized as either dynamic or static (sometimes called stationary). In contrast to static data, a dynamic data stream's distribution could alter over time. Over time t, feature vectors may undergo changes that cause them to no longer represent the class label. The following characteristics exist in digital data that comes from many sources [7]:

- Data streams are potentially infinite and arrive at varying speeds, making real-time processing challenging.
- Information within streams may change over time, and memory limitations constrain data processing.

Large volume of streaming data are being produced by Business and Government, and in order to assist in making forecasts and choices, they critically require effective data analytics and ML approaches. But the quickly evolving landscape of new goods, markets, and consumer behavior eventually leads to the emergence of the concept drift issue [3]. Machine learning models are especially vulnerable to drift, which occurs when underlying data distributions or decision boundaries change over time, due to the dynamic and hostile nature of cybersecurity. Despite the fact that drift detection in machine learning has been the subject of several research, a thorough grasp of its function and difficulties in cybersecurity situations is still missing. Current surveys often concentrate on either domain-specific applications or algorithmic approaches, but they seldom ever offer a cohesive viewpoint that connects the two. This review seeks to fill that gap by systematically analyzing the literature on drift in machine learning with an emphasis on cybersecurity, highlighting key methodologies, applications, and unresolved issues. The following research questions (RQs) serve as a framework for this investigation:

- What detection methods are commonly employed (statistical, adaptive, hybrid)?
- Which types of drift have been explored in cybersecurity-related ML models?
- How effectively do current approaches address real-time or streaming cybersecurity data?
- What research gaps and future directions can be identified?

These questions establish the scope of the review and provide a structured framework for evaluating existing work and identifying potential opportunities for advancement.

1.1. Background and Definitions

The main categories of drift, related detection techniques, adaption tactics, and real-world limitations are described in the *table 1*, which offers a clear framework for comprehending their function in cybersecurity-centered machine learning.

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 1 Terminology of drift in machine learning models and in cybersecurity contexts

Terms	Meaning	Example
Concept drift	Change in the relationship between input features and output labels *(P(Y|X))*	A spam filter's rules no longer match evolving spam content.
Sudden drift	Abrupt and large-scale change in data distribution	An attacker suddenly switches from password brute-force to phishing attacks
Dataset shift	Change in the overall input data distribution *(P(X))*	Normal network traffic patterns change due to new devices being added
Label/target shift	Change in the distribution of output labels *(P(Y))*	Proportion of "attack" vs. "benign" traffic labels changes over time
Covariate shift	Input distribution changes, but label distribution remains the same	User login times shift from daytime to nighttime without altering outcomes
Gradual drift	Slow, incremental changes in data distribution over time	Malware signatures evolve little by little as new variants emerge
Model drift	Degradation of ML model performance due to outdated training	An IDS trained on old data fails to detect new cyberattacks
Recurring drift	Old patterns disappear and then reoccur later	Certain seasonal phishing scams reappear every year during tax season
Seasonal/periodic drift	Predictable cyclic changes in data distribution	Network traffic changes during weekends vs. weekdays
Supervised detection	Drift detection using labeled data	Comparing predictions against known attack labels
Unsupervised detection	Drift detection without labels, using statistical measures	Monitoring unusual spikes in login attempts
Window-based method	Uses only the most recent subset of data for training/adaptation	Always train on the latest 1,000 log entries
Ensemble methods	Multiple models are combined and updated, weaker ones replaced	Keep several intrusion detection models, drop the weak ones, add new ones
Incremental learning	Continuously updates the model as new data arrives	An IDS that updates itself with each new packet observed

Concept drift is widely recognized across various fields, though it may be referred to by different terminologies [4]:

Table 2 Concept drift terminology across different fields

Field	Terminology
DM (Data mining)	Concept drift
ML (Machine Learning)	Concept drift, Covariate shift
EC (Evolutionary Computation)	Changing Environment
AI and Robotics	Dynamic Environment
Statistics, Time Series	Non-stationary Databases Concept Drift, Load shedding
Information retrieval	Temporal Evolution

1.2. Overview of drift in Machine Learning

In machine learning the word ***Drift*** describes the process by which a machine learning model's performance deteriorates over time. As Heraclitus says that "*The only constant in life is change"*. Drift is a phenomena that causes ML models, which are trained on previous existing data, to become out of date and lose accuracy over time when applied in real world scenarios. There are several reasons why this could occur, including alterations in the input data overtime or shifts in the correlation between the target (X) and the desired variable (Y). Drift can be a significant issue for machine learning in real world scenarios because data is frequently

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

changing. Take a Machine learning model for instance that uses previous data to forecast a company's Stock price. The model might initially perform well if it was trained with data from a steady market. It is possible that the model will lose its predictive accuracy if the market experiences increased volatility overtime. While no machine learning models can totally prevent drift, some are better than others at handling changes in the data. ML researchers are very interested in data drift as a result of COVID-19 pandemic, because machine learning frequently depends on the fundamental premise

$$The\ Past == The\ Future$$

which makes drift a serious problem. This is almost rarely the case in the actual world [2]. The ultimate objective of machine learning algorithm is to finding pattern in historical data and using them to forecast behavior for future occurrences. Often called "concepts" these patterns aid in the classification of data and the identification of correlations between various variables, making them essential to machine learning [5]. For what reason does drift happen? Many factors can lead to drift in ML like as the outside world has changed or transformed and a new context is applied to the model, but two primary categories of causes are often identified: poor training data and shifting environmental conditions. *Periodic drifts* are those that consistently happen at regular intervals of time such as seasonal variations in sales or recurring pattern in network traffic. These drifts follows a regular cycle and can often be anticipated in advance. However, when such changes become inconsistent or unstable, they are referred as *irregular drift* [29].

1.3. Understanding drift in Cybersecurity System

The dynamic nature of environments, particularly in fields such as cybersecurity, reinforces the notion that change is the only constant. This drives the need for robust anomaly-based drift detection methods. Machine learning (ML) has become a powerful instrument for identifying malware, preventing breaches, and detecting anomalies in the quickly changing field of cybersecurity. However, concept drift might cause these ML models' performance to deteriorate with time. For example, 68% of phishing emails that Gmail blocks are unique every day, requiring frequent updates and modifications to Google's security features [37]. Furthermore, 30% of Windows malware that is produced every day can come from unidentified families, necessitating the updating of ML models with behavioral reports in order to identify drifting samples [39]. Additionally, at least 53% of businesses are using Artificial Intelligence in various Cybersecurity domains [38]. In cybersecurity, drift signals a possible assault [12].
Since attackers are always creating adversarial samples to avoid detection, one of the biggest problems is the volatility of the data used to create models. As a result, it becomes necessary to update the models often in order to stay informed of emerging assaults. Since the class labels often show a huge imbalance that is, hundreds of attacks amid thousands of harmless samples applying fully supervised algorithms presents another difficulty. Such cases are difficult to label as doing so necessitates subject expertise and could slow down the learning process; saying differently, the analyst categorizing the data may act as a learning bottleneck. This drives the creation of anomaly detection and semi-supervised techniques [40-44].
As more people use the internet, the likelihood of cyberattacks is also rising. These kinds of attacks are frequently unique, necessitating the use of intelligent systems to identify them. To find any kind of malicious activity, an intrusion detection system (IDS) examines network traffic [48].

