Expert Systems With Applications 248 (2024) 123342

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Comprehensive analysis of change-point dynamics detection in time series data: A review

Muktesh Gupta *, Rajesh Wadhvani, Akhtar Rasool

Department of CSE, Maulana Azad National Institute of Technology, Link Road Number 3, Bhopal, 462003, M.P., India

A R T I C L E I N F O

Keywords:
Sequential data
Pattern change detection
Supervised and unsupervised detection analysis
Real-time pattern change analysis
Statistical methods

A B S T R A C T

In the ever-evolving field of time series analysis, detecting changes in patterns and dynamics is paramount for accurate forecasting and meaningful insights. This article thoroughly explores several algorithms for detecting and analyzing pattern changes in time series data. The exploration covers a broad spectrum of algorithms, further venturing into their categorization based on functional modalities and the ability to identify complex changes. Recognizing pattern changes in time series data holds pivotal importance as it aids in anticipating future trends, ensuring efficient resource allocation, and mitigating potential challenges. This research goes beyond a basic overview and conducts a thorough comparative analysis, highlighting each algorithm's strengths, drawbacks, and computational complexities. This comparative approach provides practitioners and researchers with the necessary information to select the most suitable algorithm for their requirements. Additionally, this review provides insight into potential future research directions, proposing possible improvements and breakthroughs in the design and application of algorithms. This review also provides a runtime analysis of various pattern change detection algorithms, presenting an in-depth evaluation of the existing methodologies. It serves as a vital reference for individuals dealing with the dynamic nature of time series data.

1. Introduction

Pattern Change Detection (PCD) plays a pivotal role in numerous domains ranging from finance and healthcare to environmental monitoring. The primary purpose is to observe data so that any shift in the behavior of the data pattern is recognized as a change point. Sequential data, a significant subset of data types where the order is inherently crucial, is especially relevant for PCD. Each point in a sequential dataset does not stand alone; instead, it is intricately linked to its predecessors and successors. The importance of sequential data cannot be overstated within the Internet of Things (IoT) framework. IoT devices, such as temperature sensors, continuously produce data, offering insights into patterns or potential anomalies (Sgueglia, Di Sorbo, Visaggio, & Canfora, 2022). These devices can also yield multi-modal data from diverse sources like motion detectors and security cameras. This analysis is essential for optimizing system performance and identifying issues like equipment failures in manufacturing setups (Gupta, Wadhvani, & Rasool, 2023). Detecting changes in sequential data provides a crucial understanding of the performance and security of diverse systems. Neglecting to manage this issue adequately can have significant consequences, including losing intervention opportunities, unnecessary false alarms, or serious misinterpretations.

Over the years, numerous algorithms have emerged to tackle the challenges inherent to pattern change detection. While some have showcased exemplary performance in specific contexts, others have proven resilient with noisy or imperfect data. However, despite the advancements, a comprehensive understanding of their comparative advantages, limitations, and ideal application scenarios still needs improvement. This study seeks to bridge this knowledge gap. The main goal is to provide a detailed comparison of various PCD algorithms, highlighting their techniques, performance measures, and appropriateness in multiple situations. The subsequent section enumerates various applications, followed by an outline of the scientific questions addressed in this survey.

1.1. Applications of pattern change detection in time series data

- Healthcare: Detecting pattern changes in patient medical data can aid in identifying the emergence of diseases or other health issues (Liu, Wright, & Hauskrecht, 2018). For instance, alterations in the frequency or duration of sleep may indicate sleep disorders, whereas changes in physical activity levels may indicate alterations in general health.

* Corresponding author.
E-mail address: mg.203112003@manit.ac.in (M. Gupta).

https://doi.org/10.1016/j.eswa.2024.123342
Received 14 November 2023; Received in revised form 11 January 2024; Accepted 23 January 2024
Available online 29 January 2024

- Manufacturing: Detecting changes in the signal pattern generated by machinery or equipment can assist in identifying potential failures or other issues that may affect production (Guo et al., 2019). For instance, changes in the vibration patterns of a signal generated by the machine sensors may indicate a worn bearing or other mechanical issues.
- Fraud detection: Detecting pattern changes in financial transactions can aid in the identification of fraudulent activity (Pourhabibi, Ong, Kam, & Boo, 2020). For instance, changes in the frequency or location of credit card transactions may indicate fraudulent use.
- Smart home system: Detecting pattern changes in home sensor data can help optimize the home environment and increase energy efficiency for smart home systems (Aminikhanghahi, Wang, & Cook, 2018). For instance, temperature changes can be used to maximize comfort and energy consumption by adjusting heating or cooling systems.
- Network security: Detecting pattern changes in network traffic can assist in identifying potential security breaches and other problems (Li, Zeng, Zhou & Chen, 2019). For example, changes in the frequency or nature of network traffic may indicate an intrusion or other security threat.

