IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 March 2024, accepted 29 March 2024, date of publication 15 April 2024, date of current version 22 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3388529

RESEARCH ARTICLE

Unsupervised Tuning for Drift Detectors Using Change Detector Segmentation

RICARDO PETRI SILVA[1], SYLVIO BARBON JUNIOR[2], BRUNO BOGAZ ZARPELÃO[3], AND LEONIMER FLAVIO DE MELO[1]
[1]Department of Electrical Engineering, State University of Londrina, Londrina 86.057-970, Brazil
[2]Department of Engineering and Architecture, Università degli studi di Trieste, 34127 Trieste, Italy
[3]Department of Computer Science, State University of Londrina, Londrina, Brazil

Corresponding author: Sylvio Barbon Junior (sylvio.barbonjunior@units.it)

ABSTRACT Concept drifts can occur due to various factors such as changes in the environment or sensor degradation, posing significant challenges to machine learning systems by potentially skewing decision-making processes. Therefore, detecting drifts is essential to maintain the integrity and functionality of these systems. Automatic detectors based on statistical information are usually used to accomplish this task. Optimizing drift detectors through tuning is crucial for effective concept drift analysis. However, relying on user expertise or labeled data for supervised tuning can be challenging. In such cases, the use of unsupervised tuning methods becomes a suitable alternative to adapt detectors to evolving data distributions. To address this gap in the literature an unsupervised tuning method was proposed, leveraging a time series segmentation method. This innovative approach aims to alleviate the reliance on labeled data or user expertise traditionally associated with supervised methods, offering a more adaptive and automated means of tuning detectors. Our results demonstrate that our proposed approach outperforms the default configuration in most evaluated cases. Furthermore, we show that our approach can improve the hyperparameter tuning process when the type of drift is known including supervised methods such as Random Search tuning. By adopting our approach, we can achieve better performance in drift detection and improve the accuracy and reliability of systems that rely on this critical task.

INDEX TERMS Algorithm tuning, concept drift, signal drift, drift detector.

I. INTRODUCTION

A sequence of signals, commonly referred to as a time series, encapsulates representations of diverse natural phenomena [1], [2], [3], [4]. Signal analysis is crucial for understanding historical trends, current states, and formulating future predictions, organizing information sequentially while preserving its temporal structure.

When a signal deviates from its expected trajectory, a closer examination becomes necessary to determine if the deviation arises from an inherent signal phenomenon or external interference. Persistent deviations over time indicate a phenomenon known as Concept Drift (CD), meaning a fundamental change in the signal's behavior. In such cases, it is crucial to identify the exact moment when this CD occurred. To address this need, a methodology known as Drift Detection (DD) is employed [5], [6].

DDs are specialized algorithms designed to quickly and accurately identify CD instances in time-sensitive situations. These detectors assess key performance metrics of a signal, such as predictive accuracy or error rates, over consecutive time intervals. By comparing these metrics across different time frames, they detect statistically significant changes in the model's behavior, indicating the presence of CD. The deployment of DDs is crucial across various domains [7], [8], including signal processing [9], [10], [11], system monitoring [12], process control [13], data analysis [14], [15], and numerous other practical applications [16], [17].

Their main goal is to promptly identify substantial changes in signals, indicating drifts, anomalies, trends, or other critical events, to facilitate timely corrective actions. For instance, in a healthcare monitoring system [18], a DD

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

can quickly alert medical professionals to significant shifts in a patient's cardiac function, enabling swift intervention. In financial analysis, the ability to identify trends in the stock market [19] empowers investors to make well-informed decisions. Furthermore, DDs are invaluable in detecting fraudulent activities within financial systems [20], monitoring security systems [21], and various other applications.

There is a clear need in the literature for the optimization of DDs through tuning techniques that are independent of user knowledge [22], [23], [24]. Despite the advancements in this domain, the prevalent tuning techniques largely depend on supervised learning paradigms [25], which may not align with the practicalities of real-world implementations. The challenge of acquiring labeled data for the purpose of training and evaluating DDs presents a significant obstacle, rendering the process not only impractical but also resource-intensive. This limitation hinders detailed detector parameter optimization, potentially resulting in suboptimal detection and adaptive model performance. In this context, our research aims to address this autonomous tuning process. To achieve this goal, we propose an innovative unsupervised learning approach to fine-tune DDs.

This unsupervised learning approach yields numerous substantial advantages. Firstly, it amplifies the adaptability and performance of detectors across diverse data distributions which can be extended to real-world scenarios. The optimizer facilitates automatic fine-tuning of hyperparameters, alleviating the necessity for manual adjustments and expert intervention. This, in turn, broadens the applicability of the detectors. Furthermore, an efficient hyperparameter optimizer contributes to heightened generalization capabilities of DDs, ensuring robust performance on unseen data. This aspect assumes a significance in unsupervised settings where data distribution may dynamically evolve, requiring detectors to adjust seamlessly without labeled information. Additionally, an automated hyperparameter optimization framework helps mitigate overfitting or underfitting issues, thereby enhancing the overall reliability and accuracy of change detection models.

We hypothesize that relying solely on default values for DDs, commonly found in prevalent literature and libraries like scikit-multiflow [26] and river [27], might not always provide the optimal configuration. Building on this idea, our methodology aims to enhance this decision-making process by using an unsupervised approach for hyperparameter tuning. This approach is grounded in the Change Detector Segmentation (CDS) framework [28]. By leveraging the principles of the CDS framework, our unsupervised tuning strategy aims to surpass the limitations of default DDs hyperparameters. It seeks a more precise configuration that aligns with the dynamics of distribution shifts in signal data.

The unsupervised tuning method we propose begins by segmenting a signal into various parts, followed by performing a stationarity analysis on each segment. Within this framework, we implement three specific heuristics anchored in the principles of the Augmented Dickey-Fuller (ADF) test. These heuristics establish new rules for identifying CD by mapping the most change-prone and stable regions.

Given the nature of this unsupervised approach, which does not inherently provide validation for the identification of confirmed drift occurrences, we undertook a validation process using a synthetic signal dataset. This dataset was generated by the Configurable Signal Generator (CSG), a novel tool introduced in our study. The CSG addresses a notable void in the landscape of signal generation tools, offering a mechanism for generating signals with precisely defined change points. This capability ensures that the signal generation process is fully configurable, enabling accurate simulation and analysis of various signal behavior scenarios.

To fulfills a gap in existing signal generators by providing precise information about where changes in the signal occur, ensuring fully configurable signal generation to confirm drift occurrences, we propose the Configurable Signal Generator (CSG). The artificial signals produced by the CSG are instrumental in evaluating the effectiveness of our approach, offering a controlled environment to simulate various scenarios of concept drift. This allows for a thorough examination and confirmation of the unsupervised tuning's capability to adaptively identify and respond to changes

The paper includes a performance comparison of commonly used DDs, demonstrating that our proposed tuning process generally surpasses the default configuration in most cases. These findings advocate for the wide-scale adoption of our approach in diverse real-world contexts, especially in online drift detection scenarios such as data streams [29]. Additionally, we observed instances where our unsupervised tuning outperforms supervised approaches like Random Search [30]. It is important to emphasize that up to the time when this study was conducted, no other unsupervised tuning technique for DDs had been found.

The main contributions of this work are:

- An unsupervised approach for tuning DD hyperparameters.
- A review of hyperparameter optimization techniques for DDs.
- Identification of the most important factors for identifying drifts in signals.
- The CSG, an artificial signal generator, for labeled drift ocurrences within a signal.

The paper follows a structured approach. Section II lays the groundwork by introducing essential definitions and foundational concepts. Moving to Section III, we delve into related work, highlighting the evaluated DDs and the tuning process. In Section IV, we unveil our primary contribution: an unsupervised approach for tuning DD configuration. Section V details the materials and methods and the specific configuration applied in our experiments. Within Section VI, we present our experimental findings in detail, offering a comprehensive analysis. Finally, Section VII concludes the paper by summarizing our discoveries and pointing towards potential future research directions.

II. PRELIMINARIES

We begin by defining DDs. Next, we explore the crucial relationship between the tuning process and stationarity, a significant concept in this study. Understanding these definitions is pivotal for grasping our forthcoming proposal. Stationary signals possess specific statistical properties, which the tuning process leverages to enhance DDs accuracy and resilience in change detection. This process entails a meticulous analysis of the input signal's statistical properties and the selection of optimal hyperparameters for the DD. This fine-tuning enables the DD to respond more effectively to dynamic changes.

A. DRIFT DETECTOR

DD is commonly employed in handling data streams, which consist of an infinite sequence of data denoted as $S = (x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t), \ldots$. Here, each instance is represented by (x_t, y_t), where x_t represents a d-dimensional vector arriving at time stamp t, and y_t denotes the class label of x_t [31].

For univariate time series, the DD algorithm analyzes each sample X of the time series at time t. It anticipates that a given point X_{t+n}, with n as a time shift, should have a distribution similar to the point X_t. Any deviation prompts an alarm. However, in unsupervised approaches, lacking y_t information complicates the DD process significantly.

DDs analyze past data patterns to predict potential changes. They are commonly deployed to detect CD, which means alterations in data behavior over time [19]. Such changes are frequent in data stream analysis [32]. Calibrating these detectors involves selecting hyperparameters that align best with data behavior, often employing techniques for enhancement.

This study seeks to enhance the process of drift detection by leveraging insights from the analysis of time series stationarity. A stationary time series is characterized by consistent statistical properties over time, such as mean and variance, implying that intervals exhibiting less stationarity are more likely to be fertile grounds for identifying detection points by DDs.