1.4. Review Methodology

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To ensure a systematic and comprehensive coverage of existing research, this review adopted a structured methodology. Relevant literature was collected from reputable scholarly databases, including IEEE Xplore, Springer, ScienceDirect, ACM Digital Library, ResearchGate and Scopus. Keyword combinations like "concept drift," "data drift," "machine learning," "cybersecurity," "intrusion detection system," and "adaptive models" were used to create the search searches. The timeframe under consideration was 1990–2025 because of the notable progress made in drift detection techniques and cybersecurity applications during this time. According to their applicability to the two main areas of this review - (i) drift in machine learning and (ii) its consequences in cybersecurity settings, the original pool of papers was culled. Additionally, duplicate records were also eliminated. The chosen works were then grouped according to their main contributions, including drift detection techniques such as statistical, window-based, ensemble approaches, cybersecurity application areas such as intrusion detection, malware analysis, and anomaly detection and identified challenges such as real-time adaptation, data heterogeneity, and adversarial conditions. The review was able to draw attention to important findings, methodological trends, and unresolved research gaps because to this theme grouping.

1.5. Classification of drift in Machine learning

When discussing drift, it is important to distinguish between its primary forms: ***data drift*** reflects changing statistical distribution of data, and ***concept drift*** focuses on relationship between input features and target variables evolve over time.

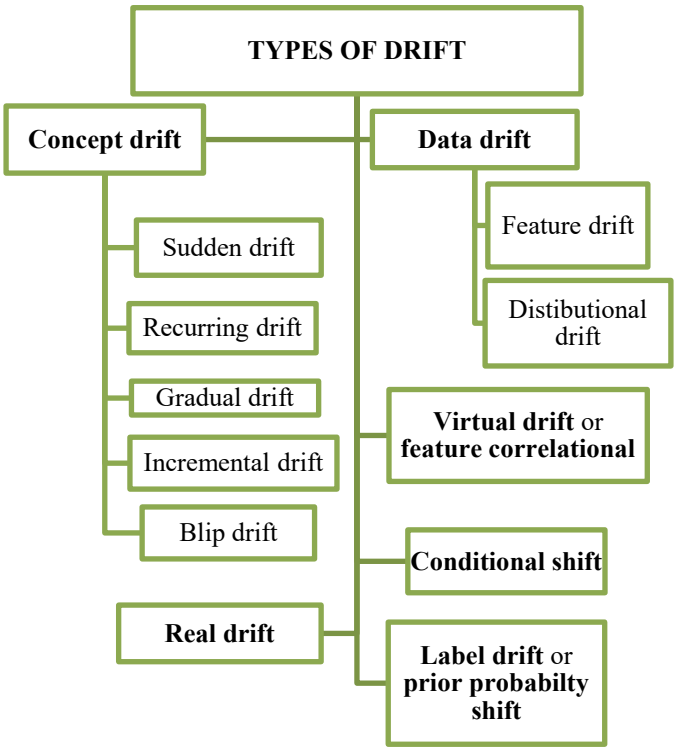

Figure 1 Types of drift in machine learning

1.5.1. Data drift

It is often referred to as *input drift*, *covariate shift* and *population drift*, relates to how the distribution of data has changed overtime. These changes can impact the effectiveness of a ML models which are trained on historical data under the assumptions that the future data will follow the same distribution. The causes of data drift are diverse and may include changes in market condition, environmental changes, evolving user behavior, seasonal variations or regulatory modifications - for instance, increased sales during holidays.

Data drift can also be categorized into different types: Changes in the statistical properties of features - such as shifts in the mean or variance of

Received: August 05, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

individual variables or groups of variables are referred to as *feature drift*. This form of drift alters how the model interprets input data, which can ultimately impact its predictive accuracy. Whereas, *population drift*, occurs when the underlying population generating the data changes, leading to a shift in the overall data distribution, also termed as distributional drift. A common example is demographic shifts that alter the representation of groups within the dataset.

1.5.2. Concept drift

It is often referred as *dataset shift* or *model drift*, occurs when the relationship between its input features and goal variable in machine learning model changes overtime [31]. In machine learning, concept drift describes a situation where the target variable's statistical properties, which the model attempts to predict, evolve over time [29]. Real concept drift erodes the model's performance and modifies decision boundaries. The only thing affected by virtual concept drift is the conditional probability density function. Also referred to as feature or data drift, virtual drift [29].

1.5.3. Additional types of drift

Apart from the commonly discussed types of drift such as data drift and concept drift researchers have also identified several other variations that impact ML models in different ways: ***Label drift*** is the term used to describe variations in the output labels' distribution. This frequently happens when model predictions are impacted by changes in the proportion of the classes in classification issues. It can also termed as *prior probability shift.* Another important form is *feature correlation drift*, also called ***virtual drift***, when there is a shift in the core connections between characteristics but input and output remains unchanged. Even yet, the model's ability to forecast outcomes accurately may still be affected by the way characteristics interact over time. In contrast, changes in the data distribution as well as the fundamental connection between the input characteristics and the goal variable are referred to as ***real drift***. This type of drift is the most complicated as it impacts the entire model, necessitating both drift detection and model adaptation [30]. When the marginal distributions stay the same but the conditional distribution of the output variable varies based on the input characteristics, this is known as ***conditional shift***. Changes in the correlations between the input variables and the target are frequently a part of this kind of drift.

A practical example can be seen in spam E-mail, consider a scenario where a ML model is taught to identify spam E-mail. The model could no longer be able to correctly identify spam if users start receiving different kinds of spam E-mail. If a concept at different times has the same meaning, there is no concept drift [6]. Thus, these additional types of drift such as label, virtual, real, and conditional highlight the complexity of dynamic data environments. Each presents unique challenges, reinforcing the need for continuous monitoring and adaptation of ML models to maintain reliability.

1.6. Static vs Dynamic modelling in presence of drift

ML model performance require constant monitoring and adjustment, which presents a problem in data analysis and machine learning. To illustrate this, consider a binary classification issue in which a binary goal variable, A (0 or 1), and a set of characteristics B are both indicated. Learning a classifier f (A) that predicts the target variable B is the aim [9].

- **Static model-** The relationship between A and B is constant in a static setting (no concept drift), and you may utilize a fixed model: $\boldsymbol{B = f_{static}(A)}$

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Concept drift-** When concept drift is present, A & B's connection evolves over time. To depict this, we may introduce a time-dependent function f_t (A) (where t stands for time) that encapsulates the evolving idea, the function f_t may alter over time [13] as a result of changing data distributions or other reasons: $\boldsymbol{B = f_t A}$.

Understanding the distinction between static and dynamic modeling is critical, as data in practice rarely remains unchanged. In use cases such as spam filtering, fraud detection, and cybersecurity, adaptive models are needed to respond to evolving patterns and ensure consistent accuracy.

2. Concept drift

As highlighted earlier. It is often referred to as dataset shift or model drift, occurs when the relationship between inputs features and goal variable in ML model changes overtime, this changing nature of data distributions is known as *concept drift* [29,31]. In several fields, such as robotics, sensors, system monitoring, and anomaly detection, the issue of concept drift has received a lot of attention recently among researchers [1]. The two main components of traditional machine learning are prediction and training. Nevertheless, three new dimensions have been brought to the field of ML with concept drift research: the detection of concept drift (*drift detection*); the comprehension of the drift (*drift understanding*); and the adaption of the drift (*drift adaptation*). Recently, research has concentrated on determining how to identify concept drift, comprehend it, and adjust relevant knowledge accordingly, so that prediction and decision-making can be flexible in concept drift-affected environments.
Retrieving knowledge regarding "When" (the time and duration of the concept drift), "How" (the degree and severity of the concept drift), and "Where" (the drift areas of the concept drift) is referred to as drift understanding. The drift detection methods produce this status data, which is then utilized as an input for drift adaptation [3]. As discussed, the distinction between static modeling, where relations in data remains fixed, and dynamic modeling, where the relationship in data keeps changing, is fundamental, since real-world data is rarely stable.

