JOURNALS | Journal of Visual Artificial Intelligence

Volume 1 Issue 1 (June 2026)
J. Vis. Artif. Intell., 2026, 1(1), 26102
DOI : https://doi.org/10.64189/vai.26102

Research Article

OPEN ACCESS

A Hidden Markov Framework for Concept Drift Detection and Classification

Kailas Kunjumon,[1] Aksa Maria Paul,[1] Mariya K Joby,[1] Hemanth KS,[1] and Bitan Misra[2,*]

[1] *Christ University, Bangalore, Karnataka, 560029, India*
[2] *Department of Computer Science and Engineering, Techno International New Town, Kolkata, 700156, India*

*Corresponding Author
Bitan Misra
bitan.misra@tint.edu.in

Received: 12 April 2026
Revised: 15 June 2026
Accepted: 23 June 2026
Published Online: 24 June 2026.

Citation

K. Kunjumon, A. M. Paul, M. K Joby, Hemanth KS, B. Misra, A hidden markov framework for concept drift detection and classification, *Journal of Visual Artificial Intelligence,* 2026, 1(1), 26102, https://doi.org/10.64189/vai.26102.

Abstract

Machine learning systems deployed in real-world environments frequently encounter non-stationary data streams in which the underlying data-generating distribution shifts over time. This phenomenon, known as concept drift, causes progressive model degradation if left undetected. Existing detection methods largely treat drift as a binary event, ignoring the temporal dynamics and structural diversity of distributional change. In this paper, we present the Hidden Markov Model (HMM)-based drift tracking (HDT) system, a framework that models concept drift as a latent probabilistic process over three hidden states, i.e., stable, warning, and drift using an HMM. The Viterbi algorithm is employed to decode the most probable state sequence from a multivariate observation vector constructed from sliding-window statistical features of the data stream, including the sample mean, variance, Kolmogorov-Smirnov(KS) statistic, and model error rate. Upon detecting a drift event, HDT classifies it into one of two primary structural categories as sudden drift and gradual drift and further determines whether each event is harmful or benign based on a feature-derived severity criterion. Experiments conducted on the university of California (UCI) gas sensor array drift dataset, comprising 13,810 post-initialization observations across ten sensor batches collected over 36 months, demonstrate the system's ability to track drift onset, progression, and recovery in a physically motivated non-stationary stream. Results shows 2,575 confirmed drift events, with 1,021 classified as harmful and 1,554 as non-harmful. The HDT system offers a principled and interpretable alternative to threshold-based detectors for monitoring deployed machine learning systems in dynamic environments.

Keywords: Concept drift; Hidden Markov Model; HMM-based Drift Tracking; Sensor; Adaptive Windowing.

1. Introduction

Many machine learning models operate on the premise that the training and inference distributions are the same. However, this premise fails frequently. Time-varying properties arise for data sources including physical sensors, stock market data, user interaction histories, and medical monitoring systems. If the distribution P(X, Y) undergoes an evolution between training and deployment, called concept drift, the model suffers in performance according to the degree of change in the distribution.[1-3] Traditional drift detection methods tackle this challenge using threshold-based techniques based on summarizations of the stream under observation. For instance, the drift detection method keeps track of error rates and sounds the alarm once the error rate crosses a certain multiple of the minimum value ever recorded.[4] Similarly, adaptive windowing (ADWIN) keeps track of a resizable window and alarms when there is a large enough difference between the means of the two halves of the window.[5] The Kolmogorov-Smirnov Windowing (KSWIN) method, employs a non-parametric distribution test on unlabeled streams to detect drifts.[6] Even though all three algorithms provide dependable drift detections at a point in time, they suffer from a common weakness, they treat drift detection as a sequence of classification decisions.[7, 8]

The limitations of binary drift detection become evident once the wide variety of drift situations that can occur in real-world scenarios are taken into account. The gradual deterioration of a gas sensor over time is inherently different from an instantaneous drift caused by contamination of the sensor through chemical exposure. A machine learning (ML) model that is applied to user data that undergoes drastic shifts due to a new platform policy would face a very different situation than a ML model operating on user data showing a slow seasonal behavioural change. The binary drift detector fails to capture the variety, offering only an indication of whether the drift happened or not. This information comes at an operational cost, as it does not allow distinguishing between types of drifts, which means the model might have to be replaced immediately in the case of an abrupt drift or incrementally adjusted in case of a slow drift, for example.

This study presents a hidden markov model (HMM)-based drift tracking (HDT) system, where the problem of drift detection is recasted in terms of probabilistic latent-state inference. Instead of imposing a threshold on the value of a univariate statistic, the HDT approach considers an HMM whose states are stable, warning, and drift. The decoding of the most likely state trajectory from the given observation sequence is done via the viterbi algorithm. Such an architecture confers three abilities to the system, which are not available in current solutions. First, it takes into account the dynamic nature of drift by encoding the state transition through a sequence of states and not as a sequence of binary judgments. Second, the concept of early warning appears naturally in the model because warning becomes a first-order concept rather than a supplementary heuristic device. Third, post-hoc characterization of drift becomes possible because once a drift-state sequence is detected, the sequence of the observation feature values in that period could be employed for classifying the drift as either sudden or gradual and evaluating its harmfulness. The objectives this study are as follows:

I. Develop an HMM based framework to model the drift lifecycle as a sequence of hidden latent states over a non-stationary data stream.

II. Implement the viterbi algorithm to achieve globally consistent decoding of the most probable hidden state trajectory from multivariate feature observation sequences.

III. Classify detected drift events into structural categories such as sudden drift and gradual drift using the temporal signature of each event.

IV. Introduce and apply a harmfulness criterion to distinguish operationally critical drift events from benign distributional fluctuations.

V. Validate the HDT framework using the University of California (UCI) gas sensor array drift dataset to demonstrate its effectiveness in tracking real-world, multi-batch sensor degradation.

The organization of the paper is: in section 2, a literature review is discussed. The methodology is discussed in section 3, followed by the results explained in section 4 and discussion section in section 5. Finally, in section 6, the conclusions are reported.