B. TUNING

Tuning in machine learning involves systematically optimizing a model's hyperparameters to enhance its performance on a specific task or dataset [33]. Hyperparameters are preset configuration settings, not learned from the data, that govern the model's behavior and characteristics. They include parameters like learning rate, regularization strength, network architecture, and kernel parameters, among others.

The tuning process entails exploring a predefined hyperparameter space to discover the value combination that maximizes a selected evaluation metric, like accuracy, precision, recall, or F1 score. Techniques for this search include Grid Search, Random Search, Bayesian optimization, and gradient-based optimization methods [34].

During tuning, several iterations occur, evaluating various hyperparameter settings by training and testing the model on a validation dataset. Each configuration's performance is measured using the selected evaluation metric, and the set of hyperparameters that yields the best performance is chosen.

Tuning is an iterative and computationally demanding process, involving repeated model training and evaluation for different hyperparameter combinations. Yet, fine-tuning the hyperparameters enhances the model's generalization, performance, and robustness to unseen data.

C. STATIONARITY

In the context of random processes [35], a discrete-time or continuous-time random process $X(t)$ is deemed stationary if the joint distribution of any set of samples remains unchanged regardless of the time origin. Put simply, the joint cumulative distribution function of $X(t_1)$, $X(t_2)$,..., $X(t_k)$ remains the same as that of $X(t_1 + \tau)$, $X(t_2 + \tau)$,..., $X(t_k + \tau)$ for all time shifts τ, all k, and all choices of sample times $t_1, \ldots, t_k$.

In signal analysis, stationarity refers to a signal's inherent stability over time, implying consistent statistical properties. trends can disrupt this stability, rendering the signal non-stationary. A signal's temporal profile, marked by alternating periods of stability, may be interrupted by trend changes that alter its behavior. These fluctuations, prompted by trend shifts, challenge the signal's stationarity, revealing a dynamic interplay between stability and variability. This interplay underscores the complex dynamics of signal behavior.

The effectiveness of DDs relies on their capacity to adapt accurately to CD in data signals. Leveraging stationarity-based techniques allows us to detect shifts in statistical properties, enhancing the robustness and adaptability of DDs. This method ensures that DDs can address evolving data distributions, aligning with the broader goal of achieving superior performance in dynamic data environments.

III. RELATED WORK

This section provides a literature review of commonly used DDs and their particularities. In many DD applications, hyperparameters often receive less attention compared to the obtained results. Furthermore, this section reviews studies focusing on tuning processes for drift identification.

A significant study, outlined in [36], performed a comparative analysis of 14 different DDs. This investigation evaluated their accuracy in detecting actual drift instances within the dataset. Among the assessed detectors were HDDM_{A}, Fisher Test Drift Detector (FTDD), Wilcoxon Rank Sum Test Drift Detector (WSTD), FHDDM, and HDDM_{W}. An unexplored gap in this study lies in the absence of tuning techniques for the DDs, given that all were compared with their default configurations.

Santos et al. [37] discuss the concept of data streams and CD, where the distribution of data changes over time. Several approaches have been proposed to address CD, including using ensembles, focusing on recurring concepts, and CD detection methods. Various DD methods have been proposed, each with its own hyperparameters, and optimal values vary depending on the datasets used. The paper introduces a genetic algorithm designed to optimize DD hyperparameters. This approach aims to identify suitable configurations adaptable to diverse scenarios, facilitating a fair comparison of methods by adjusting hyperparameter values based on the dataset characteristics utilized in the experiments.

Gemaque et al. [38] introduced a taxonomy for unsupervised CD approaches via a systematic literature review. This review revealed two primary categories within recent scientific publications on CD detection: articles encompassing surveys and analyses of diverse CD approaches, and articles proposing unsupervised CD methods. The unsupervised and semi-supervised drift detection methods were further classified into batch-based and online-based methods. Upon analyzing the batch-based methods, it was noted that their primary distinction lies in whether they monitor significant data distribution changes across the entire arriving batch or solely on a selected instance set. The online-based methods were also examined, subcategorized based on their approach to comparing the two involved windows during drift detection.

Agrahari and Singh [39] discuss data streams and the need to analyze data near real-time. Data stream mining involves detecting changes in non-stationary data distribution, which is a major challenge. Data points in a stream are generated sequentially and independently under some probability distribution. To effectively mine data streams, examples must be processed only once, learned under memory restrictions, processed in a limited amount of time, and ready to predict class labels on demand. In the context of data stream mining, one of the biggest challenges is dealing with semi-supervised or unsupervised data, which often have very few or no labeled data available for training. Or even solutions based on methods such as Meta-Learning and Active Learning [23]. This lack of labeled data makes it difficult to train supervised learning models and demands the use of unsupervised or semi-supervised methods. Therefore, developing effective unsupervised and semi-supervised methods for data stream mining is crucial for making accurate predictions and discovering valuable insights in these challenging scenarios.

Some reviews focus on the study of DDs in the streaming data environment [40]. One of the main challenges in this context is the scarcity of labeled data due to the large volume of data processed in a short period of time. However, the accuracy of the analysis requires labeled data. Therefore, there is a need to develop unsupervised or semi-supervised algorithms to address this issue and achieve high accuracy.

In the domain of Hyperparameter Optimization [41], the Random Search approach has been extensively studied [42], [52]. Nevertheless, it is imperative to underscore that these methodologies predominantly operate in a supervised context, requiring validation through comparisons with labeled datasets.

Upon a thorough analysis, a significant gap emerges in the realm of hyperparameter tuning for DDs. Notably, prior research in this domain has largely overlooked the exploration of unsupervised tuning methods. Addressing this research gap, this study introduces an unsupervised approach for tuning hyperparameters. As far as our knowledge extends, our approach appears to be the first to explicitly address these concepts. Several works identified in the literature utilize supervised approaches for hyperparameter tuning. However, the primary novelty of this study lies in the autonomous optimization process, eliminating the necessity for user intervention or prior domain knowledge.

IV. PROPOSED APPROACH

Our approach is oriented towards achieving unsupervised tuning for DDs. To fulfill this objective, we propose a novel combination of stationarity criteria derived from the Augmented Dickey-Fuller (ADF) test. At the core of our methodology lies an unsupervised framework called Change Detector Segmentation (CDS), which adjust the inherent hyperparameters of DD. Importantly, the CDS framework was initially introduced to eliminate non-stationary samples in time series data. In this study, we adapt this framework to apply to any signal type, shifting its focus from segmenting samples to accurately identifying them as drifts.

Unsupervised tuning methods eliminate the need for annotated data, significantly reducing the time and resources required for data labeling. This crucial advantage speeds up the optimization process and extends the applicability of tuning to contexts with limited availability or high costs linked to acquiring labeled data, as seen in online scenarios. This addresses a key concern in machine learning, where access to abundant labeled data is often limited, offering a promising opportunity to improve model performance under resource constraints.

We present a comprehensive framework designed for unsupervised tuning in signal. Our proposed approach encompasses five key stages based on the CDS framework, adeptly handling the complex challenge of precisely identifying drift occurrences while optimizing the hyperparameters governing DDs. The following steps outline the methodology used in this study, enabling replication of the process:

A. STEP 1

Our investigation starts with the acquisition of a signal, predominantly presented as a time series structure, serving as the basis for evaluating and refining our approach. Within the signal's temporal evolution, CD become points of focus, triggering subsequent steps in our process.

B. STEP 2

Step 2 encompasses segmenting the signal acquired in Step 1, which can be executed in two manners:

1) For this study, we chose to divide the signal empirically into 10 segments. This approach is advisable when the user possesses limited prior knowledge of the signal.

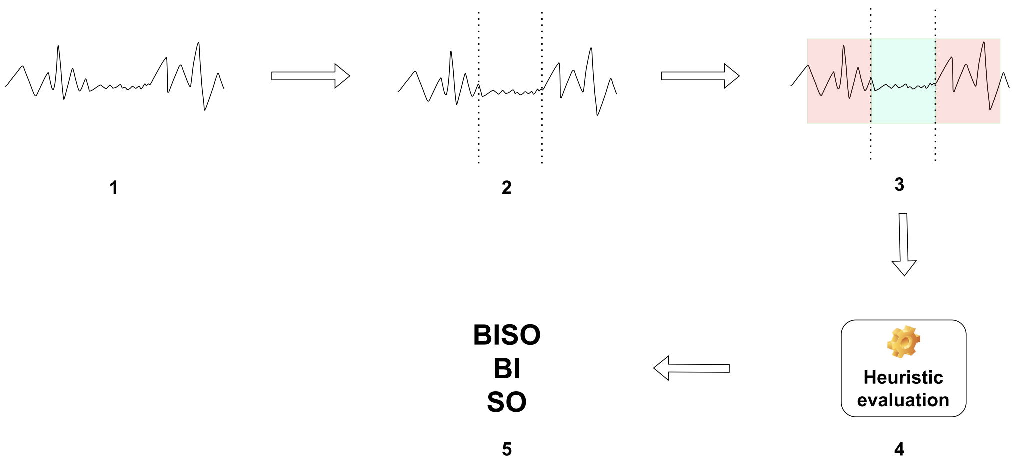


FIGURE 1. Unsupervised Tuning Methodology.

2) Alternatively, the signal can be segmented based on its seasonality and requires knowledge of the signal's behavior by the user. Seasonality refers to recurring cycles within the signal that display consistent behaviors over time.

Segmenting the signal serves a purpose: analyzing the entire signal may complicate statistical information analysis, reducing sensitivity to drifts. Utilizing smaller segments enhances sensitivity, facilitating drift detection.