2.1. Classification of Concept drift

In different contexts, concept drift can be classified into distinct types, each with unique characteristics. *Sudden drift,* also referred to as *abrupt drift,* occurs when there is a sudden change in data distribution at a certain point of time with no time overlap. For example, in fraud detection, a new type of fraud could suddenly appear, making previous detection patterns obsolete.
Formally, this can be expressed as the generating process of the data stream shifting between two successive instances or batches, i.e., $C_{t_i} \neq C_{t_{i-1}}$. In contrast, when there is a slowly change in data distribution with time overlap it is termed as *gradual drift*. Gradual drift is especially relevant in industries where data evolves slowly, like customer preferences in marketing or network traffic patterns in cybersecurity [29]. In such cases, the new concept C_{t_i} coexists with the previous concept $C_{t_{i-1}}$ before fully replacing it. Another important form of drift is *recurring drift,* which occurs when the same concept is perceived after an extended period of time. It has dual nature, incorporating both cyclic and acyclic properties. Recurring drift is particularly important in domains like retail, finance, or energy consumption, where patterns

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

often follow cycles. For instance, during the summer, sales of cold items typically increase [28]. A collection of states' $S = \{S_1, S_2, S_3 \ldots S_n\}$ exists in the data stream, and each state S_i is produced by a distinct generating process. When they appear as a new drift S_n, C_i, and transition to the process (e.g., S_2) are regarded as recurrent drift. Formally, this is stated as $S_{i+1} = S_{i-k}$, where k is the k^{th} iteration of the generating process. A closely related phenomenon is *incremental drift*, in which an old thought gradually transforms into a new one over a period of time. Incremental drift is similar to gradual drift, and the two terms are often used interchangeably [26]. While gradual drift refers to a shift from one distribution state to another with an overlap period, incremental drift involves a continuous, slow change in the data distribution over time. Finally, *blip drift* represents a different case. The term "blip" refers to an outlier in a stationary distribution (probability distribution in a stochastic process that doesn't change over time) and denotes a quick or sudden change, or unusual event. Thus in general it is not considered as drift [7]. Blip drift is often encountered in scenarios with sudden but temporary anomalies, such as traffic spikes during promotions, momentary system glitches, or rare events that do not reflect the general trend in the data. Handling it correctly ensures that models do not overfit to these short-term shifts [27].

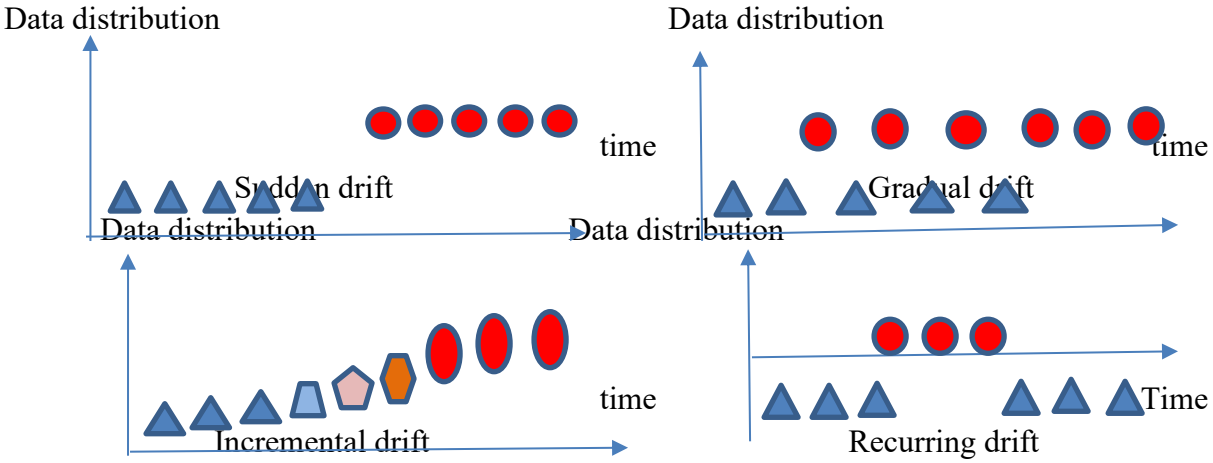

Fig.2 Types of concept drift in ML including sudden, gradual, incremental and recurring drift

2.2. Effects of Concept Drift in ML

Different types of concept drift manifest with unique characteristics and require tailored approaches for detection and adaptation.

Table 3 Effects of drift in machine learning

Types of Drift	Key characteristics	Approaches for handling concept drift	Affects in ML
Sudden drift	Immediate change Significant impact	Concept drift detection algorithms (DDM or ADWIN) Retraining and Ensemble methods	Performance degradation Need for quick adaption
Gradual drift	Slow transition Incremental impact on model performance Occurs in many real world scenarios More difficult to detect	Windowing methods Incremental learning Cumulative drift detection algorithm	Subtle degradation Continual adaption
Incremental drift	Gradual and progressive change	Incremental learning Sliding window approaches	Gradual decline in model performance

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

	Cumulative impact Occurs in slowly evolving environments Hard to detect	Adaptive drift detection algorithm	Requires frequent or adaptive updates
Recurring drift	Cyclic nature Periodic impact on model performance Common in seasonal or temporal data Model retraining challenges	Ensemble methods Drift detection with time awareness Periodic retraining	Fluctuating accuracy Need for memory based or adaptive models
Blip drift	Short duration Caused by outlier Minimal long term impact Potential to cause False positive	Threshold based detection Ensemble methods Buffering or smoothing techniques	Temporary performance degradation Risk of overreaction

2.3. Concept drift's spatial characteristics

In order to extract pertinent spatial drift parameters, drift localization and drift segmentation are used. Drift segmentation is the process of breaking the dataspace up into areas or segments having homogenous drift behavior. Drift localization aims to pinpoint the specific features, instances or time periods where the data distribution has shifted, helping in understanding and adapting to the change. Drift segmentation does not require drift detection as a prepossessing and may easily handle continuous time. Moreover, drift segmentation may contain more information and is more fine-grained than drift localization [14].

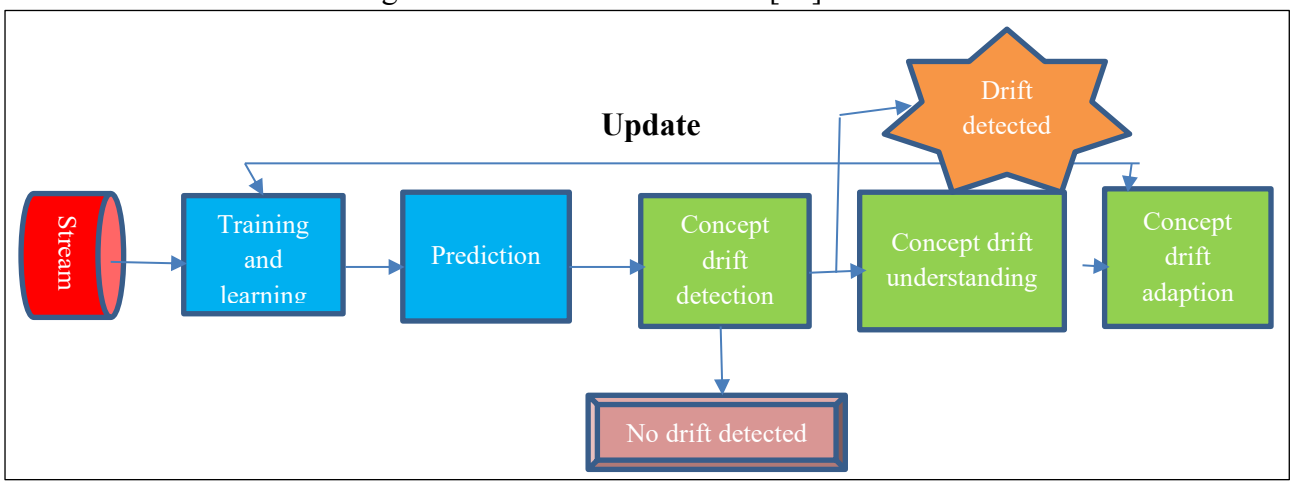

Fig.3 the process of drift detection involves multiple stages from streaming data to drift adaption