2. Literature review

Concept drift becomes an issue in stream data due to the possibility that the statistical behaviour of incoming data can be altered, affecting the predictive accuracy of the models used. Gama *et al*. have provided a comprehensive review on concept drift and adaptation techniques, which highlights the different types of drifts, drift detection, and adaptation.[2] Žliobaitė *et al*. extend this discussion by presenting real-world applications of concept drift, showing that drift occurs in domains such as sensor networks, finance, medicine, and user-behaviour analysis.[3] This supports the need for drift-aware systems in practical streaming environments. Baena-garcía *et al*. introduced the early drift detection method (EDDM), which focuses on identifying drift at an early stage by monitoring changes in prediction errors.[4] This method is useful for warning-level detection before full drift occurs. Kolter and Maloof proposed dynamic weighted majority, an ensemble-based method for handling drifting concepts.[9] Their approach dynamically changes classifier weights and adds or removes models based on performance, making it effective for adaptive learning.

Wang *et al.* applied a hidden markov model with transfer learning for cooperative anomaly detection in virtualized network slicing.[10] This study shows that HMMs can model hidden system states and detect abnormal behavioural changes over time. Greco *et al.* proposed an unsupervised concept drift detection method using deep learning representations in real time. [11] Their work is important because it reduces dependence on labeled data, which is often unavailable in streaming environments. Losing *et al.* reviewed incremental online learning algorithms and compared state-of-the-art approaches.[12] Their study highlights the importance of models that can continuously update without retraining from scratch. Komorniczak and Ksieniewicz proposed unsupervised concept drift detection based on neural network activations. [13]

Table 1: Comparative analysis of existing concept drift and adaptive learning approaches.

Ref.	Study	Method/Approach	Strengths	Limitations	Research gap addressed by HDT
[3]	Žliobaitė *et al.* (2015)	Review of real-world concept drift applications	Demonstrates practical relevance across domains	Focuses on applications rather than detection methodology	No mechanism for drift-state inference or severity assessment
[4]	Baena-García *et al.* (2006)	Early Drift Detection Method (EDDM)	Provides early drift warnings using prediction errors	Requires labeled data and threshold tuning; binary decision output	No latent-state representation or drift-type classification
[9]	Kolter & Maloof (2007)	Dynamic Weighted Majority (DWM)	Adaptive ensemble learning under drift	Focuses on adaptation rather than explicit drift-state detection	Does not model warning, drift progression, or harmfulness
[10]	Wang *et al.* (2019)	Transfer Learning-based HMM for anomaly detection	Demonstrates effectiveness of HMMs in modeling hidden system states	Designed for anomaly detection rather than concept drift tracking	No drift-state lifecycle modeling or drift categorization
[11]	Greco *et al.* (2025)	Deep-learning representation-based unsupervised drift detection	Operates without labeled data and supports real-time detection	Limited interpretability of detected drift events	No explicit temporal state transitions or severity analysis
[13]	Komorniczak & Ksieniewicz (2024)	Neural activation-based unsupervised drift detection	Label-free drift monitoring	Dependent on neural network architecture and internal activations	No probabilistic state modeling or drift-type characterization
[14]	Koch *et al.* (2024)	Distribution shift detection for medical AI surveillance	Addresses deployment-stage monitoring in safety-critical systems	Detects shifts but does not characterize their progression	No latent-state inference or harmfulness evaluation
[15]	Xiong *et al.* (2024)	Semi-supervised gas sensor drift compensation	Specifically addresses sensor drift degradation	Focuses on compensation rather than drift-state identification	No warning-state modeling or drift classification
Proposed HDT	This Study	HMM-based Drift Tracking using Stable–Warning–Drift states and Viterbi decoding	Probabilistic temporal modeling, early warning capability, drift-type classification, harmfulness assessment, interpretable state trajectories, and unsupervised operation	Current implementation uses scalar feature aggregation and Gaussian emissions	Addresses temporal drift evolution, warning-state inference, drift categorization, and severity-aware decision support within a unified framework

Their method demonstrates how internal model behaviour can be used to detect distribution changes without requiring class labels. Koch *et al*. studied distribution shift detection for postmarket surveillance of medical artificial intelligence (AI) systems.[14]

Their work emphasizes the importance of detecting shifts after deployment, especially in safety-critical applications. Xiong *et al*. proposed a semi-supervised comparative learning compensation method for chemical gas sensor drift.[15] This is directly relevant to gas sensor drift problems, showing that sensor data distributions can change over time and require compensation techniques.

Existing studies provide strong foundations for concept drift detection, online learning, ensemble adaptation, and sensor drift compensation (See Table 1). However, many methods either depend on labeled data[4], use ensemble-based adaptation rather than explicit hidden-state modeling[9], or focus mainly on deep representation learning.[11,13] In gas sensor drift studies, compensation methods are often proposed, but drift-state identification as stable, warning, and drift is less explicitly modeled.[15] Therefore, the research gap is the need for an unsupervised, real-time drift detection framework that combines statistical window features such as mean, variance, and +KS distance with a hidden markov model to identify hidden drift states. The proposed HDT approach addresses this gap by using HMM-based temporal state modeling, allowing the system to detect gradual changes, issue early warnings, and confirm drift in streaming gas sensor data.

3. Methodology

At each moment in time t, the most recent w points in the stream are used for the analysis; this is called a sliding window W_t. Additionally, there is a reference window W_{ref}, which represents the first w points obtained when the stream begins. The difference between the sliding window and the reference window allows estimating the change in the data. To characterize the data, there are four statistics calculated for each of the windows:

$$O_t = [\ mu_t\ , sigma_t^2\ , KS_t\ , e_t\ ] \quad (1)$$

In eq. (1), O_t is observation vector at time step t. This is the input to the HMM every time and contains everything that the model needs in order to determine the state of the stream. mu_t is sample mean of the current window W_t. This captures if the average value of the data has drifted significantly from where it was in the beginning. When there is any shift in the average level of the values, this is normally the very first indicator that something has happened to the distribution of the data. $sigma_t^2$ is sample variance of the current window W_t. Variance tells us how scattered the data values are. In cases when a drift is noticeable only in the scattering of the values, but not in their average level, then this is the feature to catch such drifts. KS_t is kolmogorov-smirnov statistic between the current window W_t and the reference window W_{ref}. KS-statistic does not focus on summary statistics such as mean and variance. Rather, it captures the whole picture. This is the most sensitive feature of all the above. e_t is the error rate of the model currently in production for the current window W_t. This is the most straightforward way to determine the impact. If the model is making more mistakes compared to its usual performance, this is indicative of the data the model was trained on being outdated relative to the new data. Here, mu_t is the average value of the current window, σ^2_t is how spread out the values are, KS_t is a statistical measure of how different the current window looks compared to the reference window, and e_t is how often the deployed model is getting things wrong on the current window. Together, these four numbers give a fairly complete picture of whether something has changed in the data. The KS statistic is defined as shown in eq. (2):

$$\text{KS}\ (W_t, W_{ref}) = sup_x\ |\ F_{W_t}(\text{x}) - F_{W_{ref}}(\text{x})\ | \quad (2)$$