C. STEP 3

Step 3 defines the critical intervals, highlighted in red in Fig. 1, alongside intervals indicating normal behavior, marked in green. This distinction is established via the stationarity test performed by the ADF test. To achieve this, the following procedures are implemented:

1) Calculate the stationarity statistical value for the entire signal.
2) Calculate the stationarity statistical value for each segment.
3) Identify critical intervals as segments with a stationarity statistical value higher than that obtained in Step 1.

We propose that regions with higher stationarity are less likely to exhibit drifts compared to regions with lower stationarity. Thus, we define critical intervals of the signal by their lower stationarity.

D. STEP 4

The preceding steps are crucial for implementing unsupervised tuning. Moving on to Step 4, we initiate our Heuristic evaluation. The intervals derived from the segmented signal guide us in configuring any DD appropriately. At this stage, users can set a range of values for each DD hyperparameter and employ our heuristics to identify the best choice. Alternatively, users can use values empirically tested in this study. Specifically, we explore four decay and growth percentages relative to each detector's default values. The decay percentages are 95, 90, 75, 50, and 10, while the growth percentages are 105, 110, 125, 150, and 1000. Our aim is to categorize values showing slight and significant deviations from default values to improve signal generalization.

For example, if a specific hyperparameter has a default value of 10, the values used in this study to explore a wider range of configurations will be as follows:

- Decay values: 9.5, 9, 7.5, 5, and 1.
- Growth values: 10.5, 11, 12.5, 15, and 100.

These values represent adjustments to the default value for the hyperparameter under consideration.

E. STEP 5

Step 5 employs our heuristics to select the optimal hyperparameters, offering a specific combination that satisfies the following definitions:

- Bigger in Smaller out (BISO): This heuristic selects the hyperparameter configuration that detects the highest number of drifts within the critical intervals while minimizing detections outside of these intervals. BISO aims for balanced sensitivity, focusing on placing the majority of drifts within the critical intervals while reducing detections in non-critical intervals. Consequently, BISO is expected to effectively identify both abrupt and gradual drifts.
- Bigger in (BI): This heuristic aims to select the hyperparameter configuration that detects the highest number of drifts within the critical intervals. BI is more sensitive, potentially generating more false positives compared to BISO. BI primarily excels in identifying abrupt drifts. BI should exhibit efficacy in identifying abrupt drifts.
- Smaller out (SO): In contrast to the previous heuristics, SO prioritizes the hyperparameter configuration that detects the lowest number of drifts outside the critical intervals. However, its selective nature increases the possibility of false negatives due to filtering out minor variations.

Our proposal orchestrates a methodical process that combines unsupervised tuning and drift identification within a signal. Through this innovative framework, we aim to tackle the challenges inherent in drift detection, thereby advancing unsupervised learning paradigms.

Additionally, our approach outlines a systematic steps integrating unsupervised tuning and drift identification within a signal, shedding light on the complexities associated with drifts. It offers a solution when users lack prior knowledge about the analyzed signal, allowing them to explore hyperparameter configurations surpassing the default settings.

It is crucial to emphasize that our unsupervised tuning extends its applicability to a wide range of DDs, regardless of their specific characteristics.

V. MATERIALS AND METHODS

To validate our proposed approach in this study, it is crucial to determine if our unsupervised tuning proves to be a feasible alternative compared to the default configuration of DDs. This validation hinges on demonstrating that our proposed approach consistently outperforms the default configuration. To accomplish this, we will outline the details of our experimental setup.

To empirically study drift phenomena, it was essential to create a reliable and carefully annotated synthetic database. In this regard, the Configurable Signal Generator played a crucial role, facilitating the generation of artificial signals tailored to populate our designated database. This deliberate use of synthetic data injection was intended to cultivate a dataset characterized by the necessary qualities of precision and reliability, thus providing an ideal basis for the systematic execution of our experimental investigations.

To conduct the experiment, acquiring signals with labeled CD samples—representing normal and CD-specific instances—was indispensable. However, the absence of tools providing such information was evident. Consequently, this work developed the Configurable Signal Generator, ensuring the mentioned specifications.

A. CONFIGURABLE SIGNAL GENERATOR (CSG)

Upon a comprehensive review of the current state of available datasets, it becomes apparent that the majority lack sufficient instances of concept drift or fail to provide proper labels indicating the occurrence of such shifts, as discussed in [53]. This deficiency presents a substantial obstacle for researchers and practitioners aiming to develop and evaluate models that can effectively handle concept drifts in real-world scenarios.

Labeled datasets play a pivotal role in training and validating machine learning models. However, in the context of CD, the scarcity of labeled data reflecting dynamic changes in the underlying data distribution hampers the progress in developing adaptive and resilient models. The absence of such datasets limits our ability to assess the generalization and adaptability of models to changing environments.

To bridge this gap, the CSG, introduced in this study, serves as an innovative tool designed to simplify the creation of synthetic signals with adjustable hyperparameters and labeled samples. Its purpose is to enable the generation of signals with diverse statistical distributions. The main goal is to empower users by providing precise control over the signal's amplitude and the introduction of perturbations based on their preferences.

This signal generator operates based on five key hyperparameters, each for customizing the artificial signal to meet specific requirements:

1) Number of Samples: this hyperparameter enables users to specify the granularity of the generated signal by defining the number of discrete data points or samples. This configuration is important to achieve the desired temporal or spatial resolution in the resulting signal.
2) Initial and Final Amplitude Values: this feature provides users with the flexibility to set the starting and ending amplitude levels of the signal. It is instrumental in modulating the signal's intensity over a defined time span or spatial region. Additionally, it facilitates the creation of signals with varying amplitudes, such as linear ramps or amplitude-modulated signals.
3) Minimum and Maximum Perturbation: perturbations introduce deviations from the ideal or nominal signal, incorporating stochastic elements into the generated data. By specifying the minimum and maximum perturbation values, users have precise control over the level of randomness or noise added to the signal. This capability allows for the creation of signals that simulate real-world CD scenarios with inherent variability and uncertainty.

Distinguished from prevailing signal generators [43], the distinctiveness of the CSG arises from its ability to introduce stochastic perturbance at various junctures within the signal's trajectory. Notably, the intervals demarcating these instances of perturbation are recorded and provided as an output, rendering the CSG an invaluable asset within the signal generation paradigm.

The CSG sets itself apart from existing signal generators by its unique capability to introduce labeled random disturbances at different points along the signal's path. What makes it particularly noteworthy is that it records and outputs the specific intervals where these disturbances occur. Other signal generators commonly lack the explicit demarcation of regions where perturbations are introduced. This drawback becomes especially significant when trying to pinpoint the exact areas in the signal's path that have encountered perturbations, hindering a detailed understanding of the perturbation dynamics. This feature makes the CSG an invaluable asset in the field of signal generation.

A pseudocode outlining the implementation of the CSG is available in Algorithm 1. The function returns the generated signal and a list of indices corresponding to the positions where perturbations occurred, simulating CDs in this generator.

Algorithm 1 Generate Random Stream

```
procedure GenerateRandomStream
    input: samples,            ▷ represents the desired number of samples to generate for the signal,, e.g.: 10.
    initial_amplitude,         ▷ indicates the initial value of the signal's amplitude, e.g.: 10.
    final_amplitude,           ▷ indicates the final value of the signal's amplitude,, e.g.: 100.
    min_perturbation,          ▷ represents the minimum perturbation value to be applied,, e.g.: 1.
    max_perturbation           ▷ indicates the maximum perturbation value to be applied,, e.g.: 10
    output: signal,            ▷ represents the signal generated possibly containing perturbations.
    perturbation_indices       ▷ list of indices labeled as perturbations along the signal.

    signal ← initial_amplitude + random(samples) * final_amplitude
    intervals ← obtain_intervals(signal)
    perturbation_indices ← []
    for i in range(intervals) do
        perturbation ← min_perturbation + random(samples) * max_perturbation
        signal[i] ← signal[i] * perturbation
        perturbation_indices.add(i)
    end for
    return signal, perturbation_indices
end procedure
```

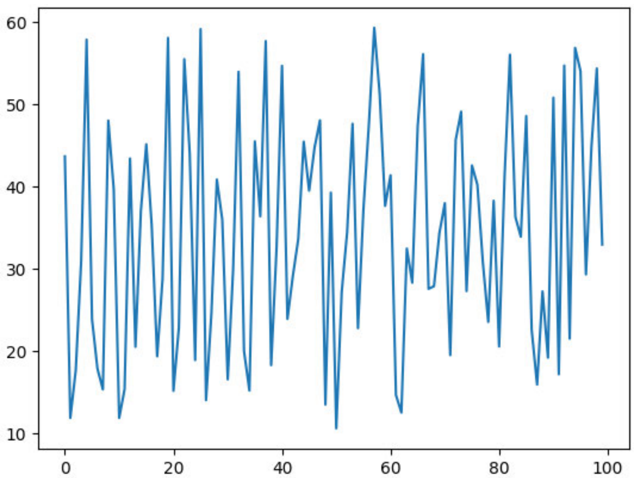


FIGURE 2. Random signal generated from CSG. Initial amplitude = 10, final amplitude = 60.

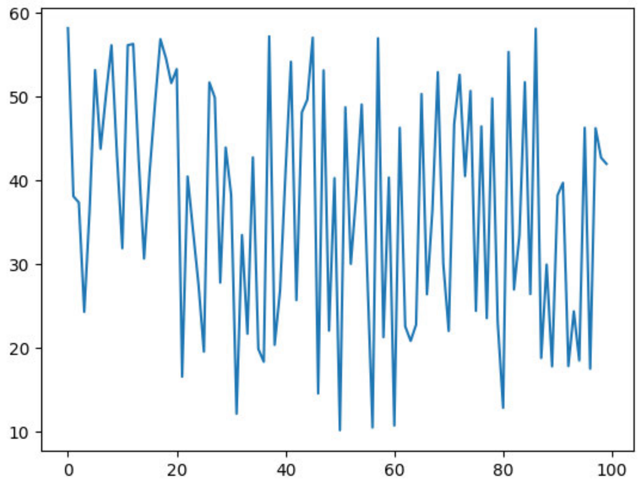


FIGURE 3. Random signal generated from CSG. Initial amplitude = 10, final amplitude = 60.