3. Drift detection

The methods and procedures used to identify points of change or change time intervals in order to quantify and describe concept drift are referred to as drift detection. Four steps make up a generic drift detection framework:

- Data retrieval or Information Extraction seeks to retrieve data chunks and is efficient at structuring such chunks to create knowledge or a meaningful pattern

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Data modeling seeks to abstract the recovered data and identify the salient characteristics that hold sensitive information, i.e., the data aspects that, should they drift, would most affect a system
- Test statistics calculation is the distance estimation or assessment of dissimilarity
- Hypothesis test determines the p-value, or statistical significance of the change seen in test statistics computation, using a certain hypothesis test

3.1. Statistical Method

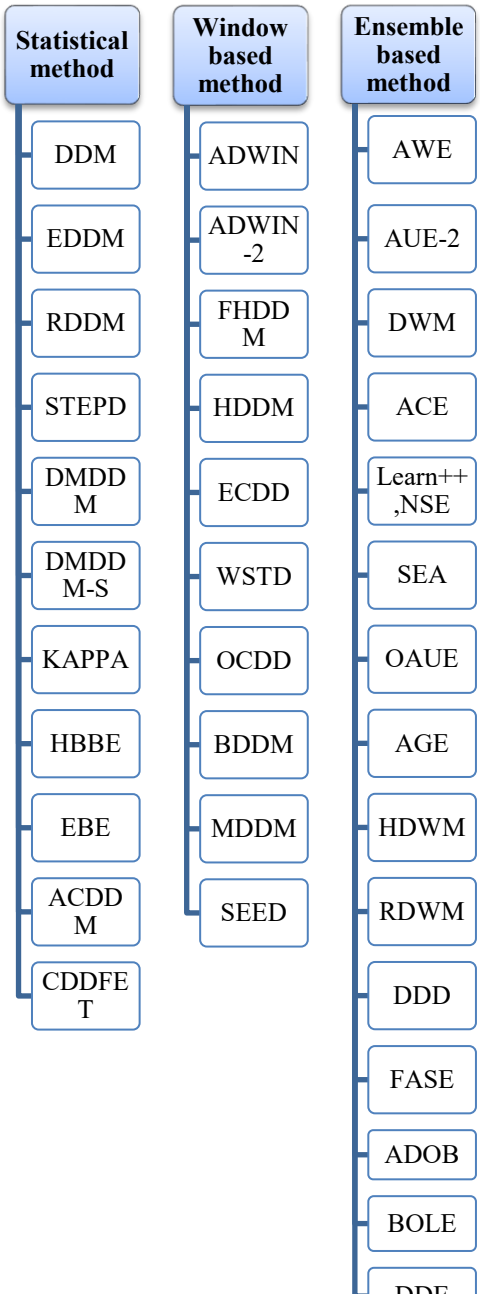

Fig. 4 Different Drift detection methods

•DDM (Drift Detection Method) keeps track of the classifier's error rate. This technique manages the mistakes generated by a learning model during prediction by statistically comparing two windows. All of the data is contained in one window, and the only data in the second window is from the beginning of the stream until the time at which the prediction model's error rate starts to rise. Only statistical data and recent mistakes are retained in memory; the windows are not. More or less DDM is memoryless [18]. When drifts happen, algorithms in this discipline usually alter models completely or in part when obvious alterations are found [8]. Supervised and unsupervised contexts are the two main categories into which the Drift Detection Method (DDM) fall. When the sample and comparison data are labeled, supervised environments are able to identify drift. In unsupervised contexts, detection techniques are dependent on the availability of data. They fall into two categories: batch techniques, which determine the batch size based on the available data, and online techniques, which take into account all incoming data at a given moment. There are many more subcategories of batch techniques, including ones that use partial batches vs complete batches. One further division of online techniques is to distinguish between sliding reference methods for data with fewer labels and fixed reference methods based on reference windows [24].

- EDDM (Early Drift Detection Method) is a variant of DDM that employs the distance error rate instead of the classifier's error rate, the gap between two misclassification errors will increase as predictions get better. The process for resizing windows is the same as with DDM. The primary drawback with EDDM is that it requires at least 30 mistakes to calculate, which makes it difficult to utilize with datasets that are imbalanced [20].

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- RDDM (Reactive Drift Detection Method) regularly updates the statistics used for drift detection and discarding older cases. Saying differently, RDDM avoids possible memory overflows by discarding previous instances of drifts that arrive very slowly [19].
- STEPD (Statistical Test of Equal Proportions) is similar to DDM and EDDM, tracks a base classifier's predictions for drift detection. STEPD compares the classifier's output using two windows. The 1st window is a "recent" window, and a parameter with a default value of 30 occurrences determines the size. The 2nd window is the “older” window which comprises all instances observed since the last reported drift [21].
- The disagreement measure and the Page-Hinkley test are combined in DMDDM (Diversity Measure as a new Drift Detection Method) to swiftly and with less memory and time consumption identify concept drift.
- To identify abrupt drifts in the absence of class labels, DMDDM-S (Diversity Measure as a Novel Drift Detection Method for Streaming Data without Class Labels) tracks the diversity of a pair of classifiers instead of error estimates.
- KAPPA minimizes computational resources by measuring the degree of agreement between several classifiers in order to identify concept drifts.
- HBBE (Hybrid Block-Based Ensemble) is designed to handle various sorts of drifts in multi-class classification in changing data streams. It combines the advantages of an online drift detector for a k-class issue with the concept of block-based weighting.
- The EBE (Entropy Based Ensemble) incorporates entropy as a drift detector into the developing ensemble to enhance performance by utilizing information entropy to identify concept drifts.
- In order to identify concept drift in dynamic data streams, ACDDM (Accurate Concept Drift Detection Method) analyzes prequential error rate consistency using Hoeffding's inequality.
- Three modified versions of CDDFET (Concept Drift Detection based on Fisher's Exact Test), a modification of STEPD, are offered for drift detection. The p-value is calculated using Fisher's exact test.

3.2. Window Based Method

- The ADWIN (Adaptive Sliding Window) method detects drift by comparing the means of two sub-windows w_0, w_1 of a larger window w. When the mean of the two sub-windows diverge significantly, ADWIN detects this and eliminates the window's tail until the mean return to normal.
- An improvement to ADWIN called ADWIN2 makes use of a variable-sized window to adjust to variations in the data distribution. This detector keeps a sliding window that is dynamically adjusted and split into two sub-windows that reflect the old and new data. If the mean difference between the two sub-windows is greater than a certain threshold, ADWIN2 flags drift [34]. When drift is present, the window size shrinks, and when stability is present, it grows. It is necessary to set up a different instance of ADWIN2 with a lower threshold to identify alerts [35].
- Hoeffding's inequality and a window of size n are used by FHDDM (Fast Hoeffding’s Drift Detection Method) to identify drift by comparing the present probability to the maximum level of accurate forecasts.