1.2. Scientific questions addressed in this review

- Performance and Accuracy: How do the highlighted algorithms stack against each other regarding detection accuracy across diverse datasets?
- Complexity and Efficiency: With limited computational resources, which algorithms offer the best trade-off between complexity and efficiency?
- Robustness and Reliability: In real-world scenarios with noisy or incomplete data, which algorithms are resilient?
- Interpretability and Transparency: Beyond mere detection, which algorithms offer transparent and easily interpretable insights?
- Scalability: In the era of Big Data, which algorithms are best equipped to scale seamlessly with increasing data volumes and complexities?
- Adaptability: Which algorithms are skilled at dynamically adjusting to evolving data streams and changing behaviors?
- Intervention and Actionability: Post detection, which algorithms furnish actionable insights that can be leveraged for timely interventions?

The structure of the paper is as follows: Section 2 provides an overview of pattern change detection and the terminology used in this survey. Section 3 presents a classification of various algorithms and discusses their detailed methodologies. In Section 4, a thorough comparative analysis is conducted to evaluate the performance of multiple methods, covering datasets, outcomes, and discussions. Lastly, Section 5 wraps up by addressing specific questions raised during the survey and analyzing which algorithm is the most efficient for pattern change detection.

2. Exploring the dynamics of pattern change detection in sequential data

2.1. Background

A multivariate time-series of n data points and f features can be denoted as $\boldsymbol{Y}_t^{(M)}$, where each sequence is represented as $[Y_{t1}^{(M)}, Y_{t2}^{(M)}, \dots, Y_{tn}^{(M)}]$. The notation $\boldsymbol{Y}_t^{(M)}$ represents a sequence in $\mathbb{R}^{n\times f}$, and M is the number of time series in a sample. Mathematically, detecting a shift in the pattern of a sequence involves pinpointing a Change Point (CP) at timestamp Y_{p+1}, where the statistical properties of the series before and following the CP differ, i.e., $Y_{t1} = \cdots . = Y_p \neq Y_{p+1} = Y_{tn}$. The statistical properties of Y_{t1} to Y_p are equal to each other but differ from the statistical properties of Y_{p+1} to Y_{tn}. Therefore, the point Y_{p+1} is identified as the beginning of the change and is referred to as the CP.

2.2. Hypothesis formulation

Let the $[Y_{t1}^{(i)}, Y_{t2}^{(i)}, \dots, Y_{tn}^{(i)}]$ be a ith sequence of time series variables. The pattern change detection can be framed as a hypothesis testing problem involving two possibilities: the Null Hypothesis H_0, asserting "No pattern change in the sequential data occurs", & the Alternative Hypothesis H_a, asserting "A pattern change in the sequential data occurs".

$$\begin{aligned} &H_0 : \lambda Y_{t1}^{(i)} = \cdots .. = \lambda Y_{tp}^{(i)} = \lambda Y_{tp+1}^{(i)} ..... = \lambda Y_{tn}^{(i)} \\ &H_a : \exists(p); t1 < tp < tn \mid \lambda Y_{t1}^{(i)} = \cdots .. = \lambda Y_{tp}^{(i)} \neq \lambda Y_{tp+1}^{(i)} ..... = \lambda Y_{tn}^{(i)} \end{aligned} \tag{1}$$

In this case, λ represents the threshold value set according to the application by the algorithms. This threshold may differ depending on the requirements, while $Y_{t1}^{(i)}$ denotes the starting point, and $Y_{tn}^{(i)}$ indicates the endpoint of the time series. Additionally, $Y_{tp+1}^{(i)}$ refers to the change point. Time series can also contain multiple change-points, which can be represented as $\{Y_{t\alpha 1}^{(i)}, Y_{t\alpha 2}^{(i)}, \dots, Y_{t\alpha n}^{(i)}\}$. Each element in this set represents a distinct point in the time series where a significant change occurs.

2.3. Types of pattern changes in sequential data

The statistical properties of a time series can vary due to multiple factors, leading to changes in its pattern (Aminikhanghahi et al., 2018). Some of these factors are outlined below:

2.3.1. Alteration in average within the pattern of sequential data

A change in the mean represents a movement in the data's central tendency (Truong, Oudre, & Vayatis, 2020). It is one of the most frequently noticed alterations in sequential data. Many factors, such as a shift in underlying trends or an external shock to the system, can cause a change in the mean. For example, Part (a) of Fig. 1 illustrates a change in mean occurring at timestamp 500 while the variance remains constant. This type of Shift is typically detected using control charts or statistical tests.

2.3.2. Alteration in dispersion within the pattern of sequential data

A variance change refers to an alteration in the distribution or scattering of the data. It is frequently noticed in financial data, where fluctuations in volatility result from movements in market sentiment or underlying economic conditions. Part (b) of Fig. 1 illustrates a change in variance. A shift in variance can be noticed using approaches such as the CUmulative SUM (CUSUM) control chart or statistical tests.