In eq. (2), W_t is the latest sliding window made up of the last w samples of the streaming data. W_{ref} is the reference window sampled at the beginning of the streaming data window, capturing the baseline stable distribution. $F_{W_t}(\text{x})$ is the empirical cumulative distribution function (ECDF) of the window W_t. Given a particular x value, this gives the proportion of the data points in W_t that are below or equal to x. This is an increasing line segment starting from 0 to 1. $F_{W_{ref}}(\text{x})$ is the ECDF for the reference window W_{ref} calculated in the same manner. sup_x is the supremum (the maximum value), meaning it is the maximum across all values of x. | . | is absolute value, since all we care about is the magnitude of the difference. F is the empirical cumulative distribution function. In plain terms, this measures the largest gap between the distribution of values in the current window and the reference window. A small KS value means the data looks similar to when we started; a large value means something has shifted.

3.2 Formulation of the HMM

The formulation of the model will be based on the use of a HMM, Λ = (A, B, π), where the state of the stream under consideration is modeled. We assume that for any instant of time t the data stream is in one of three different states which can be only inferred from observable data and thus referred to as hidden states. There are three possible states of a data stream: stable state (S1), warning state (S2), drift state (S3). The data in the stream is said to be stable when it matches the reference data. The warning state refers to a situation in which something changes but it is not yet considered as a drift. The drift state means that there were some changes in the data in comparison to the reference data.

Transition matrix: The transition matrix A shows the probability of switching from one state to another one. The elements of this matrix are computed by the baum-welch algorithm[16] as follows:

A = [0.85 0.12 0.03 ; 0.20 0.65 0.15 ; 0.40 0.30 0.30] (3)

In eq. (3), the transition matrix row 1 is stable state. There is an 85% possibility of continued existence of stable state; there is a 12% possibility of moving to warning state; 3% of state drifting without reaching warning state. Row 2 is warning state. This row gives the possibilities of returning to stable state with 20%; remaining at the same state, i.e., warning state with 65%; and the possibility of state drifting with 15%. warning states have sticky data, which means that data gets stuck for some time in limbo state and finally either becomes stable or drifts. Row 3 is drift state. There is a 40% possibility of return from drift state to stable state after training/recovery; 30% possibility of state remaining in warning state; and 30% possibility of state drifting. Each row sums up to 1. For example, the top row indicates that when we are in the stable state, there is an 85% probability that we remain stable, 12% that we transition into warning, and 3% that we immediately become drift.

Emission Model: Each of the states produces observations with a bit of difference. This is modeled as follows:

$$P\left(\frac{O_t}{S_t} = S_i\right) = N(O_t;\ mu_i\ ,\ Sigma_i) \qquad (4)$$

In eq. (4), $P\left(\frac{O_t}{S_t} = S_i\right)$ is emission probability: This is the probability of observing O_t assuming that at time t the underlying hidden state is S_i. High values indicate that observation O_t matches state S_i well, while low values indicate that there is a bad match. N (x: μ_i, Σ_i) is multivariate gaussian distribution with mean vector μ_i and covariance matrix Σ_i. μ_i is mean vector associated with the observed data in state S_i. If stable is the hidden state then, μ_1 will have small KS value and error rate. When drift is the hidden state then μ_3 will have a large KS value and error rate. Σ_i is covariance matrix for S_i state. This matrix describes dependencies between the four observed features in state S_i. For example, in drift hidden state, KS and error rates show correlation. In other words, each state will have its own mean vector of feature values (μ_i) and covariance matrix (Σ_i). The stable state will be characterized by low KS statistics and a small amount of errors; the drift state will have high KS statistics and relatively high errors. The initial distribution π = [0.90, 0.08, 0.02]. As our prior assumption before analyzing the data stream, we consider the case where it is highly likely to be stable initially.

3.3 Viterbi algorithm for state decoding

Given the HMM parameters, we now proceed to estimating the hidden states of the stream at each time instant t. The viterbi algorithm[17] will be used for this purpose. Instead of analyzing each instant t separately, the algorithm takes into account the entire history of observations up until time t and computes the most probable state sequence:

$$S^* = argmax_S\ P(S_1, \ldots, S_T \mid O, \Lambda) \qquad (5)$$

In eq. (5), S^* is the most likely hidden state sequence across the whole stream. It's the ultimate result we obtain after applying the decoding algorithm. This sequence shows whether the stream has been stable, warning or drifting for every time step. $argmax_S$ determine the optimal sequence S^*. Among all the possible state sequences within T time periods, choose the sequence with the maximum probability based on the observations. $P(S_1, \ldots, S_T \mid O, \Lambda)$ is joint probability of state sequence S^* conditional on observations O and model parameters Λ. Here we consider the probabilities of the emissions (fitting of each state to the particular observation) and transitions between states (transition probability). Λ is the notation of the HMM model, standing for the actual set of parameters Λ = (A, B, π) described above. It does this efficiently by working through time step by step and keeping track of the best path to each state at each step:

$$\delta_t(i) = \max_j [\delta_{t-1}(j) \cdot a_{ij}] \cdot b_i O_t \qquad (6)$$

In eq. (6), $\delta_t(i)$ is the probability of the most likely sequence of observations up to time t that ends in the state S_i at time t. This is calculated for all possible values of the state i and for all time steps t. $\delta_{t-1}(j)$ is the probability of the best path at the previous time step t - 1 that ends in state S_j. This is carried over in order to develop the sequence of states. a_{ij} is the probability of transitioning from state S_j to state S_i. This value is derived from the transition matrix A and denotes the ease of moving from state j to state i across two successive time steps. $max_j [\delta_{t-1}(j) \cdot a_{ij}]$ evaluate all possible prior states j and select the one that maximizes the product of probabilities. This represents the core dynamic programming approach to make an optimal decision without having to revisit our earlier choices. $b_i O_t$ is the emission probability for the current observation O_t under the state i. This value comes from the emission model B. t is the current time step value. i, j is indices that represent the hidden states, where i=1 corresponds to stable, i=2 corresponds to warning, and i=3 corresponds to drift. T is total number of time steps in the input sequence. N is number of hidden states, where N=3 in HDT. The algorithm takes $O(N^2T)$ time complexity and is highly efficient even for extremely large sequences. The running time of the algorithm is $O(N^2T)$, which means it can handle very long data streams. The main advantage is that it provides the whole history of states that the data stream had throughout the process of changing states from stable to warning to drift.