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Based on a confidence level (usually selected by the user) and the standard deviation of the error rate, ECDD (Exponential Cumulative Drift Detection) determines a threshold. A drift may have happened, in which case the model would need to adjust to the new distribution, as shown by the error rate exceeding this threshold.
- The EWMA approach is modified by ECDD (EWMA for Concept Drift Detection) to identify drift by use weights to differentiate between more current and older occurrences.
- The accuracy of a learner's performance is assessed using two windows using STEPD (Statistical Test of Equal Proportions), which warns if there is a significant change in accuracy within the most recent window.
- The implicit concept drift detector known as OCDD (One Class Drift Detector) approximates the distribution of a new concept using a sliding window method. The fraction of outliers is estimated in order to identify drift inside the sliding frame.
- The Bhattacharyya distance is used by BDDM (Bhattacharyya Distance Drift Detector) to detect shifts in distribution by tracking changes in the mean and variance over time.
- By tracking the prediction outcomes over a size-n window, MDDMs (McDiarmid's Drift Detection Methods) use McDiarmid's inequality to identify drift. A 1 is added to the window if a prediction comes true; if not, a 0 is added.

3.3. Ensemble Based Learning

- AWE (Accuracy Weighted Ensemble) keeps a fixed-size ensemble of models in order to manage streaming data. It assesses each model's accuracy using the most current data, and models with better accuracy have a bigger impact on the predictions made by the ensemble.
- By making the weighting process more adaptable, AUE2 (Adaptive Ensemble 2) improves AWE. By combining online learning and weighting modifications to effectively manage idea drift in data streams, it overcomes the drawbacks of AWE.
- DWM (Dynamic Weighted Majority) is a more flexible approach made especially to identify and adjust to notion drift. In reaction to variations in the models' performance on recent data, it modifies the ensemble's composition as well as the weights of individual models.
- The goal of the ACE (Adaptive Classifier Ensemble) technique is to adaptively maintain a classifier ensemble that can adapt to idea drift. In order to keep the ensemble current as the underlying idea evolves, it constantly updates it by eliminating underperforming classifiers and adding new ones. It is effective in both gradual and sudden drift scenario.
- By gradually learning from new data while keeping information from earlier data, Learn++ is an ensemble-based incremental learning algorithm created to address concept drift. Additionally, it may be specially modified for non-stationary situations, like Learn++ Non-Stationary Environment, or NSE.
- The block-based ensemble learning technique known as the SEA (Streaming Ensemble technique). A fixed size ensemble is added to the individual classifiers that are built from examples read in consecutive blocks or chunks. The classifier with the lowest performance gets eliminated from the ensemble completely if it is full [23].
- An extension of conventional ensemble techniques, OAUE (Online Accuracy Updated Ensemble) dynamically updates classifier weights according to their performance on the most recent data, therefore adapting to concept drift. It is intended for use in online learning settings where a stream of data is received and the model must swiftly adjust without having to be retrained from the beginning.
- AGE (Adaptive Greedy Ensemble) is an ensemble technique that adaptively chooses the top-performing classifiers in an ensemble using a greedy strategy. Its main goal is to control idea

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

drift by optimizing the ensemble's composition such that only the best classifiers are kept and put to use.

- The Dynamic Weighted Majority (DWM) algorithm is extended by HDWM (Hierarchical Dynamic Weighted Majority), which adds a hierarchical framework to better manage complicated data streams with concept drift. The ensemble is arranged into layers using the hierarchical technique, and each layer adjusts to various drift or data characteristics.
- Another DWM variant that adds recursion to increase the ensemble's flexibility is called RDWM (Recursive Dynamic Weighted Majority). The goal of RDWM is to keep an ensemble that can more dynamically adjust to drift by recursively updating its structure in response to variations in the data distribution.
- The foundation of DDD (Diverse Dynamic Ensemble for Drift Detection) is the notion that a varied group of classifiers can more effectively identify and adjust to concept drift. By ensuring that multiple models capture distinct elements of the data, classifier diversity increases the ensemble's resilience to changes in the data distribution. Additionally, the technique dynamically modifies the ensemble in response to drift detection.
- Fast switching between base classifiers in response to concept drift is the main goal of the ensemble approach known as FASE (Fast Adaptive Switching Ensemble). Maintaining numerous models trained on various data distributions and rapidly switching between them when drift is observed is the fundamental notion. It has the ability to handle sudden drift because of fast switching.
- ADOB (Adaptive Online Bagging) is a variation of the online bagging algorithm that responds to idea drift by modifying the weights of base classifiers. By adding an adaptive mechanism that concentrates on recent data to preserve accuracy in drifting situations, it alters conventional bagging.
- In order to adjust to idea drift, the BOLE (Batch Online Learning Ensemble) ensemble approach blends online and batch learning. To ensure that the ensemble can react to drift while preserving long-term performance, it seeks to achieve a compromise between the stability of batch learning and the flexibility of online learning. Best suited for gradual drift with adaptive weighting.
- DDE (Dynamic Drifting Ensemble) is an ensemble technique that continually modifies the ensemble's weights and structure in response to the available data, allowing it to dynamically adapt to idea drift. Its fundamental goal is to keep a dynamic ensemble that can react to drift without requiring specific drift detection methods. It adapts to both gradual and sudden drift through continuous updates.

4. Application areas

Different application domains experiences drift types due to evolving condition, changing patterns or external influences. Below are some key fields and the types of drift they encounter:

Table 4 Fields of drift across various domains

Area or field	Reason	Types of drift
Finance and Banking	Fraud detection, Financial conditions and investor sentiment changes	Concept drift, Distribution drift
(NLP) Natural Language Processing and Sentiment Analysis	Language usage, slang and terms change overtime	Semantic drift
Healthcare	Patient demographics, treatment protocol or diseases trends changes	Data drift
Real Estate	Market trend prediction, property evaluation, demand forecasting	Neighborhood drift

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Supply Chain Logistics	fluctuations in demand, fuel costs, or geopolitical events	Distribution drift
Cybersecurity	Threat landscape evolve with new malware, phishing technique and attack vectors	Concept drift
Recommendation System	Changing user preferences, seasonal or temporal changes, evolving trends, dynamic user behavior.	Data drift
Human resource and Workforce Analytics	Employee behavior, talent acquisition and recruitment	Data, covariate drift
Telecommunications	Network traffic, user behavior, and device types	Distribution drift
Agriculture	Pest and diseases management, crop yield prediction, soil health monitoring.	Data drift
Weather forecasting	Climate change and seasonal variations	Environmental drift
Energy Sector	Renewable energy management, smart grid and IOT and energy consumption pattern	Concept drift and Covariate drift
Advertising	Customer behavior, trends in ad interaction	Covariate drift
Marketing	Personalized recommendation, customer lifetime value, pricing strategies, market trend analysis	Concept drift
Manufacturing and Industry	Machine wear and tear, sensor aging, and changes in operating conditions	Input feature drift
Retail and Ecommerce	Market trends evolve, purchasing behavior and preferences of customer changes	Feature drift
Autonomous system	Environmental conditions, traffic patterns, object appearances	Covariate, label and sensor drift
Social media analytics	Trends, hashtags and topics changes rapidly	Temporal drift