In this study, we consider the points where perturbations were introduced as representations of CD. These marked points will act as indicators, enabling us to evaluate whether our unsupervised tuning approach can surpass the default DD configuration.

Figs. 2 and 3 depict outputs generated by the CSG. In these figures, it is evident that each generated signal exhibits a random behavior with distinct characteristics. This simulation aims to emulate the representation of real-world data, capturing the inherent variability and unpredictability often observed in practical datasets.

B. EXPERIMENTAL SETUP

We utilized various configurations during signal generation to mimic natural signals and demonstrate the effectiveness of the proposed unsupervised tuning approach for DDs. Table 1 showcases the diverse experimental setups involving the generation of random perturbations for each signal. The Signal column denotes the number of signals used for the experiments. Each signal is generated with random distributions. The Size column indicates the sample sizes for the signals, set at 100, 1000, and 5000 samples, respectively.

Additionally, the Amplitude column details the amplitude variation for each configuration. This variation is calculated by determining the difference between the final and initial amplitudes set during signal creation. The Perturbation column quantifies the degree to which the signal's samples were perturbed, with higher values indicating increased variability. Here, perturbation values of 1.5 and 2 are considered as low variations, while values of 4 and 10 are categorized as high

TABLE 1. Signal generation configurations.

	Signal	Size	Amplitude	Perturbation
A	10	[100, 1000, 5000]	10	1
B	10	[100, 1000, 5000]	10	1.5
C	10	[100, 1000, 5000]	10	2
D	10	[100, 1000, 5000]	10	4
E	10	[100, 1000, 5000]	10	10
F	10	[100, 1000, 5000]	30	1
G	10	[100, 1000, 5000]	30	1.5
H	10	[100, 1000, 5000]	30	2
I	10	[100, 1000, 5000]	30	4
J	10	[100, 1000, 5000]	30	10
K	10	[100, 1000, 5000]	90	1
L	10	[100, 1000, 5000]	90	1.5
M	10	[100, 1000, 5000]	90	2
N	10	[100, 1000, 5000]	90	4
O	10	[100, 1000, 5000]	90	10

variations. A perturbation value of 1 means signals without any perturbations.

To assess the efficacy of the unsupervised tuning process, we evaluated various DDs: PH, ADWIN, DDM, $HDDM_A$, $HDDM_W$, and KSWIN. Our evaluation aimed to validate the performance of the default configuration by examining the accuracy of each configuration in identifying drifts. Initially, we visualized the Correct Detection of each DD for every configuration using boxplots. Next, we analyzed how detection was influenced by the number of samples in the signal. Finally, we employed Friedman's statistical test to determine any significant statistical differences between the techniques.

We used the default configuration as our starting point. Additionally, we conducted a comparative analysis against the Random Search (RS) tuning strategy, which acts as a fundamental benchmark. RS aims to identify the best hyperparameter configuration within a predefined range by employing a random selection process rather than exhaustively exploring all possible configurations [30]. It is important to note that RS assumes a supervised context and may not thoroughly explore the entire configuration space.

C. DRIFT DETECTORS

The selected DDs for experimentation in this study will be delineated herein, with particular emphasis on the hyperparameters associated with each of them, which will be evaluated during the unsupervised tuning process.

1) PAGE-HINKLEY

The Page-Hinkley (PH) detector operates by maintaining simultaneous averages of the ongoing analysis moment, focusing solely on capturing detections without necessitating alert generation. It leverages the Page-Hinkley test [44] to discern deviations within the data. The pivotal determiner in the drift detection process is the parameter lambda, serving as a threshold value. The instantiation of a detection is prompted when the prevailing mean surpasses the stipulated lambda value, thus activating the detection mechanism.

2) PAGE-HINKLEY HYPERPARAMETERS

There are several key hyperparameters to consider in PH detector. First, the hyperparameter m_i refers to the minimum number of instances that must be evaluated before the detector can accurately identify a change. This hyperparameter is critical for ensuring that the detector is reliable and produces accurate results. Another important hyperparameter is delta (δ), which represents the delta factor for the Page-Hinkley test. This hyperparameter plays a key role in determining the sensitivity of the detector to changes in the data. The change detection threshold (t) is another critical hyperparameter in the PH detector. This threshold is used to determine when a change has occurred in the data, based on the magnitude of the change relative to the threshold. Finally, the alpha (α) hyperparameter is the forgetting factor used to weight the observed value and the mean. This hyperparameter is important for ensuring that the detector can adapt to changes in the data over time, while still maintaining a stable baseline for comparison. By carefully adjusting these hyperparameters, it is possible to build a PH detector that is highly effective at detecting changes in a wide range of data sets.

3) ADWIN

ADWIN is a change detection algorithm proposed by Bifet and Gavalda [45]. Its objective is to monitor a data stream and maintain statistical information about it. The algorithm uses a sliding window approach, where small portions of the data stream are analyzed. The size of the window is predefined, and statistical information such as the mean and variance are calculated for each window. The ADWIN algorithm then compares the mean of two adjacent windows, and if the difference between the two means is greater than a predefined threshold value referred to as delta (δ), it is considered as a change point or drift in the data stream. ADWIN does not issue alerts for every change detected but rather keeps track of the changes and adjusts the window size accordingly to adapt to the new data distribution.

4) ADWIN HYPERPARAMETERS

In the assessment of ADWIN, the focus was directed solely towards the δ hyperparameter, meaning the confidence level required for initiating a contraction of the sliding window size. The complementary hyperparameter associated with the sliding window dimensions, denoted as W, was not subject to explicit tuning. This restraint was exercised owing to ADWIN's intrinsic capacity for autonomous adaptation, ensuring the real-time adjustment of the window size in response to the continuous influx of data.

5) DDM

To detect significant changes in data patterns, data monitoring systems frequently use error rate and standard deviation analysis [7]. When the error rate increases beyond a certain threshold, it is assumed that a change has been detected.

This approach is known as the Drift Detection Method (DDM). The DDM is commonly used in machine learning applications to track model performance and detect changes in data distribution over time.

6) DDM HYPERPARAMETERS

The DDM detector has a hyperparameter denoted as m_i, which specifies the minimum number of samples that need to be analyzed before a change can be detected. In addition to m_i, DDM also has two other hyperparameters, w_l and o_{cl}, which adjust the confidence levels for the warning and out-of-control signals, respectively. Specifically, w_l sets the threshold for raising a warning signal, while o_{cl} sets the threshold for raising an out-of-control signal.

7) HDDM_A AND HDDM_W

Both HDDM_A and HDDM_W detectors [46], [47] are based on Hoeffding's bounds inequality, which is a statistical theory used to determine the minimum number of observations needed to estimate a distribution. These detectors estimate CD using different statistics.

HDDM_A estimates CD through the mean and has three possible output states: stable, warning, and drift. The drift state is the most relevant because it indicates a significant change in the data stream. HDDM_W, on the other hand, uses the Exponentially Weighted Moving Average (EWMA) statistic as a change estimator. Like HDDM_A, it also has three possible output states: stable, warning, and drift.

8) HDDM_A AND HDDM_W HYPERPARAMETERS

The HDDM_A and HDDM_Adetectors have similar parameterizations, with one notable difference: HDDM_A has an additional hyperparameter called lambda option (l_o), which controls the weight given to recent data. Specifically, smaller values of l_o correspond to less weight given to recent data. Both detectors share two hyperparameters: d_c, which sets the confidence level for detecting a drift, and w_c, which sets the confidence level for issuing a warning.

9) KSWIN

KSWIN is a change detector proposed by Kolmogorov-Smirnov, named after its basis in the Kolmogorov-Smirnov (KS) statistical test [48]. It operates using a sliding window, hence the name KSWIN. The detector identifies CD by comparing the distance of the empirical cumulative data distribution of different statistical windows with the last identified CD.

Santos et al. [49] employed a differential evolution algorithm, a technique that emulates the principles of natural selection to optimize solutions in multidimensional problem spaces, in order to derive optimal configurations for DDs. Their investigation encompassed an array of detectors, leading to the observation that the DDM in conjunction with the Naive Bayes classifier exhibited superior performance. The research additionally presented a comprehensive tabular representation enumerating the hyperparameters associated with each individual detector.

To enhance the clarity of hyperparameterization patterns across detectors, a refinement was introduced to Table 2 based on the insights garnered from Santos et al.'s work. The modified rendition of the table has been thoughtfully curated to exclusively encompass the detectors examined within this current study. This strategic choice ensures that the pertinent hyperparameter details are highlighted without undue complexity, thereby facilitating a succinct and illuminating overview of the studied detectors' configurations.

The "range variation" column encompasses a spectrum of values that were meticulously assessed during the course of the unsupervised tuning process. This process involved an exhaustive exploration of potential values within this specified range to discern the optimal hyperparameter configuration for each detector. The ultimate values derived from the tuning process were rounded to ensure the detector's operation was not compromised.

D. EVALUATION METRICS

The experimental validation used metrics to evaluate DD performance. One of these metrics, "Correct Detection", inspired by the confusion matrix [50], calculates the difference between the number of accurately identified CD (True Positives - TP) and the non-drift samples mistakenly classified as CD (False Positives - FP). This difference is divided by the total number of CD in the analyzed signal, represented as N. Equation 1 outlines the calculation for this value.

$$\text{Correct Detection} = \frac{TP - FP}{N} \times 100 \tag{1}$$

Other metrics such as Accuracy were not a valid choice due to the natural imbalance in this application, where the amount of normal signals far outweighs those classified as CDs. Precision was also deemed inappropriate since only TP with FP would be evaluated, without taking into account the actual quantity of CDs present in the signal. Similarly, conventional metrics like Recall suffer from the same limitation, as they only consider TP with False Negatives (FN).