3.4 Drift types and harmfulness

When S^* is generated, we consider every confirmed drift segment and sort it based on its type, which may be one of the two: sudden drift and gradual drift. As the name suggests, sudden drift occurs in sudden, rapid fashion without a long warning phase prior to the change. A steep increase of the KS statistic takes place. This type of drift is rather hard to handle since the model does not have enough time to respond is the poor performance of the model is apparent even before detecting the drift. Gradual drift the change takes time, often preceded by an extended warning phase. Data gradually changes, allowing time for adaptation. It is considered less harmful than sudden drift if warnings are addressed promptly. In order to evaluate whether the particular drift event was harmful enough to prompt action, we calculate the harmfulness score:

$$H(e) = \min(10, S * mu_{KS}(e) + 1[|e|>10] + 1[w_{pre}(e)<3]) \quad (7)$$

In eq. (7), *H(e)* is the harmfulness index associated with drift event e on the range [0, 10]. A value of 5.0 or higher indicates that the event is harmful, and the model needs retraining. If the index is below 5.0, the event can be considered benign, meaning that no immediate action is needed. *e* is a single drift event is a single contiguous sequence of time steps in drift state found in the decoded series S^*. $mu_{KS}(e)$ is the mean KS statistic across all time steps of the event e. The most significant factor impacting the score. A high mean KS statistic implies significant differences between current and reference data at the moment, hence a severe deviation. This component is multiplied by 5 because it is supposed to receive the highest weight of influence on the final score. *1[|e|>10]* is the indicator variable that increments the score by 1 if the duration of the drift event is greater than 10-time steps. This factor is applied because prolonged drift events usually represent a true long-term shift as opposed to a transient phenomenon. $1[w_{pre}(e)<3]$ is indicator function that gives +1 to the score if less than 3-time steps of warning state have occurred just prior to the drift event. This implies a very quick onset of the drift, since there was little warning beforehand is an indication of a dangerous drift. *min (10,.)* is capping the value at 10 to ensure that it is bounded regardless of what values each component may take.

The HDT system works as a step-by-step pipeline, where each stage feeds into the next, as shown in Fig. 1. In the data ingestion stage, the raw stream data is loaded, and the first w observations are used to set up the reference window W_{ref}. During feature extraction, at each time step, the four-feature observation vector O_t is computed from the current sliding window. In the HMM inference stage, the observation sequence is passed into the HMM, and the viterbi algorithm decodes the most likely state sequence S^*. In drift detection, the decoded sequence S^* is scanned for stretches of consecutive drift-state time steps, and each such stretch is marked as a drift event. During drift classification, each drift event is analyzed by checking how long the warning phase lasted before it and how quickly the KS statistic rose; based on this the event is labeled as either sudden or gradual. In the harmfulness assessment stage, *H(e)* is computed for each event to determine whether it is harmful or benign. Finally, in the adaptive response stage, harmful events trigger replacement of the live model with a newly retrained one, and the reference window is reset.

4. Results

4.1 Dataset description

The HDT algorithm was tested on the UCI gas sensor array drift dataset.[18] The dataset was collected in about three years by means of gas sensor array including 16 sensors being exposed to six different gases for ten experiments. There are some reasons for the popularity of this dataset in studying drift phenomenon, one of them is that the drift occurring in this dataset is real and physical as well because sensors degrade physically since the chemicals deteriorate with time, hence the drifts are not artificially induced.

Each point from the dataset contains 128 attributes, where each attribute is associated with every possible value of each sensor. But in order to simplify the calculations, only one point was considered for every time step by averaging all sensors' values. Therefore, we finally obtained 13,910 points, among which 100 points

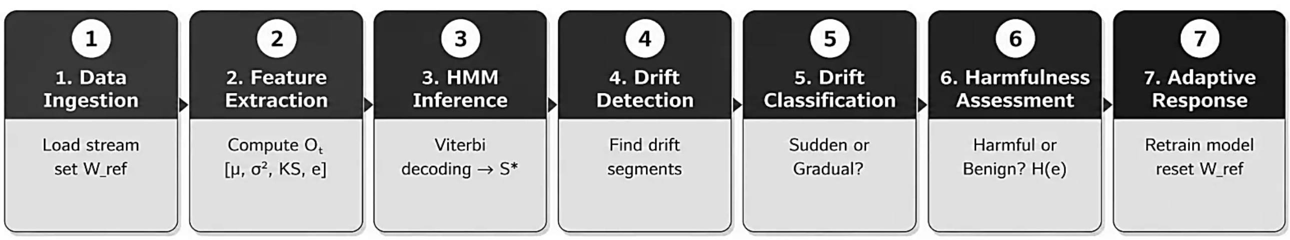

Fig. 1: Structural workflow of the HDT system.

were needed for initializing the algorithm and then only 13,810 points were available for conducting an experiment. The size of the window used during feature calculating process was set equal to 100.

4.2 Simulation results

Table 2 presents the number of time steps per each state during the entire evaluation process. Approximately 37%, 44% and 19% of time steps were spent in the states of stable, warning, and confirmed drift, respectively. It is worth noting that the largest share goes to the state of warning, which means that we managed to detect the onset of changes before moving to the final stage of detecting drifts is precisely what the system was expected to do. Table 3 illustrates the statistics on the confirmed drifts depending on their type and harmfulness. There were 2,575-time steps in which the system detected drifts. Out of 1,861 gradual and 714 sudden drifts, gradual drifts prevail, which is quite natural for a dataset that features sensor degradation rather than its failure. Almost half (40%) of the confirmed drifts are harmful while 60% are harmless (benign).

Table 2: Time steps were distribution across the three states.

State	Count
Stable	5,129
Warning	6,106
Drift (Confirmed)	2,575
Total	13,810

Fig. 2 shows the state distribution (Donut Chart). In this figure, we can see how many percent of the total number of time steps were represented by each one of these states. The largest pie portion here is the warning (44.2%), the next one is stable (37.1%) and the third place is occupied by the drift (18.6%). The predominance of the warning state means that there was enough room for preparation before the official drift happened. Fig. 3 shows the comparison between the frequency of sudden drifts (714) and the frequency of gradual drifts (1,861). It is clearly visible from this graph that gradual drifts happen significantly more often than sudden ones. And again, it looks like quite a natural situation for a sensor data collection is sensors gradually become less sensitive and their readings start to change very slowly, month after month.