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

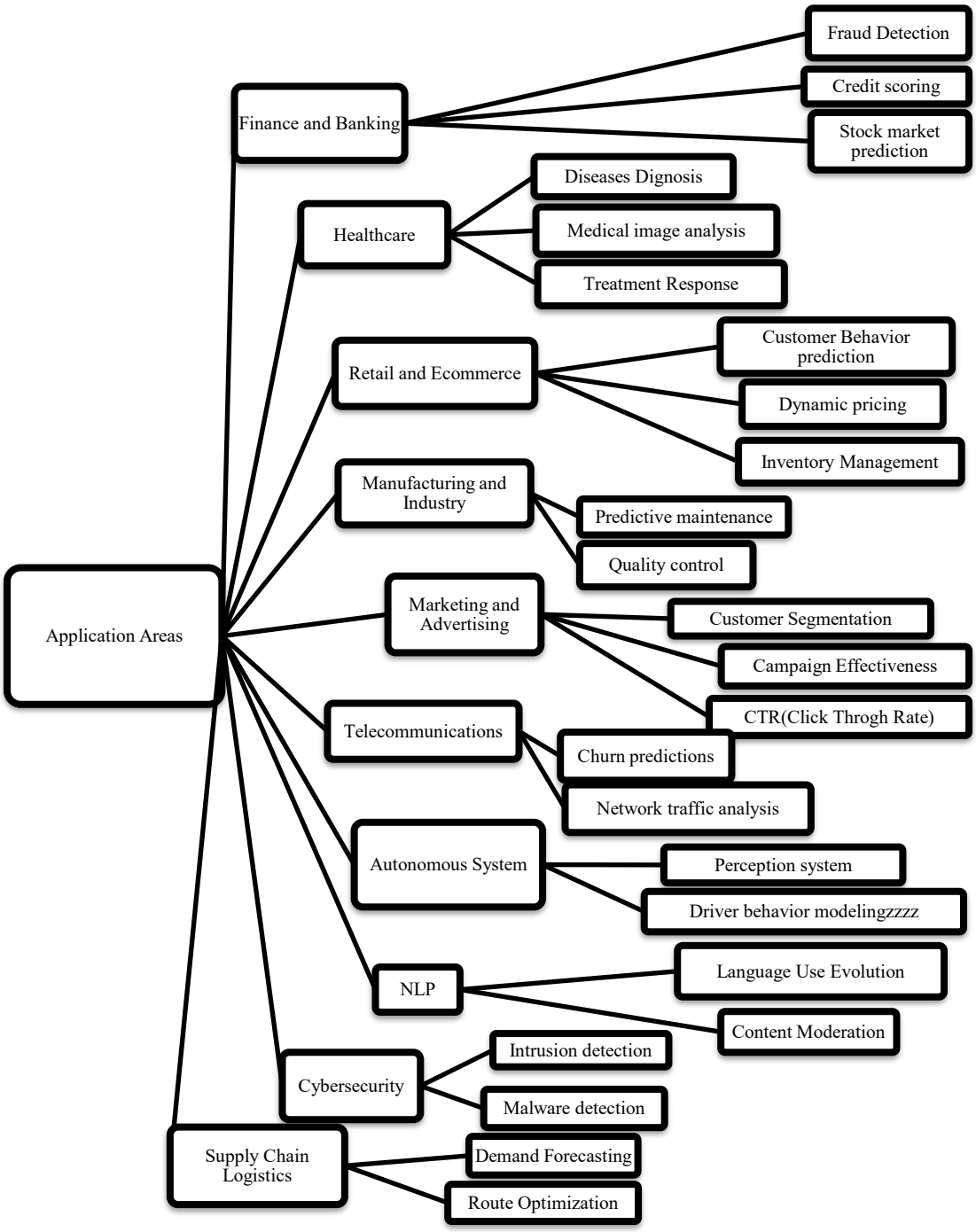

Fig. 5 Application Areas of drift

5. From Machine Learning Drift to Cybersecurity Practice

As mentioned in the above section, machine learning has become a powerful tool for detecting malware, preventing breaches, and identifying anomalies in the rapidly evolving field of cybersecurity. However, concept drift can cause the performance of these ML models to degrade over time. Concept drift directly impacts cybersecurity systems, especially IDS, where evolving attack patterns and changing network behaviors can reduce detection accuracy.

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Effective drift detection ensures IDS remain adaptive, reducing false alarms and missed threats while improving resilience against emerging attacks.

5.1. IDS (Intrusion Detection System)

IDS keep an eye out for attacks on computer networks [47]. When a software program is used to monitor network traffic and computer activities in order to find unexpected events, the technique is known as intrusion detection or Intrusion Detection System [15]. IDS are often separated into two primary categories: Anomaly-based IDS and Misuse-based IDS. Misuse-based detection looks for previous attacks and compares the current traffic to them; if a similarity is found, an alarm is triggered [49]. Conversely, anomaly-based detection looks for any deviation from typical behavior in the new traffic and labels it as abnormal. Identifying zero-based attacks requires anomaly-based detection [48]. IDS is a kind of network security that has the ability to recognize and detect threats before data is lost, services are interrupted, or unauthorized access is allowed. In addition, IDS can offer a graphical user interface (GUI) where users can engage and access different functions during the IDS training and testing procedure [16].

There are two types of intrusion detection systems: a host-based system (HIDS) and a network-based system (NIDS). HIDS looks at events on a host computer, for example, identifying illegal modifications to important system files and NIDS looks at packets collected by network devices like router, For instance, identifying abnormally high data transfer rates that can point to a DoS (Denial-of-Service) attack. A method that incorporates the best aspects of both worlds is called hybrid detection.

In intrusion detection systems, static machine and deep learning techniques are frequently employed. However, the changing data distribution and the obsolescence of the static data sources utilized for model training limit their efficacy [46].

Its main objective is to identify possible intrusions by examining user activity, system behavior, or traffic patterns. These intrusions might include illegal access, assaults, or abnormalities.

5.2. Illustrative Applications of ML in Cybersecurity

The capacity of ML to instantly adjust to complex patterns and changing threats has made it a crucial component of cybersecurity. The table given below represents several key areas:

Table 5 ML applications in Cybersecurity

Security tasks	Specialized tasks	Machine learning tasks
Attack Detection	Malware detection Intrusion detection Spam filtering	Classification Outlier Detection Classification
Incident Response	Malware labeling Log analysis	Clustering
Security Analysis	Malware analysis Risk assessment	Reinforcement Learning/ranking Regression
System Forensics	Attribution Object recognition	Clustering Dimensionality Reduction

5.3. Relation between IDS and Concept Drift

Intrusion Detection System is a hardware or software program that keeps an eye out for hostile activities and rules violations on a network or system. To put it differently, a system or network is monitored by an IDS to look for malicious activities and policy breaches [10]. IDS, especially anomaly-based systems, must adjust to changing network traffic patterns in the context of concept drift in order to successfully identify new threats and prevent false positives.

International Journal of Applied Mathematics
Volume 38 No. 10s , 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Key components of cybersecurity are intrusion detection systems, which are responsible for spotting unusual or illegal activity in networks and systems. These systems strive to accurately detect malware, attack, or unauthorized data exfiltration.
The ADLC (Attack Development Life Cycle) is an essential technique for explaining how cyberattacks are carried out. Having a thorough understanding of attacker tactics is crucial to creating strong defenses. Here are some strategies to tackle with concept drift in machine learning for cybersecurity.