2.3.3. Shift in average and dispersion in the pattern of sequential data

A change in the mean–variance (MV) indicates a shift in the data's central tendency and dispersion. This type of variation is frequently found in climate Time Series Data when both the magnitude and volatility of returns fluctuate over time. Part (c) of Fig. 1 illustrates a change in both variance and mean simultaneously. It is detectable using techniques such as the moving average, statistical analyses, and control charts.

2.3.4. Alteration in auto-covariance in the pattern of sequential data

It is the correlation between observations in a time series at various delays. A shift in auto-correlation indicates a change in the data's dependence structure. Part (d) of Fig. 1 illustrates an auto-correlation change in sequential data.

2.3.5. Shift in periodicity in the pattern of sequential data

A change in frequency is an alteration in the periodicity of the data. This type of change is frequently found in seasonal data when the cycle frequency alters due to underlying factor shifts. Part (e) of Fig. 1 illustrates a change in frequency in sequential data. Techniques such as spectrum analysis and wavelet analysis can detect it.

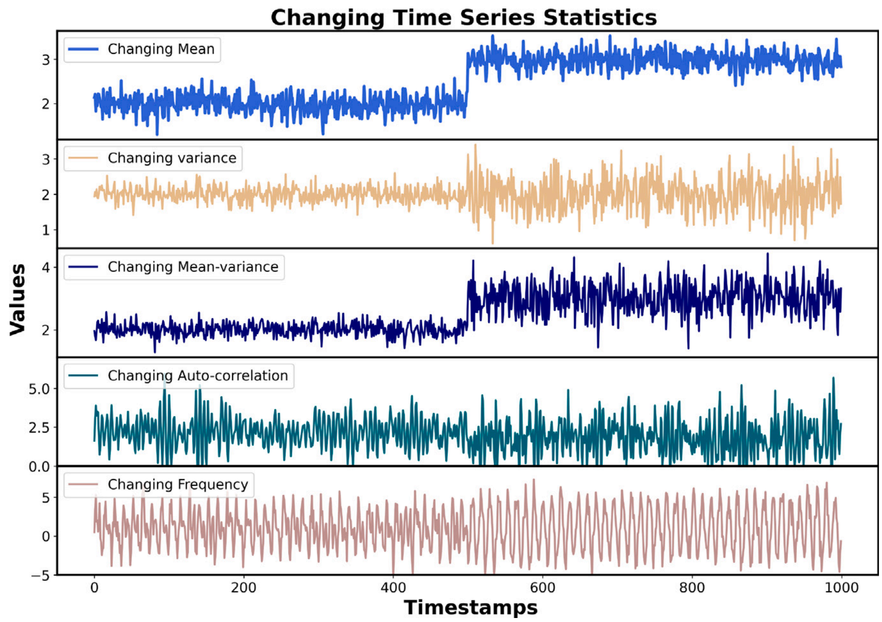

Fig. 1. Part (a) of the figure demonstrates a shift in the mean while maintaining a constant variance. The subsequent part (b) shows a change in the variance. Part (c) reveals a change in both mean and variance, whereas part (d) indicates a shift in auto-correlation. Lastly, part (e) presents a change in frequency. In this figure, the 500th timestamp is selected as the change point for all these types of changes.

2.4. Approaches to detect pattern change in time series data

These approaches are proficient in identifying changes in the statistical patterns of time series data over a period (Aminikhanghahi & Cook, 2017). The algorithms are divided into two groups based on their operational mode: offline and online pattern change detection algorithms. The subsequent section provides a detailed overview of these categories, emphasizing the unique attributes and capabilities of offline and online PCD algorithms.

2.4.1. Offline working mode of pattern change detection algorithms

These algorithms are generally used for retrospective analysis that requires the entire data to identify changes. Their conventional procedure encompasses data modeling and conducting change detection using the model's residuals. Timestamp delay refers to the time gap between the moment an algorithm predicts a CP and the actual occurrence of the CP. A characteristic feature of offline algorithms is their instantaneous detection of behavioral shifts in a time series with a minimal timestamp delay, as shown in part (a) of Fig. 1. However, it is pivotal to note that they operate on the entire dataset, frequently requiring multiple iterations to detect changes accurately.

2.4.2. Online working mode of pattern change detection algorithms

These techniques are designed for real-time data processing, rendering them highly effective for continuous monitoring. In contrast to offline methods, online algorithms analyze incoming data by comparing it to a pre-established benchmark model, signaling a shift when the data shows significant divergence. A major challenge for these methods is the ability to detect changes while reducing false positives rapidly. As illustrated in part (b) of Fig. 2, there is a timestamp delay that exists between the actual and predicted delays. This delay is an inherent aspect of the real-time processing nature of the algorithm. Consequently, a primary goal for online detection algorithms is to minimize this delay period as much as possible, ensuring prompt and accurate detection of changes.