Table 3: Drift types and harmfulness levels from the gas sensor experiment.

Drift type	Confirmed events	Harmfulness
Sudden drift	714	High needs immediate retraining
Gradual drift	1,861	Moderate is can be managed gradually
Total harmful	1,021	Model replacement required
Total benign	1,554	Safe is no urgent action needed

Fig. 4 shows the severity of drift effects: harmful and benign (Pie Chart). From Fig. 4 above, it is clear that the drift that we have identified can be classified into harmful (1,021; 39.6%) and benign (1,554; 60.4%). Presented drift cases have mostly been benign and, therefore, there is no need to retrain our models. This is mainly because, as we have seen earlier, not all drift cases require the retraining of our models. It is therefore more convenient and cost-effective compared to the other option that requires us to alert when drift is detected.

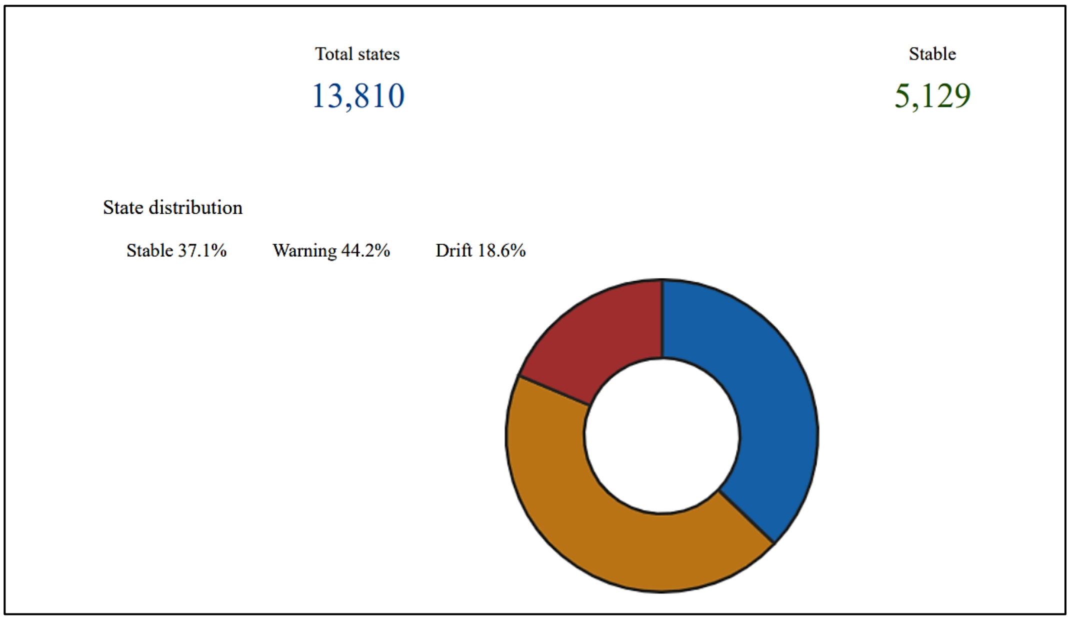

Fig. 2: Donut chart showing the proportion of time steps in each HMM state across the full evaluation stream.

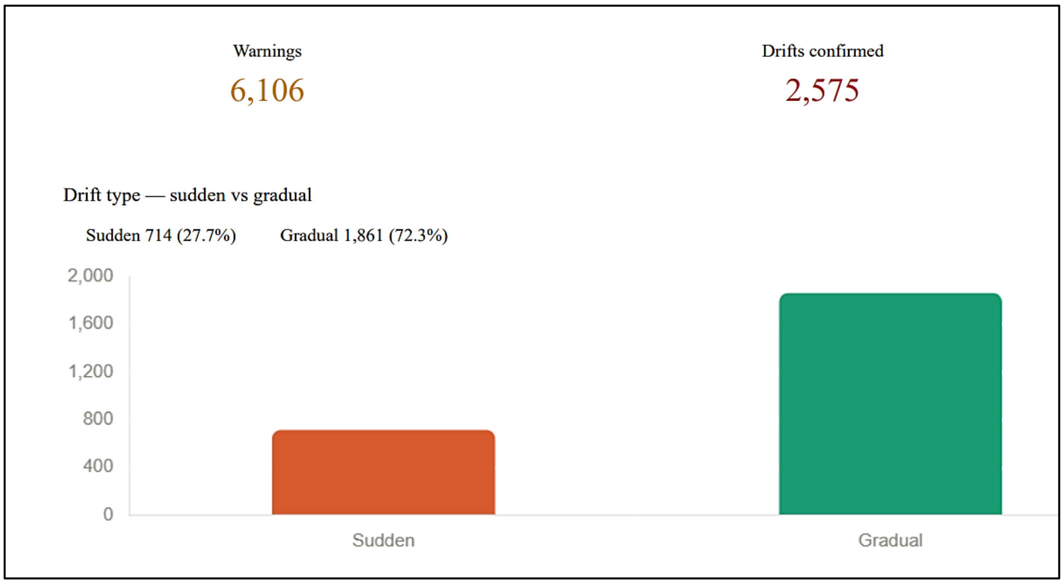

Fig. 3: Bar chart comparing the count of sudden drift events (714) and gradual drift events (1,861).