It is important to note that the outcome in Equation 1 is presented as a percentage. When the number of FP equals or surpasses the TP, the result becomes 0%. Also, the parameter N represents the count of intervals with perturbations in the analyzed signal. TP indicates correct detections where the DDs identified a drift within intervals genuinely containing a perturbation. It is relevant to mention that each interval receives only one count, regardless of the number of detections within it.

We also used Friedman's statistical test alongside the Nemenyi post-hoc test to evaluate the statistical differences between the different configurations of DDs. The Friedman's test provides an unbiased evaluation of the techniques by considering both mean differences and ranking order. This is crucial when performance metrics show substantial variation

TABLE 2. Drift detector parameter values.

Detector	p1 default (range variation)	p2 default (range variation)	p3 default (range variation)	p4 default (range variation)
PH	$m_i = 30$ [3, 300]	$\delta = 0.005$ [0.0005, 0.05]	$t = 50$ [5, 500]	$\alpha = 0.9999$ [0.99999, 0.999]
ADWIN	$\delta = 0.002$ [0.0002, 0.02]	N/A	N/A	N/A
DDM	$m_i = 30$ [3, 300]	$w_l = 2.0$ [0.2, 20]	$o_{cl} = 3.0$ [0.3, 30]	N/A
HDDM$_\mathbf{A}$	$d = 0.001$ [0.0001, 0.01]	$w = 0.005$ [0.0005, 0.05]	$t = 1$ [0.1, 10]	N/A
HDDM$_\mathbf{W}$	$d = 0.001$ [0.0001, 0.01]	$w = 0.005$ [0.0005, 0.05]	$t = 1$ [0.1, 10]	$\lambda = 0.05$ [0.005, 0.5]
KSWIN	$\alpha = 0.005$ [0.0005, 0.05]	$w = 100$ [10, 1000]	$ss = 30$ [3, 300]	N/A

across techniques. Additionally, it determines if there are statistically significant differences in performance between the techniques.

E. UNSUPERVISED TUNING ALGORITHM

To enhance the comprehension of our proposed approach, Algorithm 2 outlines the necessary steps to conduct the unsupervised tuning process for any DD that incorporates a set of hyperparameters. In Algorithm 2, the *CDS* method is applied to our adaptation of the Change Detector Segmentation framework, allowing us to obtain critical intervals in the signal. Subsequently, the *drift_detector* method generalizes the application of our unsupervised approach to any Drift Detector (DD). Following this, we derive the heuristics *BISO*, *BI*, and *SO* through the *compare_drift* method based on the DD detections and the assessment of critical intervals. Finally, our Correct Detection metric is calculated based on the accuracy of the detections according to our proposed architecture. The output of this method is precisely the heuristic with the configurations of BISO, BI, and SO, along with the Correct Detection metric for each of them.

It is essential to underscore that the functionality of our unsupervised tuning proposal does not rely on the input *perturbation_indices*, as it is designed to operate effectively on unlabeled datasets. The inclusion of *perturbation_indices* in this pseudocode is solely for the purpose of validating the metric Correct Detection that evaluates the performance viability of the technique.

VI. RESULTS AND DISCUSSION

This study aims to address specific Research Questions (RQ) concerning signal drift identification via unsupervised tuning. The RQs are outlined as follows:

1) Can unsupervised tuning for DDs based on stationary analysis improve the drift identification process?
2) What is the most impactful factor in detecting drifts in signals?

To address RQ1, we assessed six different DDs and illustrate the outcomes for four of them in Fig. 4 to Fig. 7 using a boxplot analysis. These figures portray the Correct Detection level achieved by each hyperparameter configuration in drift detection. Higher Correct Detection percentages mean superior performance. The ADWIN and DDM detectors were omitted from the analysis as they consistently yielded a 0% result, regardless of tuning. The aim of these evaluations is to analyze various signal behaviors amid concept drift presence.

Figs. 4 and 5 present outcomes associated with low amplitude variations, representing items A to J in Table 1. Fig. 4 displays results without perturbations in the signal, pertaining to items A and F in Table 1. In contrast, Fig. 5 includes perturbations, corresponding to items B, C, D, E, G, H, I, J in Table 1. These boxplots compare the default configuration of each detector against the BISO, BI, and SO tuning heuristics proposed in this study.

Figs. 6 to 7 depict outcomes obtained from high amplitude variations using the same configurations as those utilized for low amplitude variations.

Fig. 6 demonstrates results from high amplitude variations without perturbations, referring to item K in Table 1. Conversely, Fig. 7 exhibits the results from high amplitude variations with perturbations, relating to items L, M, N, and O in Table 1.

In Fig. 4, it is evident that the BISO and BI configurations outperformed the default configuration across all DDs, except for BI in the HDDM$_W$ detector. Fig. 5 shows a similar trend, except in the case of the HDDM$_W$ detector, where the BISO heuristic performed worse than the default configuration. However, the SO heuristic stood out in this scenario, indicating that in instances of increased signal variability and low amplitudes, a less sensitive heuristic is more appropriate. Another notable observation is that the PH detector was more affected by the introduction of perturbations compared to the HDDM detectors, as inferred from the Correct Detection metric.

In Fig 6, the BI detector demonstrated exceptional performance across most cases, failing to improve drift detection only for the HDDM$_A$ detector. Fig. 7 depicts a similar pattern, where the BI heuristic closely matched the default configuration and underperformed only in the HDDM$_W$ detector. Notably, the BI heuristic was the sole approach to achieve results above 0% for the KSWIN detector, which is an interesting observation.

The DDs results showcase promising median performance following the unsupervised tuning process. Upon analyzing the figures, it becomes apparent that the primary factor influencing a decrease in Correct Detection of DDs is the presence of signal perturbation, which addresses RQ2. Notably, for the PH detector, it is clear that unsupervised tuning using the BISO and BI approaches consistently delivered better results across all scenarios, slightly favoring the BI approach. This validates our RQ1.

To precisely identify drifts, it is vital to detect them within their actual occurrence range. Any identification beyond this range leads to a FP. Remarkably, the BI configuration

Algorithm 2 Unsupervised Tuning

procedure UnsupervisedTuning
    input: signal, ▷ refers to the signal for which the existence of CDs will be evaluated.
    perturbation_indices, ▷ indicates the labeled perturbations in the signal.
    dd_hyperparameters_percentages ▷ represents the percentages of hyperparameters to be evaluated, as described in Step 4 in Section IV.
    output: BISO, ▷ as described in Step 5 in Section IV.
    BI, ▷ as described in Step 5 in Section IV.
    SO, ▷ as described in Step 5 in Section IV.
    correct_detection, ▷ returns a list containing the Correct Detection metric for each of the heuristics.

    for *i* in range(dd_hyperparameters_percentages) **do**
        $intervals \leftarrow CDS(signal)$
        $detections \leftarrow drift\_detector(signal, dd\_hyperparameters\_percentages)$
        $BISO \leftarrow compare\_drifts(detections, intervals)$
        $BI \leftarrow compare\_drifts(detections, intervals)$
        $SO \leftarrow compare\_drifts(detections, intervals)$
    end for
    $correct\_detection \leftarrow (BISO, BI, SO, perturbation\_indices)$
    return $BISO, BI, SO, correct\_detection$
end procedure

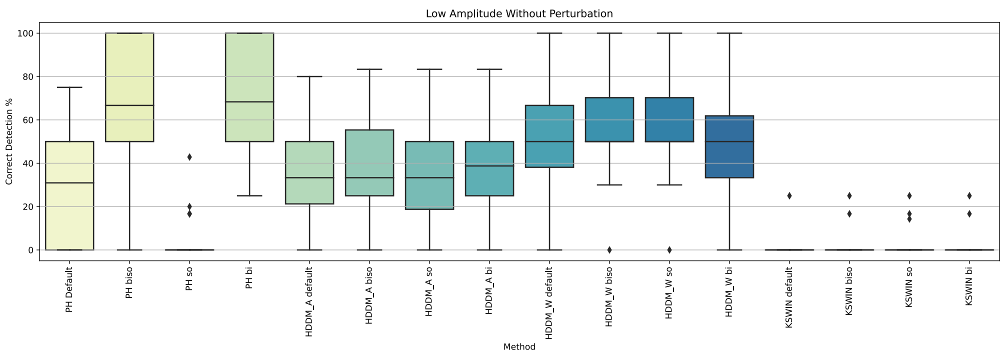


FIGURE 4. **Boxplot of low amplitudes variation without perturbations on signal experiments.**

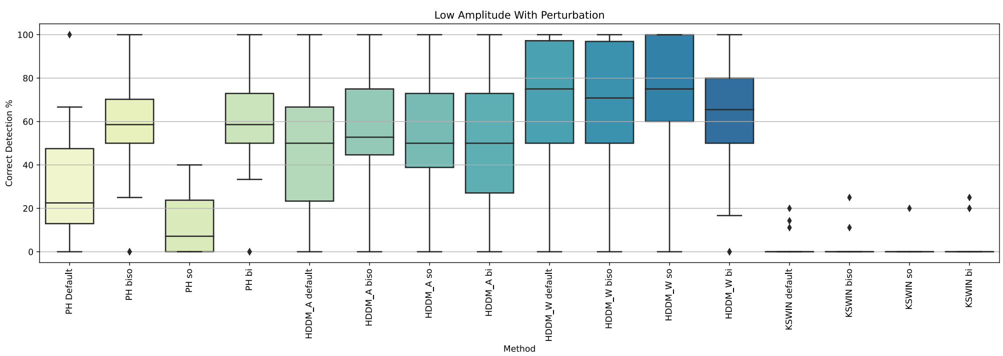


FIGURE 5. **Boxplot of low amplitudes variation with perturbations on signal experiments.**

showcased the most favorable performance in the conducted assessments. We carried out experiments employing various signal sizes, amplitudes, and perturbation levels to ensure the method's applicability across different scenarios and to simulate the behavior seen in natural phenomena.