2.5. Nomenclature and definitions

Throughout this survey, a diverse range of sequential data types and various terminologies are frequently used, defined in the subsequent sections.

2.5.1. Heteroscedastic sequential data

In sequential data, it is expected to encounter heteroscedasticity, which refers to the situation where the variance of the data changes over time instead of remaining constant (Pushkar, Gupta, Wadhvani, & Gyanchandani, 2022). Heteroscedasticity can arise due to several factors, such as changes in the underlying data generation process, the instrument used to measure the data or the sampling frequency. Mathematically, heteroscedasticity can be represented as:

$$V(Yt) = \sigma t^2 \tag{2}$$

Here, V(Yt) represents the time series variance at time t. As statistical models usually assume constant variance over time, the presence of heteroscedasticity can significantly affect the accuracy of these models. Therefore, it is essential to consider heteroscedasticity when modeling sequential data to obtain more accurate results.

2.5.2. Multi-modal sequential data

This sequential data type contains multiple peaks or modes in its underlying distribution, unlike uni-modal data with a single peak (El-Sappagh, Abuhmed, Islam, & Kwak, 2020). In other words, the data can exhibit different patterns or behaviors over time, representing a distinct state of the measured system. For instance, in healthcare monitoring systems, multi-modal time series data may include a patient's pulse rate, blood pressure, and respiratory rate, with each mode representing a different state, such as resting, active, or distressed. Mathematically, multi-modal sequential data can be represented as a Probability Distribution (PD) Function with multiple modes, which traditional time series analysis methods that assume uni-modal data distributions may not be equipped to handle. Hence, innovative methods and techniques are required to manage multi-modal sequential data effectively.

2.5.3. Sliding window in sequential data

The Sliding Window (SW) technique is a prominent method for evaluating sequential data, particularly for detecting change points. A window of a specified size is shifted along the data sequence, and the data points within the window are evaluated to discover any changes in the underlying distribution or statistical features (Aminikhanghahi & Cook, 2017). Fig. 3 demonstrates an example of a SW, where the window size is four with sliding step one. Leveraging past observations,

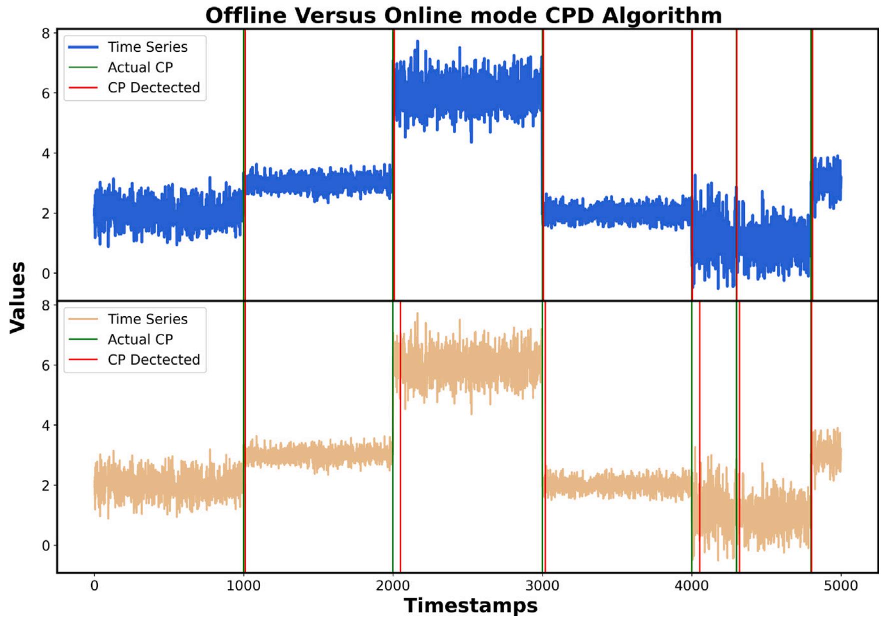

Fig. 2. The time series has changing mean and variation with six change points. Part (a) displays the changes identified by the algorithm working offline mode, while part (b) demonstrates the changes detected in real-time. Some CP are predicted without delay by the online algorithm, others exhibit a delay.