- Retraining the model on new data is crucial for capturing changing attack patterns; for example, retraining the model every day or every week using the most recent network traffic data can greatly improve its accuracy. An alternate strategy is incremental learning, which enables models to update themselves over time without requiring total retraining.
- By utilizing many models that capture different features of the data, ensemble approaches improve resilience.
- One useful tactic for controlling concept drift in machine learning models is adaptive windowing. One method is to retrain the model using sliding window approaches, which keep a fixed-size window of recent data. Dynamic windowing is an additional strategy that modifies the training window's size based on the concept drift's rate.
- In cybersecurity, anomaly detection models are crucial because they move the emphasis from recognizing known attackers' static signs to spotting odd patterns of activity. Organizations can efficiently identify zero-day or previously undetected assaults and adjust to changing attack vectors by training models to detect abnormalities. Active learning- This approach increases the model's capacity to respond to emerging risks while lowering the quantity of labeled data required.

IDS have unique barriers from each kind of drift, and in order to remain successful, they must adapt to shifting data distributions [10].

Table 6 Kinds of drift and their relationship with IDS

Kinds of Drift	Example	Description
Signature based drift	Protocol mutation, Signature variations	Static IDS models are unable to identify new or updated protocols when older protocols phased out
Anomaly based drift	User behavior	Network traffic patterns may change significantly if regular user behavior is altered, for as by working remotely.
Hybrid models	Mixed strategies	IDS that use both signature-based and anomaly-based techniques deal with several levels of idea drift.
Application specific	IOT devices	Older IDS models become outdated as IOT devices grow more prevalent and alter network interactions.
Geopolitical changes	State sponsored attacks	Concept drift brought about by new types of APTs as a result of geopolitical shifts.

6.Future scope

In future research, there is a need for more robust adaptive algorithms that can detect and respond to different types of drift early using self-learning systems and online learning that automatically adapts to changes without human intervention. Furthermore, integrating deep learning and reinforcement learning holds potential for improving performance in dynamic environments. Future developments should focus on creating more flexible, scalable, and effective algorithms to deal with the ever-changing threat landscape in Machine learning based

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

drift detection in Cybersecurity. Without periodic retraining, machine learning models must constantly evolve to identify new harmful activities as cyberattacks get ever more sophisticated and stealthy. The development of real-time drift detection systems capable of functioning on high-dimensional, multimodal data streams, as well as the integration of unsupervised and semi-supervised learning approaches to manage scarce or delayed labeled data, are emerging areas of interest. Moreover, resource-efficient drift detection techniques tailored for limited contexts will be crucial as the number of Internet of Things (IoT) sensors increases. Over time, it will also be important to continue researching how to seamlessly incorporate these cutting-edge methods into legacy cybersecurity systems.

7.Conclusion

This review provides a comprehensive examination of the various types of concept drift in machine learning as reported in the literature. Our study systematically analyzes all the aspects of concept drift across multiple ML Models. It further surveys widely used drift detection techniques and evaluates their applicability across diverse application domains, with a particular focus on cybersecurity. The review also highlights the mechanisms and challenges associated with drift in both supervised and unsupervised contexts. Through detailed discussion and critical analysis, we establish the relationship between concept drift and cybersecurity. Additionally, potential future research directions are outlined to encourage innovative approaches. To the best of our knowledge, this review serves as a complete and informative repository for the readers, practitioners and researchers working in this field.

References

[1] Aguiar, Gabriel J., Alberto Cano, 2024, A comprehensive analysis of concept drift locality in data streams, Knowledge-Based Systems, Volume 289, 111535, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2024.111535.

[2] Peter Mardziel, 2021, Drift in machine learning, Towards Data science, https://towardsdatascience.com/drift-in-machine-learning-e49df46803a

[3] Lu, Jie & Liu, Anjin & Dong, Fan & Gu, Feng & Gama, João & Zhang, Guangquan, 2018, Learning under Concept Drift: A Review. IEEE Transactions on Knowledge and Data Engineering. PP. 1-1. 10.1109/TKDE.2018.2876857, https://doi.org/10.48550/arXiv.2004.05785.

[4] Mahdi O.A., Ali N, Pardede E, A. Alazab, T. Al-Quraishi and B. Das, 2024, Roadmap of Concept Drift Adaptation in Data Stream Mining, Years Later, in *IEEE Access*, Vol. 12, pp. 21129-21146, doi: 10.1109/ACCESS.2024.3358817

[5] Superwise, 2022, May 5, *Everything you need to know about drift in machine learning*. https://superwise.ai/blog/everything-you-need-to-know-about-drift-in-machine-learning.

[6] Wang, Shenghui & Schlobach, Stefan & Klein, Michel, (2010), What Is Concept Drift and How to Measure It? 6317. 241-256. 10.1007/978-3-642-16438-5_17.

[7] Supriya Agrahari, Anil Kumar Singh, 2022, Concept Drift Detection in Data Stream Mining: A literature review, Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 10, Part B, Pages 9523-9540,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2021.11.006

[8] Andrés L. Suárez-Cetrulo, David Quintana, Alejandro Cervantes, 2023, A survey on machine learning for recurring concept drifting data streams,Expert Systems with Applications, Volume 213, Part A, 118934,ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.118934.

Received: August 05, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[9] Benni, Rashmi & Totad, Shashikumar & Mulimani, Deepa & K G, Karibasappa. (2024), Impact analysis of real and virtual concept drifts on the predictive performance of classifiers. Procedia Computer Science. 235. 617-627. 10.1016/j.procs.2024.04.061.
[10] Shyaa, Methaq & Farizah, Noor & Zainol, Zurinahni & Abdullah, Rosni & Anbar, Mohammed & Alzubaidi, Laith. (2024). Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. Engineering Applications of Artificial Intelligence. 10.1016/j.engappai.2024.109143.
[11] Aggarwal, Charu C., ed, 2007, *Data streams: models and algorithms*. Vol. 31. Springer Science & Business Media.
[12] Alejandro Guerra-Manzanares, Marcin Luckner, Hayretdin Bahsi, 2022, Concept drift and cross-device behavior: Challenges and implications for effective android malware detection,Computers & Security,Volume 120, 102757, ISSN 0167-4048, (https://www.sciencedirect.com/science/article/pii/S0167404822001523)
[13] K. Murphy, A. Lavignotte and C. Lepers, April 2024, Fault Prediction for Heterogeneous Telecommunication Networks Using Machine Learning: A Survey, in IEEE Transactions on Network and Service Management, vol. 21, no. 2, pp. 2515-2538, doi: 10.1109/TNSM.2023.3340351.
[14] Hinder, Fabian & Hammer, Barbara. (2021). Concept Drift Segmentation via Kolmogorov-Trees. 41-46. 10.14428/esann/2021.ES2021-93.
[15] Ha, Asmaa & Gunawan, Teddy & Habaebi, Mohamed & Halbouni, Murad & Kartiwi, Mira & Ahmad, Robiah. (2022). Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3151248.
[16] Xin, Yang & Kong, Lingshuang & Liu, Zhi & Chen, Yuling & Li, Yanmiao & Zhu, Hongliang & Mingcheng, Gao & Hou, Haixia & Wang, Chunhua. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2836950.
[17] Wares, S., Isaacs, J. & Elyan, E. (2019), Data stream mining: methods and challenges for handling concept drift. *SN Appl. Sci.* **1**, 1412 https://doi.org/10.1007/s42452-019-1433-0
[18] Gama, João & Medas, Pedro & Castillo, Gladys & Rodrigues, Pedro. (2004). Learning with Drift Detection. Intelligent Data Analysis. 8. 286-295. 10.1007/978-3-540-28645-5_29.
[19] Barros, Roberto & Cabral, Danilo & Alves, Paulo & Santos, Silas. (2017). RDDM: Reactive drift detection method. Expert Systems with Applications. 90. 344-355. 10.1016/j.eswa.2017.08.023.
[20] Baena-Garcıa M, del Campo-Ávila J, Fidalgo R, Bifet A, GavaldaR, Morales-Bueno R (2006) Early drift detection method. In:Fourth international workshop on knowledge discovery from data streams, vol 6. pp 77–86
[21] Muhlbaier MD, Polikar R, (2007), Multiple classifiers based incremental learning algorithm for learning in non-stationary environments. In: 2007 International conference on machine learning and cybernetics, vol 6. IEEE, pp 3618–3623
[22] Domingos P, Hulten G, 2000, Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 71–80
[23] Street WN, Kim Y (2001), A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the seventh ACMSIGKDD international conference on Knowledge discovery and data mining. ACM, pp 377–382
[24] Yongsoo Lee, Yeeun Lee, Eungyu Lee, Taejin Lee, 2023, Explainable Artificial Intelligence-Based Model Drift Detection Applicable to Unsupervised Environments, Computers, Materials and Continua, Volume 76, Issue 2, Pages 1701-1719, ISSN 1546-2218, https://www.sciencedirect.com/science/article/pii/S1546221823001558
[25] Muhammad Amin, Feras Al-Obeidat, Abdallah Tubaishat, Babar Shah, Sajid Anwar, Tamleek Ali Tanveer, 2023, Cyber security and beyond: Detecting malware and concept drift in AI-based sensor data streams using statistical techniques, Computers and Electrical Engineering, Volume 108, 108702, ISSN00457906, (https://www.sciencedirect.com/science/article/pii/S004579062300126X)
[26] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014, A survey on concept drift adaptation. ACM Comput. Surv. 46, 4, Article 44 (April 2014), 37 pages. https://doi.org/10.1145/2523813
[27] Kuncheva, Ludmila. (2008). Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. Proc. Eur. Conf. Artif. Intell.