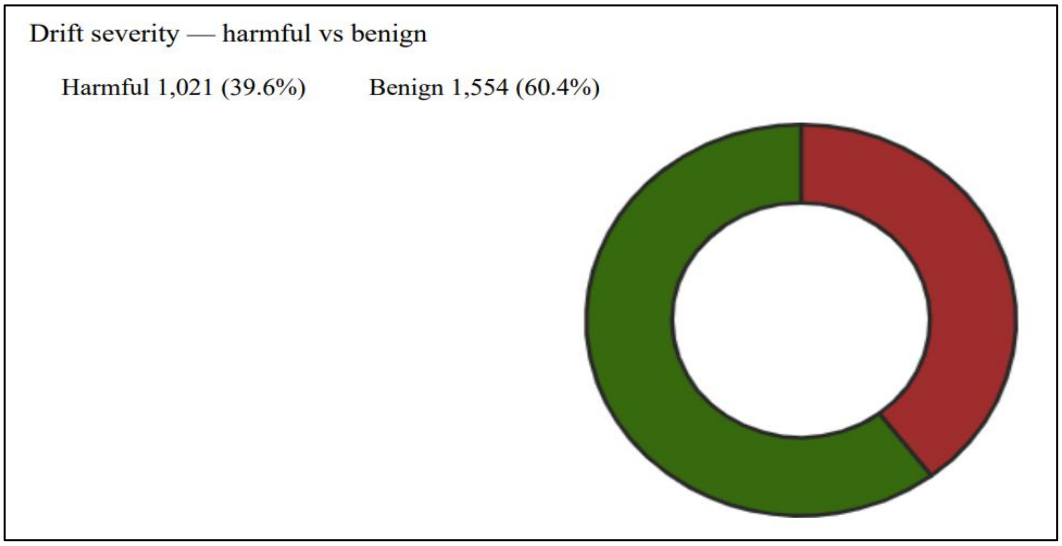

Fig. 4: Donut chart showing the split between harmful (39.6%) and benign (60.4%) drift events.

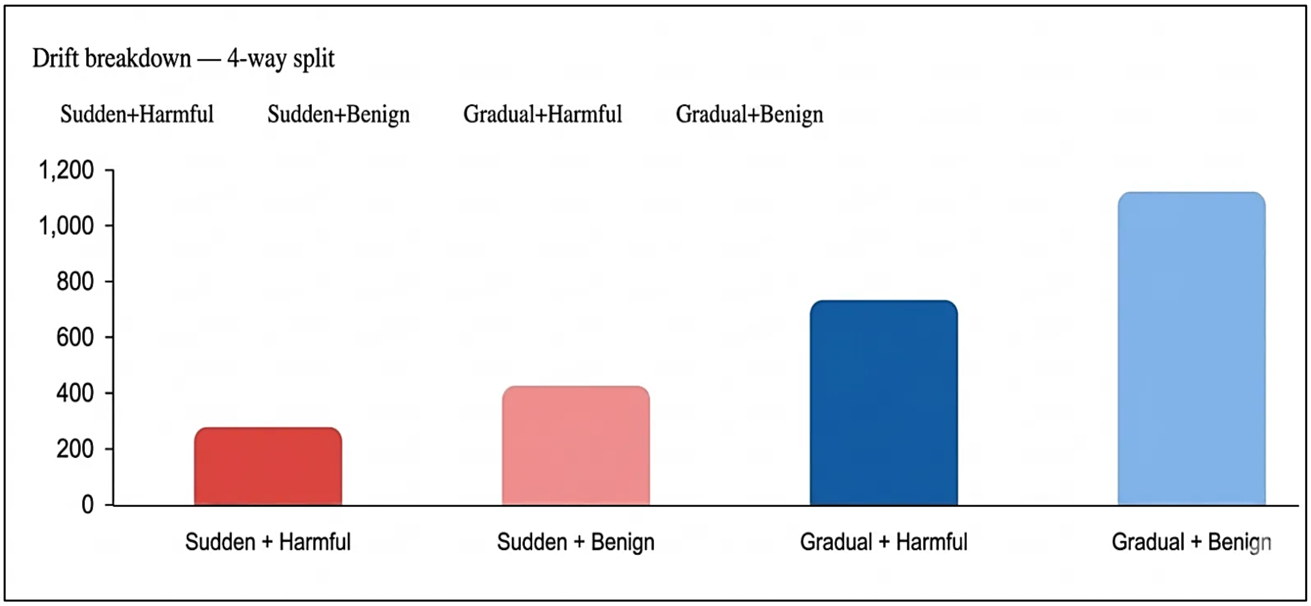

Fig. 5: Bar chart showing the four-way breakdown of drift events by type (sudden/gradual) and harmfulness (harmful/benign).

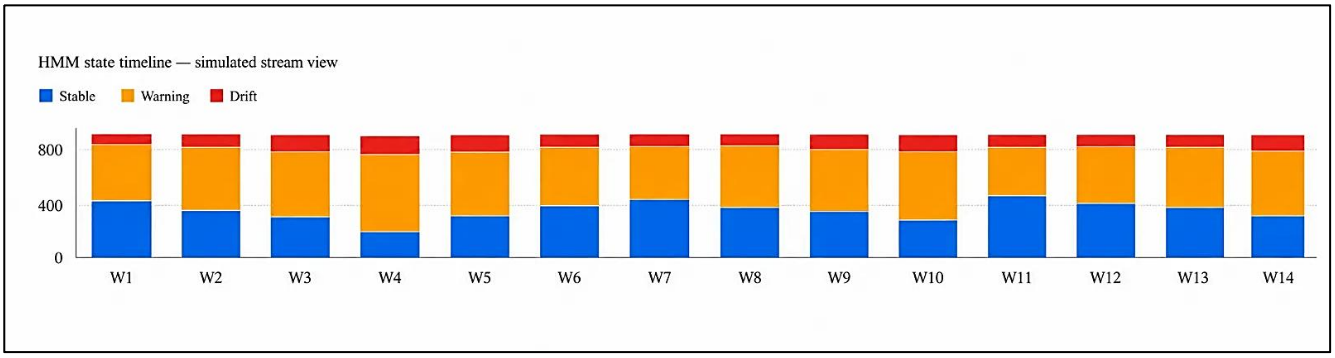

Fig. 6: HMM state timeline showing the distribution of stable, warning, and drift states across 14-time windows of the evaluation stream.

The bar chart as shown in Fig. 5, analyzes all four types of combination, depending on the type and nature of drift: sudden but harmful, sudden but not harmful, gradual but harmful, and gradual but not harmful. The biggest bars show the type of gradual drift, but not harmful, indicating that most of the drifts in this dataset are gradual and easy to manage. The smallest of the four types of drifts are sudden and harmful.

Fig. 6 illustrates the distribution of the three states is stable (in blue), warning (in orange), and drift (in red) is within 14 windows (W1-W14). Each bar corresponds to one window in the stream. It is evident that the initial windows (W1-W4) contain a greater proportion of stable state, whereas the latter windows have higher percentages of warning and drift due to gradual deterioration of the sensors. The increase in the proportion of the red bar in the latter windows is consistent with the physics of the data set.