Overall, our experiments highlighted the proposed method's viability as an alternative for configuring DDs. Notably, the SO heuristic displayed comparatively weaker performance among the heuristics, mainly due to its conservative design. It focuses on identifying the fewest drifts

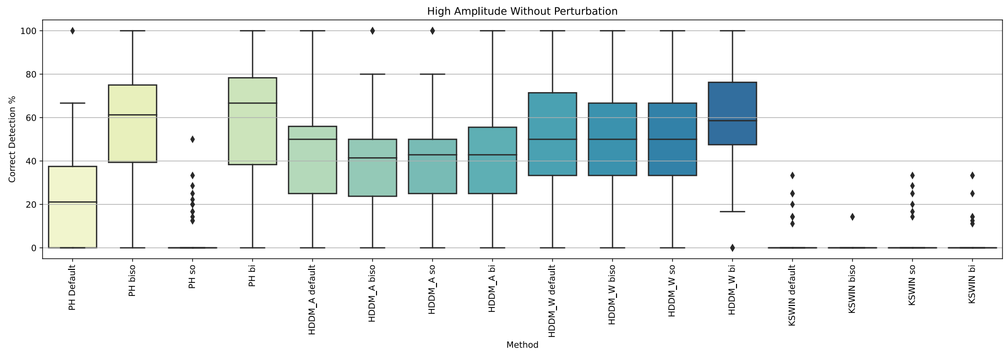


FIGURE 6. Boxplot of high amplitudes variation without perturbations on signal experiments.

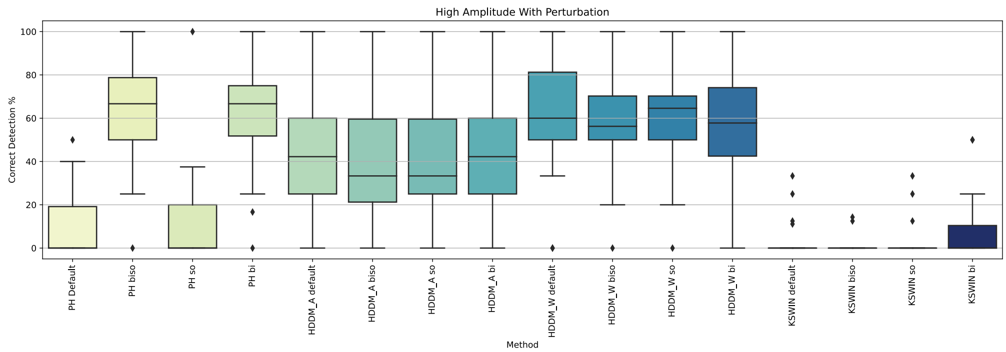


FIGURE 7. Boxplot of high amplitudes variation with perturbations on signal experiments.

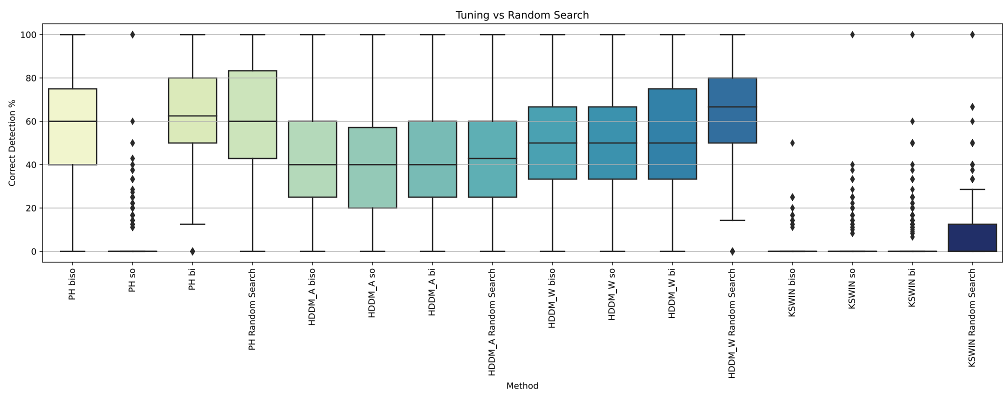


FIGURE 8. Boxplot of our tuning process vs Random Search.

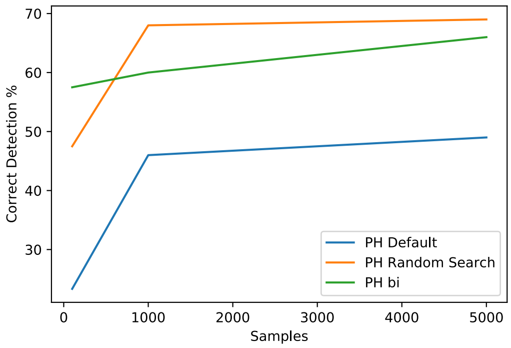
outside critical intervals to reduce false positives, potentially missing actual drifts occurring within these regions.

One potential explanation for KSWIN frequently yielding 0% results could be misconfigured hyperparameters. For instance, if the window size does not align well with the data's change rate, subtle CD might not be detected. Similarly, setting a high change threshold could cause the algorithm to overlook changes below that threshold. ADWIN and DDM detectors might have faced similar issues due to their operational characteristics. Additionally, sudden or large changes in data magnitude could pose challenges for detection, especially using a sliding window approach to calculate CD statistics.

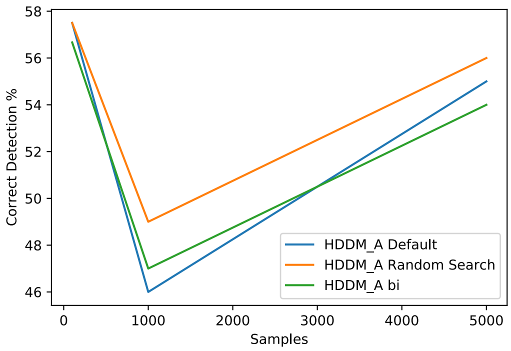
A. UNSUPERVISED TUNING VS RANDOM SEARCH

In light of the current void in the literature pertaining to unsupervised tuning approaches, a pragmatic decision has been made to introduce Random Search (RS) [52], [54] as a competitor in our study. Unsupervised tuning, in its nuanced complexities, poses a unique set of challenges that, unfortunately, has not been extensively addressed in existing research. The absence of established methodologies and benchmarks for unsupervised tuning strategies underscores the need for an adaptable and exploratory approach.

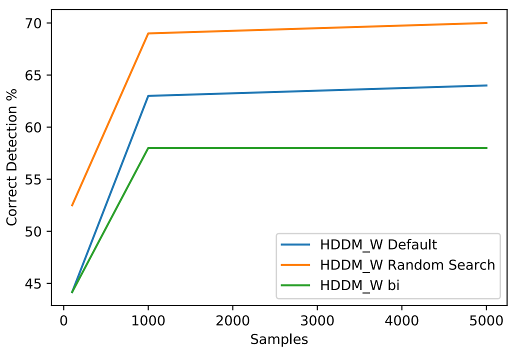
Fig. 8 presents a comparison between our unsupervised tuning method and supervised Random Search (RS) tuning. RS is known for accurately defining hyperparameters but



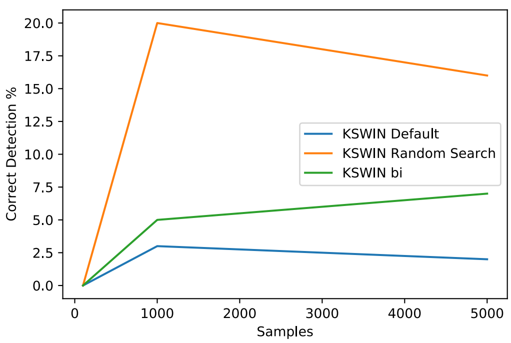
(a) Page-Hinkley without perturbations.



(b) $HDDM_A$ without perturbations.

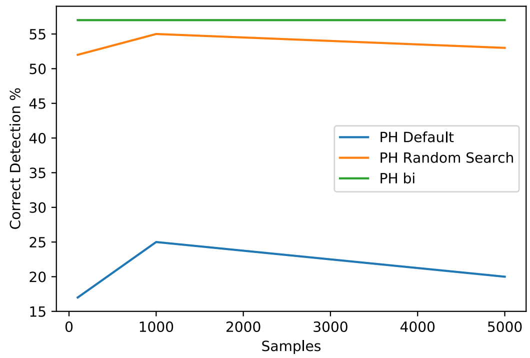


(c) $HDDM_W$ without perturbations.

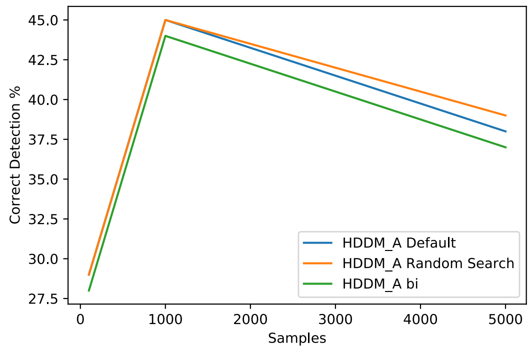


(d) KSWIN without perturbations.