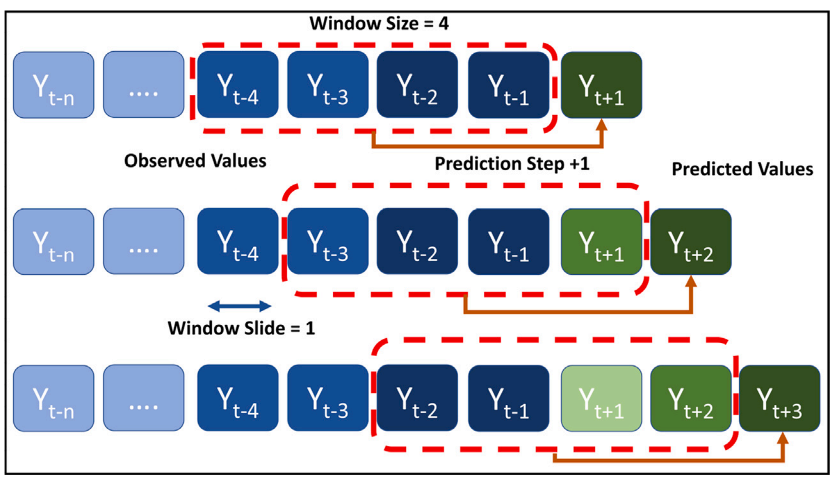

Fig. 3. An instance of a sliding window with a window size=4 & sliding step=1.

the model is trained to predict subsequent values sequentially. The fundamental benefit of the SW technique is its capacity to process real-time data, as only a small percentage of the data is processed at any given moment. The window size selection is crucial as it balances sensitivity to changes and false alarms (Iqbal, et al., 2020). A larger window size can reduce the impact of noise and strengthen the detection robustness, but it may fail to detect short-lived changes. On the other hand, a smaller window size is effective in identifying brief changes but tends to be more vulnerable to noise.

2.5.4. *Hankel matrix of sequential data*

It is a matrix with constant values along its anti-diagonals. It may be derived from a data sequence by organizing the data points into rows and columns so that each anti-diagonal includes equal values (Li, Lin, Lau & Zeng, 2019). The unique symmetry of a Hankel matrix is evident in the regularity of values across these anti-diagonals, where elements are mirrored symmetrically about the anti-diagonal axis. In detecting changes in patterns, the Hankel matrix can be utilized to represent the data's structure or patterns. If a change point is present, it will cause a discontinuity or break in the Hankel matrix's structure. The Hankel matrix, with a window size of W and a length of L, can be depicted as:

$$\mathbf{Y} = \begin{bmatrix} Y_t & Y_{t+1} & \cdots & Y_{t+L-W} \\ Y_{t+1} & Y_{t+2} & \cdots & Y_{t+L-W+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{t+W-1} & Y_{t+W} & \cdots & Y_{t+L} \end{bmatrix}$$

The Hankel matrix plays a vital role in the Singular Spectrum Analysis (SSA) technique, a broader concept mainly used for noise removal in data. This technique decomposes the Hankel matrix into singular values and vectors and is a typical technique for discovering change in the sequential data (Wen, Lu, Liu, & Yan, 2020). By analyzing the singular values and their corresponding vectors, one can detect sudden changes in the data's structure, indicating a change point. The application of Hankel matrix methodologies for change detection in sequential data varies depending on the specific characteristics of the data and the desired sensitivity to changes. Examining singular values and the accompanying singular vectors, which may reveal abrupt shifts or leaps in the data structure, can identify the presence of a transition point. Depending on the specific qualities of the data and the desired

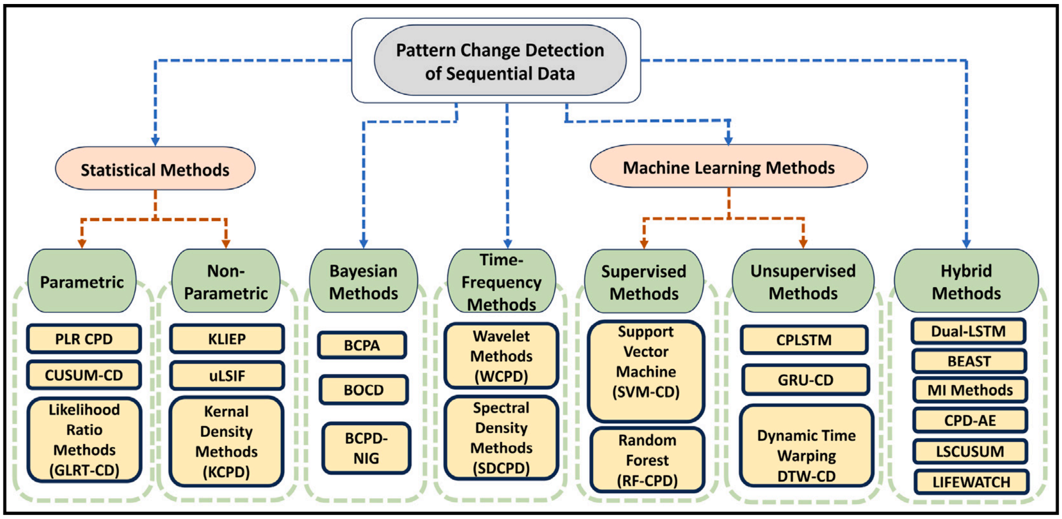

Fig. 4. Classification of pattern change detection algorithms.

level of sensitivity to changes, Hankel matrix techniques can be utilized for change detection in sequential data.