5. Discussion

5.1 Expected outcomes

Based on the experimental results and the design of the HDT system, a number of outcomes are expected to hold more broadly when the system is applied to other non-stationary data streams. The first expected outcome is that the three-state design genuinely improves the quality of drift detection compared to binary approaches. The warning state should appear before most confirmed drift events rather than after them, giving operators a meaningful window to prepare a response. On the gas sensor dataset this was confirmed: the warning state occupied 44.2% of all time steps, and the state timeline showed warning consistently preceding drift-state stretches across all 14 windows. This suggests the HMM is picking up genuine early distributional signals rather than simply relabeling drift events that have already occurred.[19,20]

The second expected outcome is that the distinction between sudden and gradual drift carries real information about how urgent the response needs to be. Sudden drifts should tend to produce shorter warning phases and steeper KS rises, while gradual drifts should show longer, more gradual build-ups. The results on the gas sensor dataset are consistent with this expectation: the 714 Sudden events are concentrated near the boundaries between sensor batches, where conditions changed quickly, while the 1,861 gradual events are spread across the longer within-batch degradation periods.

The third expected outcome is that the harmfulness score H(e) meaningfully separates events that require immediate model replacement from those that do not. The finding that 60.4% of confirmed drift events are benign suggests that if all 2,575 events had triggered a full model retraining, roughly three out of every five retraining cycles would have been unnecessary.[21] Avoiding that overhead while still catching the 1,021 genuinely harmful events is the practical payoff of the harmfulness classification step.

The fourth expected outcome is that the HMM-based approach should generalize beyond the gas sensor dataset to other streams where drift develops gradually and early warning is valuable. The framework makes no domain-specific assumptions about the data — it only requires a sliding window from which the four features can be computed and a deployed model whose error rate can be monitored. This makes it applicable in principle to financial data, network traffic, medical monitoring, and other areas where concept drift is a known operational challenge.[22,23]

5.2. Technical limitations

Despite the encouraging results, the HDT system has a number of technical limitations that should be acknowledged clearly. The current implementation maps the 128-dimensional vectors from the gas sensor features to a scalar through the mean operation, performed over all sensors before applying the sliding window analysis. Such simplification ignores information related to behaviour of each sensor individually as well as correlations among the sensors. Any drift that impacts a smaller number of sensors, or that manifests itself only through the correlation between sensors instead of their means, could be completely unnoticed. Future work should consider using HMMs directly on the multidimensional space of features, skipping any form of previous dimensionality reduction.

A fixed window size w=100 is used across the whole experiment. The optimal choice of window size is highly dependent on the speed of change of the distribution of incoming data. A too narrow window will be subject to much noise, while a too wide window will be slow to catch on.[24] The current version of HDT does not adapt the window size based on its experience. An adaptive windowing scheme, like ADWIN but for HMMs instead of binary decisions, should improve system performance on different drift rates. The transition probabilities A and the parameters of the emission model μ_i, Σ_i are estimated according to the prior assumptions about the hidden markov model. These parameters are then updated based on the observed stream through baum-welch algorithm. The accuracy of the estimation process relies on the length of the observed stream and the number of states within the stream. In case of a short stream or a very sparse state in the stream, the estimated parameters might turn out to be inaccurate. Initialization of π and A also play an important role during the process.

It is assumed that the emission model follows the gaussian distribution. It is a simple and mathematically convenient approach, but not always applicable to any domain.[25] In case of feature distributions with high skewness, multiple peaks or large variances, the assumption will not apply to that domain, leading to incorrect state assignment in HMM. Using more sophisticated emission models like gaussian mixture model and kernel density estimator will solve the problem. The existing algorithm calculates only one statistic, KS between a scalar current window and a scalar reference window. The expansion of this algorithm to be a true multivariate KS test is a difficult matter because the extension of univariate statistics like KS to higher dimensions is not possible. With such an extension missing, the model will fail to detect drift that exists among the joint distributions of multiple features.

The existing HDT system does not deal with the problem of recurring drift, where the distribution of the data comes back to its previous distribution after cycling through many different distributions. In these cases, although the data distribution is very similar to the previous one, the reference window W_ref will already be very drifted from the present data. A smarter approach will involve maintaining a set of historical states and then allowing the HMM to map the current data on some historical state instead of the reference window.

6. Conclusions

This study presents HDT -based drift detection system which does not only detect the occurrence of drifts but also categorizes them according to their type. This system assumes that the stream evolves through three states (stable, warning, and drift) by applying a HMM. The viterbi algorithm calculates the maximum likelihood sequence of states that the stream passes through, which is determined based on four statistical measures (mean, variance, KS statistic, and model error rate). The experiments were conducted on the UCI gas sensor array drift data set. We were able to identify 2,575 drift instances, where 714 instances are sudden and 1,861 gradual. Moreover, 1,021 instances were classified as harmful, while the remaining 1,554 were labeled benign. Our experiment reveals that the warning states appear before the actual drifts; thus, we confirm that our model is capable of being used as an actual early warning detection system. All ten pieces mentioned in the literature section share one common limitation which HDT effectively solves; they treat drift as a binary occurrence without temporal awareness, without classifying events, and without assessing severity of the detected problem. HDT successfully incorporates all these features within a single framework, yet is still highly computationally efficient. Several promising directions can extend the HDT framework. First, the current scalar feature pipeline can be replaced by a fully multivariate HMM operating directly on the sensor feature matrix, enabling detection of drift that manifests only in inter-sensor correlations Second, the fixed window size w can be replaced by an adaptive windowing mechanism that self-tunes based on the observed rate of state change, improving responsiveness across different drift velocities Third, the binary harmfulness criterion can be replaced by a continuous risk-scoring model that integrates downstream task performance loss, enabling more nuanced model update scheduling. Fourth, the framework can be extended to handle recurring drift by maintaining a library of previously learned HMM emission models and using a model-selection criterion to match the current stream to a historical regime rather than always comparing to a fixed reference window. Fifth, the HDT system can be embedded into federated or edge-computing architectures for real-time drift monitoring in distributed sensor networks, where centralized retraining is infeasible Finally, the harmfulness and drift-type labels produced by HDT can serve as training signals for meta-learning approaches that learn to select the optimal adaptation strategy given the observed drift characteristics.