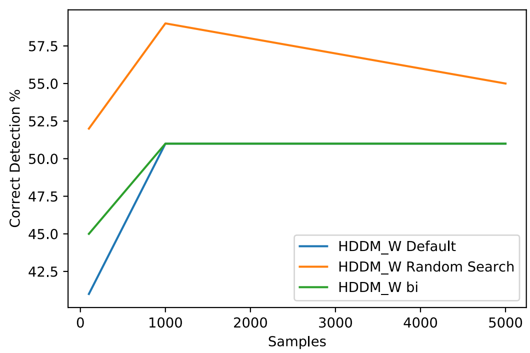
FIGURE 9. Correct drift detection without perturbations in signals with default, RS and BI configuration.



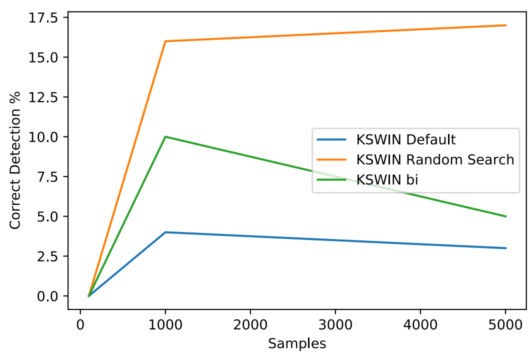
(a) Page-Hinkley with perturbations.



(b) $HDDM_A$ with perturbations.



(c) $HDDM_W$ with perturbations.



(d) KSWIN with perturbations.

FIGURE 10. Correct drift detection with perturbations in signals with default, RS and BI configuration.

relies on a supervised approach, which might not always be feasible in real-world scenarios.

Given that these experiments were intentionally designed with deliberate data perturbations, all configurations listed in

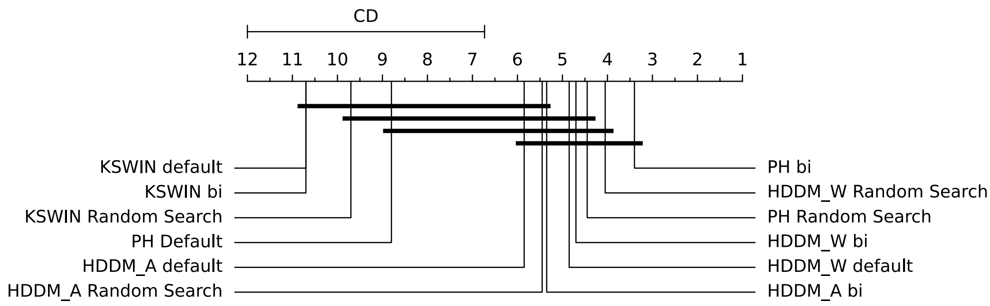


FIGURE 11. **Friedman's statistcal test on experiment J, size = 5000.**

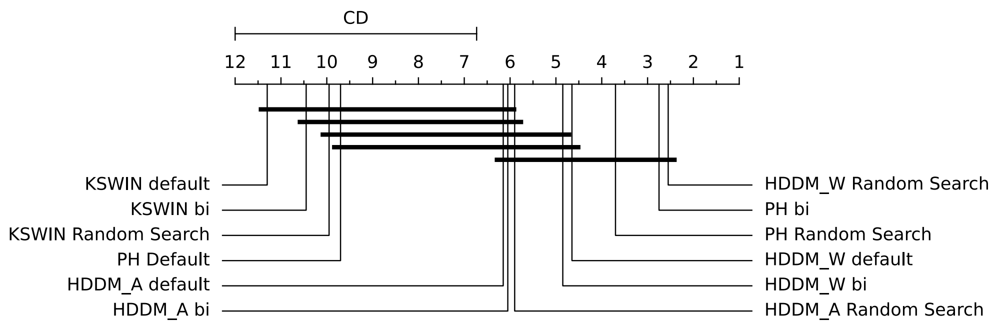


FIGURE 12. **Friedman's statistcal test on experiment O, size = 5000.**

Table 1 were utilized. As expected, the RS method showed superior results across most test cases compared to our tuning approach. However, our method notably outperformed the supervised approach specifically for the PH detector. Moreover, in all other experiments, our method consistently matched the results generated by the RS method. This comparison validates the effectiveness of our proposed tuning approach and supports RQ1 based on the achieved outcomes.

B. NUMBER OF SAMPLES IMPACT

For a more thorough validation of RQ2 and to analyze how the number of samples affects the accurate detection of drifts, we conducted a new assessment. Fig. 9 and Fig. 10 illustrate this impact on DDs performance by comparing three detection configurations: default, RS, and BI. Fig. 9 displays artificial signals generated by the CSG without perturbations, while Fig. 10 shows signals generated by the CSG with perturbations. This evaluation encompassed all instances detailed in Table 1.

In Fig. 9, we noticed a tendency: as the number of samples increased, Correct Detection also tended to rise. This trend might be due to the detectors retaining more statistical information about the signal being evaluated. A similar trend was observed in Fig. 10, albeit with a lower percentage of correct detections. Additionally, the default configuration outperformed other setups only for the $\mathrm{HDDM_A}$ detector.

In Fig. 10, the results for the PH detector show that despite RS benefiting from its supervised approach, the BI heuristic achieved better Correct Detection. Also, the BI heuristic consistently outperformed the default configuration in most cases for the DDs.

The findings across Figs. 9 to 10 reaffirm that the presence of perturbations in the signal significantly reduces the Correct Detection of DDs, further supporting RQ2, highlighting that the foremost influencing factor is the presence of these perturbations.

C. STATISTICAL ANALYSIS

To examine the noticeable differences between DDs configurations, we utilized Friedman's statistical test alongside the Nemenyi post-hoc test. Friedman's test involves ranking the observations of each configuration in ascending order and calculating the sum of differences between these observed ranks and the expected ranks, which would be identical if all samples had the same median.

For this test, we focused on configurations J and O from Table 1. These configurations represent the most complex settings highlighted in our evaluations. This specific selection aimed to illustrate the test within the most complex scenario for drift detection in comparison to the other experiments.

From the results depicted in Fig. 11, it is evident that the KSWIN, $\mathrm{HDDM_A}$, and $\mathrm{HDDM_W}$ approaches displayed no statistically significant differences between the default configuration and the tuned configurations. However, concerning the PH detector, both PH with BI and PH with RS configurations exhibited statistically significant differences compared to the default setting. A similar trend is observed in

Fig. 12, where only the PH configuration showed substantial differences compared to the standard configuration.

VII. CONCLUSION

This study presents critical findings about the impact of different hyperparameter setups on drift detection performance across various scenarios. The RS configuration stands out as the most effective in identifying drifts, yet it relies on user knowledge and labeled databases, limiting its use in real-world unsupervised settings.

Conversely, the BI tuning approach offers an unsupervised solution independent of specialized supervision or labeled datasets. It shows promise, especially in scenarios with high amplitudes and significant variations, consistently outperforming default configurations in multiple detectors. This highlights the importance of unsupervised tuning methods in practical applications for detecting drifts.

The current landscape of labeled datasets for CD is inadequate to meet the growing demands of developing adaptive machine learning models. Recognizing the urgency of this issue is imperative to undertake initiatives for generating artificial but controlled datasets such as CSG to simulate the complexities of CD.

Future research aims to broaden the tuning process by including various DDs and real-world signal datasets. This expansion, once validated, is expected to demonstrate robust performance not only in offline scenarios but also in dynamic online contexts, like data stream analysis. This advancement holds significant potential for improving the practical use of drift detection techniques across diverse domains.

REFERENCES

[1] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*. Reading, MA, USA: Addison-Wesley, 1987.

[2] Q. Xu, Q. Wen, and L. Sun, "Two-stage framework for seasonal time series forecasting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3530–3534.

[3] R. Kumar, A. Jain, A. K. Tripathi, and S. Tyagi, "COVID-19 outbreak: An epidemic analysis using time series prediction model," in *Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2021, pp. 1090–1094.

[4] F. Jiang, Z. Zhao, and X. Shao, "Time series analysis of COVID-19 infection curve: A change-point perspective," *J. Econometrics*, vol. 232, no. 1, pp. 1–17, Jan. 2023.

[5] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using LSTM-based autoencoders," in *Proc. IEEE Int. Conf. Adv. Trends Inf. Theory (ATIT)*, Jul. 2019, pp. 513–517.

[6] E. B. Gulcan and F. Can, "Unsupervised concept drift detection for multi-label data streams," *Artif. Intell. Rev.*, vol. 56, no. 3, pp. 2401–2434, Mar. 2023.

[7] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence—SBIA*. Berlin, Germany: Springer, 2004, pp. 286–295.

[8] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019.

[9] X.-D. Zhang, *Modern Signal Processing*. Berlin, Germany: Walter de Gruyter, 2022.

[10] P. Gangsar and R. Tiwari, "Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review," *Mech. Syst. Signal Process.*, vol. 144, Oct. 2020, Art. no. 106908.

[11] V. C. Silva, B. B. Zarpelão, E. Medvet, and S. Barbon, "Explainable time series tree: An explainable top-down time series segmentation framework," *IEEE Access*, vol. 11, pp. 120845–120856, 2023.

[12] S. L. Ullo and G. R. Sinha, "Advances in smart environment monitoring systems using IoT and sensors," *Sensors*, vol. 20, no. 11, p. 3113, May 2020.

[13] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2020.

[14] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, "Uncertainty in big data analytics: Survey, opportunities, and challenges," *J. Big Data*, vol. 6, no. 1, pp. 1–16, Dec. 2019.