Received: August 05, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[28] Sergio Ramírez-Gallego, Bartosz Krawczyk, Salvador García, Michał Woźniak, Francisco Herrera, 2017, A survey on data preprocessing for data stream mining:Current status and future directions, Neurocomputing, Volume239, Pages39-57, ISSN 0925-2312, https://www.sciencedirect.com/science/article/pii/S0925231217302631.
[29] Ramírez-Gallego, Sergio & Krawczyk, Bartosz & García, Salvador & Wozniak, Michal & Herrera, Francisco. (2017). A survey on Data Preprocessing for Data Stream Mining: Current status and future directions. Neurocomputing. 239. 10.1016/j.neucom.2017.01.078.
[30] Han, Meng & Chen, Zhiqiang & Li, Muhang & Wu, Hongxin & Zhang, Xilong. (2022). A survey of active and passive concept drift handling methods. Computational Intelligence. 38. 10.1111/coin.12520.
[31] Hoens, T. & Polikar, Robi & Chawla, Nitesh. (2012). Learning from streaming data with concept drift and imbalance: An overview. Progress in Artificial Intelligence. 1. 10.1007/s13748-011-0008-0.
[32] Gonzalez Hidalgo, Juan & Maciel, Bruno Iran Ferreira & Barros, Roberto. (2019). Experimenting with prequential variations for data stream learning evaluation. Computational Intelligence. 35. 670-692. 10.1111/coin.12208.
[33] Gemaque, Rosana & F J Costa, Albert & Giusti, Rafael & Santos, Eulanda. (2020). An overview of unsupervised drift detection methods. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 10. 10.1002/widm.1381.
[34] Bifet, Albert & Gavaldà, Ricard. (2007). Learning from Time-Changing Data with Adaptive Windowing. Proceedings of the 7th SIAM International Conference on Data Mining. 7. 10.1137/1.9781611972771.42.
[35] Gomes, Heitor Murilo & Barddal, Jean Paul & Enembreck, Fabrício & Bifet, Albert. (2017). A Survey on Ensemble Learning for Data Stream Classification. ACM Comput. Surv. 50. 23:1-23:36. 10.1145/3054925.
[36] Minku, Leandro & Yao, Xin. (2012). DDD: A New Ensemble Approach for Dealing with Concept Drift. Knowledge and Data Engineering, IEEE Transactions on. 24. 619 - 633. 10.1109/TKDE.2011.58.
[37] Elie Bursztein and Daniela Oliveira. 2019. Deconstructing the phishing campaigns that target gmail users. *Black Hat USA 2019* (2019).
[38] Kshetri, Nir. (2021). Economics of Artificial Intelligence in Cybersecurity. IT Professional. 23. 73-77. 10.1109/MITP.2021.3100177.
[39] Marcus Botacin. 2024. What do malware analysts want from academia? A survey on the state-of-the-practice to guide research developments. In Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '24). Association for Computing Machinery, New York, NY, USA, 77–96. https://doi.org/10.1145/3678890.3678892
[40] Apruzzese, Giovanni & Anderson, Hyrum & Dambra, Savino & Freeman, David & Pierazzi, Fabio & Roundy, Kevin. (2023). "Real Attackers Don't Compute Gradients": Bridging the Gap Between Adversarial ML Research and Practice. 339-364. 10.1109/SaTML54575.2023.00031.
[41] Apruzzese, Giovanni & Laskov, Pavel & Tastemirova, Aliya. (2022). SoK: The Impact of Unlabelled Data in Cyberthreat Detection 20-42. 10.1109/EuroSP53844.2022.00010.
[42] Ceschin, Fabrício & Botacin, Marcus & Gomes, Heitor Murilo & Soares de Oliveira, Luiz & Grégio, André. (2019). Shallow Security: on the Creation of Adversarial Variants to Evade Machine Learning-Based Malware Detectors. 10.1145/3375894.3375898.
[43] Ceschin, Fabrício & Botacin, Marcus & Lüders, Gabriel & Gomes, Heitor Murilo & Soares de Oliveira, Luiz & Grégio, André. (2020). No Need to Teach New Tricks to Old Malware: Winning an Evasion Challenge with XOR-based Adversarial Samples. 13-22. 10.1145/3433667.3433669.
[44] Tobias Braun, Irdin Pekaric, and Giovanni Apruzzese. 2024. Understanding the Process of Data Labeling in Cybersecurity. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24). Association for Computing Machinery, New York, NY, USA, 1596–1605. https://doi.org/10.1145/3605098.3636046
[45] Sugandh Seth, Kuljit Kaur Chahal, Gurvinder Singh, July 2024, Concept Drift–Based Intrusion Detection For Evolving Data Stream Classification In IDS: Approaches And Comparative Study, *The Computer Journal*, Volume 67, Issue 7, Pages 2529–2547, https://doi.org/10.1093/comjnl/bxae023
[46] Sugandh Seth, Kuljit Kaur Chahal, Gurvinder Singh. July 2024, *The Computer Journal*, Volume 67, Issue 7, Pages 2529–2547, 03 March 2024 https://doi.org/10.1093/comjnl/bxae023

Received: August 05, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[47] C. Nixon, M. Sedky and M. Hassan, 2021, Reviews in Online Data Stream and Active Learning for Cyber Intrusion Detection - A Systematic Literature Review, 2021 Sixth International Conference on Fog and Mobile Edge Computing (FMEC), Gandia, Spain, pp. 1-6, doi: 10.1109/FMEC54266.2021.9732566.
[48] Emad E. Abdallah, Wafa' Eleisah, Ahmed Fawzi Otoom, 2022, Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey, Procedia Computer Science, Volume 201, Pages 205-212, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.03.029.
[49] Malhotra, Heena & Sharma, Prabha. (2019). Intrusion Detection using Machine Learning and Feature Selection. International Journal of Computer Network and Information Security. 11. 43-52. 10.5815/ijcnis.2019.04.06.

Received: August 05, 2025