3. Algorithms for pattern change detection in sequential data

A broad spectrum of techniques for identifying pattern shifts in time series data has been found in the literature, encompassing methods based on statistics, entropy, fractals, graphs, subspace analysis, and others. Attempting to encompass all such methods capable of detecting pattern changes takes time and effort. This review focuses on five significant categories of pattern change detection algorithms, each distinguished by their operational techniques. Fig. 4 illustrates these primary categories along with their respective subcategories. Subsequent sections provide a concise overview of the algorithms associated with each category. Moreover, each section concludes with a summary table that organizes the methods chronologically and includes a detailed overview of their limitations.

3.1. Statistical methods for detecting time series pattern change

These techniques analyze the data for changes in statistical characteristics such as mean, variance, and auto-correlation. They typically model the time series data and then identify significant deviations from the model's forecasts, which may indicate a change point. These methods can identify gradual shifts and are extensively applied in various domains, including economics, meteorology, and quality control. The statistical methods are generally classified into two groups, which will be detailed in the following sections.

3.1.1. Statistical methods based on parametric approaches

Parametric statistical methods assume that the data follows a predetermined probability distribution with specific parameters. These methods necessitate a deep comprehension of the underlying distribution, which may include normal, exponential, and Poisson distributions, among others (Aminikhanghahi & Cook, 2017). These approaches are particularly effective when the data's distribution can be precisely determined and modeled. Subsequent subsections delve into models employing this approach, and the concluding part discusses the overall limitations of parametric statistical methods.

3.1.1.1 Models based on piece-wise linear regression (PLR). Muggeo et al. (2008) and Banesh, Petersen, Wendelberger, Ahrens, and Hamann (2019) proposed a method based on PLR to detect pattern change in time series using:

$$Y_{ti} = \beta_0 + \beta_1 * ti + \beta_2 * (ti - tc) * fn(ti > tc) + \epsilon_{ti} \tag{3}$$

Here, Y_{ti} is the data point at time i, model parameters are represented as β_0, β_1, and β_2, the change point is represented as tc, indicator function (fn) that equals 1 when $ti > tc$ and 0 otherwise, and ϵ_{ti} is the error term. PLR divides a time series into segments where a linear relationship approximates each segment. Change points are identified when the properties of these linear relationships shift. Typically, a cost function measures the fit of the linear models to the segments. The method seeks to minimize this cost across all segments while identifying the optimal change points. When a significant change in the linear relationship is observed between segments, it indicates a potential change point in the underlying data. Recent advancements in piece-wise linear function (PWL) fitting involve optimizing PWL functions with specific breakpoints and a distance metric for pattern change detection (Warwicker & Rebennack, 2023). This method excels in anomaly detection, as outliers in the dataset significantly skew the PWL function, making it sub-optimal for segments without outliers.

3.1.1.2 Models based on cumulative sum (CUSUM). Chatterjee and Qiu (2009) and De Oca, Jeske, Zhang, Rendon, and Marvasti (2010) proposed method based on CUSUM in which the change in pattern is calculated using:

$$S_t = S_{Y_{ti}-1} + (Y_{ti} - \mu_0) \tag{4}$$

Here, S_t is the cumulative sum for each time step t, Y_{ti} is the data point at time i, and μ_0 is the reference value. The algorithm calculates cumulative sums of deviations from a reference value, which is typically the mean or a constant. A CP is identified when the accumulated total reaches a predetermined threshold (τ) . The CUSUM statistic is reset to zero when it exceeds a threshold:

$$S_t = max(0, S_{t-1} + (Y_{ti} - \mu_0) - k) \tag{5}$$

Where k is a reference value for the false alarm rate, a CP is detected when $S_t > \tau$. An improved version of CUSUM proposed by Khusna, Mashuri, Ahsan, Suhartono, and Prastyo (2020) employed the bootstrapping technique to pinpoint change points in their research. This involves using resampling methods to estimate the distribution of a statistic without making strong parametric assumptions. Recent advancements have been made in CUSUM-based methods, specifically designed to assess rapid deviations (Kurt, Yılmaz, & Wang, 2020). These methods are applied to sequentially available data streams, focusing on deviations from the nominal dynamic sub-manifold. This approach focuses on extracting and analyzing univariate summary statistics from high-dimensional data streams to differentiate between anomalous and nominal data.