[15] S. L. Amarasinghe, S. Su, X. Dong, L. Zappia, M. E. Ritchie, and Q. Gouil, "Opportunities and challenges in long-read sequencing data analysis," *Genome Biol.*, vol. 21, no. 1, pp. 1–16, Dec. 2020.

[16] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "$*CADE*$: Detecting and explaining concept drift samples for security applications," in *Proc. 30th USENIX Secur. Symp. (USENIX Security)*, 2021, pp. 2327–2344.

[17] I. Mavromatis and A. Khan, "Demo: LE3D: A privacy-preserving lightweight data drift detection framework," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2023, pp. 917–918.

[18] H. Tian, N. L. D. Khoa, A. Anaissi, Y. Wang, and F. Chen, "Concept drift adaption for online anomaly detection in structural health monitoring," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2813–2821.

[19] T. Liu, X. Ma, S. Li, X. Li, and C. Zhang, "A stock price prediction method based on meta-learning and variational mode decomposition," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109324.

[20] A. R. Masegosa, A. M. Martínez, D. Ramos-López, H. Langseth, T. D. Nielsen, and A. Salmerón, "Analyzing concept drift: A case study in the financial sector," *Intell. Data Anal.*, vol. 24, no. 3, pp. 665–688, May 2020.

[21] M. Andreoni Lopez, D. M. F. Mattos, O. C. M. B. Duarte, and G. Pujolle, "Toward a monitoring and threat detection system based on stream processing as a virtual network function for big data," *Concurrency Computation: Pract. Exper.*, vol. 31, no. 20, p. e5344, Oct. 2019.

[22] B. Veloso, J. Gama, B. Malheiro, and J. Vinagre, "Hyperparameter self-tuning for data streams," *Inf. Fusion*, vol. 76, pp. 75–86, Dec. 2021.

[23] V. E. Martins, A. Cano, and S. Barbon Junior, "Meta-learning for dynamic tuning of active learning on stream classification," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109359.

[24] H. Yu, Q. Zhang, T. Liu, J. Lu, Y. Wen, and G. Zhang, "Meta-ADD: A meta-learning based pre-trained model for concept drift active detection," *Inf. Sci.*, vol. 608, pp. 996–1009, Aug. 2022.

[25] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020.

[26] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *J. Mach. Learn. Res.*, vol. 19, no. 72, pp. 1–5, 2018.

[27] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, and A. Bifet, "River: Machine learning for streaming data in Python," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 4945–4952, 2021.

[28] R. P. Silva, B. B. Zarpelão, A. Cano, and S. B. Junior, "Time series segmentation based on stationarity analysis to improve new samples prediction," *Sensors*, vol. 21, no. 21, p. 7333, Nov. 2021, doi: 10.3390/s21217333.

[29] J. Read, R. A. Rios, T. Nogueira, and R. F. de Mello, "Data streams are time series: Challenging assumptions," in *Proc. Conf. Artif. Intell. (SBIA)*, Rio Grande, Brazil. Berlin, Germany: Springer-Verlag, 2020, pp. 529–543.

[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–25, 2012.

[31] N. L. A. Ghani, I. A. Aziz, and M. Mehat, "Concept drift detection on unlabeled data streams: A systematic literature review," in *Proc. IEEE Conf. Big Data Analytics (ICBDA)*, Nov. 2020, pp. 61–65, doi: 10.1109/ICBDA50157.2020.9289802.

[32] A. Cano and B. Krawczyk, "Kappa updated ensemble for drifting data stream mining," *Mach. Learn.*, vol. 109, no. 1, pp. 175–218, Jan. 2020.

[33] Y. Xie, C. Zhu, W. Zhou, Z. Li, X. Liu, and M. Tu, "Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances," *J. Petroleum Sci. Eng.*, vol. 160, pp. 182–193, Jan. 2018.

[34] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.
[35] A. L. Garcia, *Probability and Random Processes for Electrical Engineering*. India: Pearson Education, 1994.
[36] R. S. M. Barros and S. G. T. C. Santos, "A large-scale comparison of concept drift detectors," *Inf. Sci.*, vols. 451–452, pp. 348–370, Jul. 2018, doi: 10.1016/j.ins.2018.04.014.
[37] S. G. T. C. Santos, R. S. M. Barros, and P. M. G. Júnior, "Optimizing the parameters of drift detection methods using a genetic algorithm," in *Proc. IEEE 27th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2015, pp. 1077–1084.
[38] R. N. Gemaque, A. F. J. Costa, R. Giusti, and E. M. dos Santos, "An overview of unsupervised drift detection methods," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 6, p. e1381, Nov. 2020.
[39] S. Agrahari and A. K. Singh, "Concept drift detection in data stream mining: A literature review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9523–9540, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.006.
[40] H. Hu, M. Kantardzic, and T. S. Sethi, "No free lunch theorem for concept drift detection in streaming data classification: A review," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 2, p. e1327, Mar. 2020.
[41] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining Knowl. Discovery*, vol. 13, no. 2, p. e1484, Mar. 2023.
[42] A. Nugroho and H. Suhartanto, "Hyper-parameter tuning based on random search for DenseNet optimization," in *Proc. 7th Int. Conf. Inf. Technol., Comput., Electr. Eng. (ICITACEE)*, Sep. 2020, pp. 96–99, doi: 10.1109/ICITACEE50144.2020.9239164.
[43] H. Sarieddeen, M.-S. Alouini, and T. Y. Al-Naffouri, "An overview of signal processing techniques for terahertz communications," *Proc. IEEE*, vol. 109, no. 10, pp. 1628–1665, Oct. 2021.
[44] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, nos. 1–2, p. 100, Jun. 1954.
[45] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. 7th SIAM Int. Conf. Data Min.*, 2007, pp. 443–448.
[46] I. Frías-Blanco, J. d. Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, Mar. 2015, doi: 10.1109/TKDE.2014.2345382.
[47] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, Apr. 2010.
[48] C. Raab, M. Heusinger, and F.-M. Schleif, "Reactive soft prototype computing for concept drift streams," *Neurocomputing*, vol. 416, pp. 340–351, Nov. 2020, doi: 10.1016/j.neucom.2019.11.111.
[49] S. G. T. C. Santos, R. S. M. Barros, and P. M. Gonçalves, "A differential evolution based method for tuning concept drift detectors in data streams," *Inf. Sci.*, vol. 485, pp. 376–393, Jun. 2019.
[50] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022.
[51] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.
[52] N. Buslim, I. L. Rahmatullah, B. A. Setyawan, and A. Alamsyah, "Comparing Bitcoin's prediction model using GRU, RNN, and LSTM by hyperparameter optimization grid search and random search," in *Proc. 9th Int. Conf. Cyber IT Service Manage. (CITSM)*, Sep. 2021, pp. 1–6, doi: 10.1109/CITSM52892.2021.9588947.
[53] O. A. Mahdi, E. Pardede, N. Ali, and J. Cao, "Fast reaction to sudden concept drift in the absence of class labels," *Appl. Sci.*, vol. 10, no. 2, p. 606, Jan. 2020.
[54] M. Zhou, X. Zhong, Y. Sun, and L. Gan, "Prediction model of coal consumption based on random forest variable selection and random-grid hyperparametric optimization algorithm," in *Proc. Int. Conf. Power Syst. Technol. (PowerCon)*, Sep. 2023, pp. 1–5, doi: 10.1109/powercon58120.2023.10331350.

RICARDO PETRI SILVA received the bachelor's and master's degrees in computer science from Universidade Estadual de Londrina, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, with a focus on online machine learning. He is also a Professor of computer science courses with UniFil, Londrina, Brazil, where he coordinates an extension project. This project aims to facilitate community access to technology and to develop socially impactful projects addressing societal needs.

SYLVIO BARBON JUNIOR received the B.Sc. degree in computer science and the M.Sc. degree in computational physics from the University of São Paulo, in 2005 and 2007, respectively, and the degree in computational engineering and the Ph.D. degree in computational physics from IFSC/USP, in 2008 and 2011, respectively. He is currently an Associate Professor with the Department of Engineering and Architecture, University of Trieste (UNITS), Italy. He is also the Co-Director of the Machine Learning Laboratory. Prior to this, he led the Research Group dedicated to the study of machine learning with the Computer Science Department, State University of Londrina (UEL), Brazil, from 2012 to 2021. His research interests include computer vision, pattern recognition, and machine learning, with a current emphasis on meta-learning, stream mining, and process mining.

BRUNO BOGAZ ZARPELÃO received the B.Sc. degree in computer science from the State University of Londrina, Brazil, and the Ph.D. degree in electrical engineering from the University of Campinas, Brazil. Since 2012, he has been an Associate Professor with the Computer Science Department, State University of Londrina (UEL). From March 2018 to February 2019, he was a Visiting Postdoctoral Researcher with the City, University of London. His research interests include data science in cybersecurity, intrusion detection, and the Internet of Things.

LEONIMER FLAVIO DE MELO received the degree and master's degrees in electrical engineering and the Ph.D. degree in mechanical engineering from the State University of Campinas, in 1985, 2002, and 2007, respectively. He is currently an Associate Professor and a Professor of the postgraduate master's program in electrical engineering with the State University of Londrina (UEL), and a Professor of associate doctorate program in electrical engineering with UEL/UTFPR-CP. He has experience in the area of electrical engineering and mechatronic engineering, involved mainly in the following areas: industrial automation, production engineering, embedded control systems, robotics, autonomous vehicles, industrial mechatronic systems, machine learning, deep learning, and artificial neural networks.

Open Access funding provided by 'Università degli Studi di Trieste' within the CRUI CARE Agreement