CRediT Author Contribution Statement

Bitan Misra: Conceptualization, Methodology, Supervision, Writing – Review & editing, **Kailas Kunjumon:** Data curation, Formal analysis, Investigation, Writing – original draft, **Aksa Maria Paul:** Investigation, Validation, Visualization, Writing – original draft, **Mariya K Joby:** Data curation, Resources, Validation, Writing – review & editing, **Hemanth KS:** Supervision, Project administration, Funding acquisition, Writing – review & editing. All authors have read and agreed to the published version of the manuscript.

Funding Declaration
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement
Data used in this manuscript is available at https://www.kaggle.com/datasets/orvile/gas-sensor-array-drift-dataset.

Conflict of Interest
There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure
The authors declare that artificial intelligence (AI)-assisted tools were used only for language refinement, grammar improvement, and manuscript structuring purposes during the preparation of this work. All technical content, experimental implementation, results, and interpretations were independently developed and verified by the authors.

Supporting Information
Not applicable

References

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under Concept Drift: A Review, *IEEE Transactions on Knowledge and Data Engineering*, 2019, **31**, 2346-2363, doi: 10.1109/TKDE.2018.2876857.

[2] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, *ACM Computing Surveys*, 2014, **46**, 37, doi:10.1145/2523813.

[3] I. Žliobaitė, M. Pechenizkiy, J. Gama, An Overview of Concept Drift Applications, In: Japkowicz, N., Stefanowski, J. (eds) Big Data Analysis: New Algorithms for a New Society, Studies in Big Data, 2016, Springer, Cham, **16**, doi: 10.1007/978-3-319-26989-4_4 .

[4] M. Baena-Garcıa, J. Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, in Fourth international workshop on knowledge discovery from data streams, 2006, **6**, 77-86.

[5] D. N. Assis, V. M. A. Souza, ADWIN-U: adaptive windowing for unsupervised drift detection on data streams: DN Assis, VMA Souza, *Knowledge and Information Systems*, 2025, 1-30, doi: 10.1007/s10115-025-02523-1.

[6] Z. Zhang, H. Zhang, An online transfer learning model for intrusion detection using FT-Transformer and KSWIN-driven concept drift detection mechanism, 2024 5th International Conference on Computer Engineering and Application (ICCEA), 2024, 128-131, doi: 10.1109/ICCEA62105.2024.10603489.

[7] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K. R. Müller, How to explain individual classification decisions, *The Journal of Machine Learning Research*, 2010, **11**, 1803-1831, doi: https://www.jmlr.org/papers/volume11/baehrens10a/baehrens10a.pdf.

[8] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K. R. Müller, Explaining nonlinear classification decisions with deep taylor decomposition, *Pattern recognition*, 2017, **65**, 211-222, doi: 10.1016/j.patcog.2016.11.008.

[9] J. Z. Kolter, M. A. Maloof, Dynamic weighted majority: An ensemble method for drifting concepts, *Journal of Machine Learning Research*, 2007, **8**, 2755–2790, doi: 10.1109/icdm.2003.1250911.

[10] W. Wang, Q. Chen, X. He, L. Tang, Cooperative anomaly detection with transfer learning-based hidden Markov model in virtualized network slicing, *IEEE Communications Letters*, 2019, **23**, 1534-1537, doi: 10.1109/LCOMM.2019.2923913.

[11] S. Greco, B. Vacchetti, D. Apiletti, T. Cerquitelli, Unsupervised concept drift detection from deep learning representations in real-time, *IEEE Transactions on Knowledge and Data Engineering*, 2025, **37**, 6232-6245, doi: 10.1109/TKDE.2025.3593123.

[12] V. Losing, B. Hammer, H. Wersing, Incremental on-line learning: A review and comparison of state-of-the-art algorithms, *Neurocomputing*, 2018, **275**, 1261-1274, doi: 10.1016/j.neucom.2017.06.084.

[13] J. Komorniczak, P. Ksieniewicz, Unsupervised concept drift detection based on parallel activations of neural network, In International Workshop on Discovering Drift Phenomena in Evolving Landscapes, Springer Nature, 2024, 111-127, doi: 10.1007/978-3-031-82346-6_8.

[14] L. M. Koch, C. F. Baumgartner, P. Berens, Distribution shift detection for the postmarket surveillance of medical AI algorithms: a retrospective simulation study, *NPJ Digital Medicine*, 2024, **7**, 120, doi: 10.1038/s41746-024-01085-w.

[15] L. Xiong, M. Wang, Z. Zhu, Me. He, Y. Hou, X. Tang, Semi-supervised comparative learning

compensation method for chemical gas sensor drift, *Analytical and Bioanalytical Chemistry*, 2024, **416**, 6827-6838, doi: 10.1007/s00216-024-05577-2.

[16] L. E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *The annals of mathematical statistics*, 1970, **41**, 164-171, doi: 10.1214/aoms/1177697196.

[17] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 1989, **77**, 257-286, doi: 10.1109/5.18626.

[18] Gas Sensor Array Drift Dataset, 2012, https://archive.ics.uci.edu/dataset/224/gas+sensor+array+drift+dataset, Accessed 18 May 2026.

[19] T. M. T. Pham, K. Premkumar, M. Naili, J. Yang, Time to retrain? Detecting concept drifts in machine learning systems, 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2025, 260-271, doi: 10.1109/ICSE-SEIP66354.2025.00029.

[20] G. Hovakimyan, J. M. Bravo, Evolving strategies in machine learning: a systematic review of concept drift detection, *Information*, 2024, **15**, 786, doi: 10.3390/info15120786.

[21] S. S. Bagui, M. P. Khan, C. Valmyr, S. C. Bagui, D. Mink, Model retraining upon concept drift detection in network traffic big data. *Future internet*, 2025, **17**, 328, doi: 10.3390/fi17080328.

[22] F. Liu, Z. Liu, X. Zhang, Z. Liu, Y. Liu, Machine learning-enhanced metal oxide gas sensor arrays: a review of selectivity and performance improvements, *Sensor Review*, 2025, **46**, 629–647, doi: 10.1108/sr-08-2025-0639.

[23] L. M. Ferrero, Continual contrastive learning for non-stationary data streams in autonomous systems, *Journal of Multidisciplinary Knowledge*, 2025, **5**, 191-200, doi: 10.36676/jmk.v5.i2.97.

[24] H. Mei, J. Peng, T. Wang, B. Zhang, F. Xuan, A semi-supervised domain adaptation for cross-gas concentration prediction with multi-target and bidirectional transfer for e-nose system, *Sensors and Actuators B: Chemical*, 2025, **450**,139242, doi: 10.1016/j.snb.2025.139242.

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