3.1.1.3 Models based on generalized likelihood ratio test (GLRT). Su, Deledalle, Tupin, and Sun (2014) proposed a statistical technique that compares the probabilities of two competing hypotheses to detect a change in a dataset's underlying distribution. The H_0 assumes no change in sequential data, while the H_a assumes a change at a specific point. The likelihood ratio test statistic λ is given by:

$$\lambda = L(H_a)/L(H_0) \tag{6}$$

The test calculates a likelihood ratio. If $\lambda > \tau$, where τ is a threshold, the null hypothesis is rejected, and a CP is detected. The recently developed method utilizing GLRT offers an effective approach for identifying sensor faults in civil structures through hypothesis testing (Li, Liu, Zhang & Li, 2019). This method explores and models multiple sensor faults, such as linear drift and background noise, providing a practical framework for fault detection.

3.1.1.4 Limitations of statistical methods based on parametric approaches

- Parametric methods rely heavily on assuming the data follows a specific distribution. If these assumptions are not met, the methods can produce misleading results.
- By relying on specific assumptions, parametric methods might over-fit the noise in the data.
- These algorithms generally operate in a non-real-time fashion, meaning it is not suited for real-time applications.
- These methods struggle to identify intricate shifts, such as changes in frequency and auto-correlations.
- These methods face challenges in detecting multiple change points within multi-modal data (refer 2.5.2) .

3.1.2 Statistical method based non-parametric approaches

These techniques emphasize the intrinsic properties of the data, using algorithms that operate independently of any specific probability distribution. By not relying on predetermined distributional assumptions, non-parametric methods can handle many data types. They excel in identifying structural changes, shifts in median or variance, and other non-linear patterns within the time series, making them versatile tools in several practical applications. The following subsections explore models that use this approach, while the final subsection addresses the broad limits of non-parametric statistical methods.

3.1.2.1 Models based on kullback–leibler (KL) divergence. Liu, Yamada, Collier, and Sugiyama (2013) and Hushchyn and Ustyuzhanin (2021) proposed a method that calculates KL divergence between two probability distribution function using:

$$D_{KL}(P \parallel Q) = \int P(Y_{ti}) log(P(Y_{ti})/Q(Y_{ti})) dy \tag{7}$$

Here, $P(Y_{ti})$ and $Q(Y_{ti})$ are the PD on which the estimation of the Kullback–Leibler (KL) divergence is obtained and the CP is determined. The Kullback–Leibler Importance Estimation Procedure (KLIEP) examines the data distribution before and after a potential CP to find substantial changes in the data's underlying features. The objective of KLIEP is to estimate the KL divergence between two distributions without knowing their true PD Function. To achieve this, KLIEP approximates the ratio of the two PD Functions using a series of basis functions:

$$r(Y_{ti}) \approx r(\hat{Y}_{ti}) = \sum_{(j=1)}^{d} \alpha_j \beta_j(Y_{ti}) \tag{8}$$

Here, α_j are the coefficients to be estimated, $\beta_j(Y_{ti})$ are the basic functions, and d is the number of basis functions. KLIEP can be utilized in CP identification by separating the data into two segments: before and after a probable CP. A high divergence value indicates a substantial change in the data's attributes. The most probable CP can be selected by comparing the calculated KL divergence values and analyzing all possible CP. Recent advancements have seen the application of KL Divergence in detecting anomalies in noisy data series (Zhou, Gueaieb, & Spinello, 2023). The KL divergence filter, designed for large-scale systems, can be used in model-based and model-free settings. This algorithm assigns a local KL divergence value to each data point, making it versatile for different sensor types.

3.1.2.2 Models based on unconstrained least squares importance fitting (uLSIF). Wang, Borsoi, Richard, and Chen (2023) proposed a uLSIF by optimizing objectives of KLIEP such as:

$$\text{minimize } \alpha \in R^d : E[(r(Y_{ti}) - r(\hat{Y}_{ti}))^2] \tag{9}$$

Here, α is a vector with d dimensions, and each vector component is a real number. KLIEP aims to reduce the empirical KL divergence between the true and estimated ratios, and uLSIF seeks to minimize the squared error between the two ratios (Yamada, Kimura, Naya, & Sawada, 2013). The objective of uLSIF is to find the optimal coefficients α_j in Eq. (8) that minimize the mean squared error between the true ratio $r(Y_{ti})$ and the estimated ratio $r(\hat{Y}_{ti})$, Unlike KLIEP, uLSIF's coefficients α_j are not constrained, making it an unconstrained optimization problem. This difference in optimization objectives and constraints results in varying estimation outcomes when comparing the two methods. Being unconstrained, the optimization in uLSIF can be more stable and more accessible than in methods with constraints, like KLIEP.

3.1.2.3 Models based on kernel density estimation. Chang, Li, Yang, and Póczos (2019) proposed a method that detects CP by calculating kernel density using:

$$r_t(Y_{ti}) = (1/Y_t) * \sum K((Y_t - Y_{ti})/B) \tag{10}$$

Here, K is the kernel function, and B is the bandwidth. Kernel Change Point Detection (KCPD) is a non-parametric approach that estimates the likelihood density function of a dataset using kernel functions. It compares the kernel density estimates before and after a potential CP using Eq. (10) to search for changes in the data distribution. When the difference in densities surpasses a specified threshold, the algorithm detects a CP using:

$$|r_t(Y_{ti}) - r_{t+1}(Y_{ti})| > \tau \tag{11}$$

Here, τ is the threshold value that depends on the application. KCPD excels at identifying shifts in non-parametric and intricate distributions. It supports diverse kernel functions like Gaussian and polynomial, enhancing its adaptability to varied data types. Recent applications in semiconductor manufacturing employ a KDE-based method for detecting pattern changes (Lang et al., 2022). This approach uses Kernel Density Estimation to generate univariate distributions for each sensor and time point, subsequently merging them and applying a low-pass filter to minimize noise. The KDE-based anomaly detection method is adaptable to multiple processes, requires only nominal data for training, and functions effectively with limited data.

3.1.2.4 Limitations of statistical methods based on non-parametric approaches

- Non-parametric methods require more data to achieve the same level of precision as a parametric method.
- Some non-parametric techniques can be computationally expensive, especially with large datasets.
- Some non-parametric methods, especially those with high adaptability, can over-fit noise in the data.

Table 1
A overivew of various Statistical methods and their associated limitations.

Ref.	Methods	Limitations
Li, Liu, et al. (2019)	GLRT	GLRT requires significant computational resources, particularly for datasets with multiple dimensions. Additionally, it depends on certain assumptions about data distribution that may not always correspond to real-world conditions.
Kurt et al. (2020)	CUSUM	CUSUM is less effective for real-time applications and struggles with complex shifts like frequency and auto-correlation changes.
Warwicker and Rebennack (2023)	PLR	PLR is efficient for linear datasets and reduces over-fitting risk. Still, it struggles with complex trends and is sensitive to noise, potentially leading to false pattern change detections.
Lang et al. (2022)	KCPD	Over-fitting occur if not properly regularized or with incorrect kernel parameters, and its non-linear nature may lead to interpretative difficulties compared to linear methods.
Zhou et al. (2023)	KLIEP	The constraint within KLIEP ensures that the estimated density ratio closely matches the target distribution, enhancing stability but complicating the optimization process.
Wang et al. (2023)	uLSIF	uLSIF may over-fit in unconstrained settings, particularly with many basis functions and sparse data, and is sensitive to basis function choice, necessitating careful selection or tuning.

- It can sometimes be more challenging to quantify uncertainty (e.g., confidence intervals) in non-parametric settings compared to parametric ones.

For reference, Table 1 lists various Statistical methods and outlines their limitations.

3.2 Bayesian approaches for pattern changes detection in sequential data

Leveraging prior knowledge and dynamically adjusting assumptions with new evidence, Bayesian methods demonstrate robustness in identifying pattern changes in sequential data (Pushkar et al., 2022). These methods utilize Probability Distributions for modeling data uncertainty and apply Bayesian inference for model updates. This approach offers adaptability and flexibility in pattern change detection and analysis, accommodating uncertainties and variations in the data generation. The following subsections explore several models that use this Bayesian framework, while the final section addresses the broad limitations of Bayesian methods in this context.

3.2.1 Marginal likelihood probability-based methods

Corradin, Danese, and Ongaro (2022) proposed a method that detects CP by calculating Joint Probability Distribution (JPD) using:

$$P(C|Y_{t1},\dots,Y_T) \propto P(Y_{t1},\dots,Y_{tn}|C) * P(C) \tag{12}$$

Here, C is a set of CP, and Y_T is the number of time-steps. Bayesian Change Point Analysis (BCPA) calculates the marginal likelihood $P(Y_{t1},\dots,Y_{tn}|C)$ using a conjugate prior for the parameters of the data distribution and assumes a prior distribution P(C) for the CP. The most probable set of CP can be obtained by maximizing the JPD using:

$$C^* = argmax\, P(C|Y_{t1},\dots,Y_{tn}) \tag{13}$$

The BCPA excels in handling scenarios involving multiple change points, accurately identifying both their number and locations. BCPA enhances resilience against outliers and data inconsistencies using prior distributions and hierarchical structures. The Bayesian methodology seamlessly accommodates models ranging from basic mean shifts to intricate distributional alterations.

3.2.2 Bayesian method based on online pattern change detection

Online Algorithm based on Bayesian proposed by Altamirano, Briol, and Knoblauch (2023) estimates the probability of a CP at each time step using:

$$P(I_t = 1|Y_{t1},\dots,Y_{tn}) \propto P(Y_t|I_t = 1, Y_{t1},\dots,Y_{tn-1}) * P(I_t = 1|Y_{t1},\dots,Y_{tn-1}) \tag{14}$$

Here, at time t, I_t indicates the presence of a CP. The Bayesian Online Change Point Detection (BOCD) method calculates the predictive probability $P(Y_t|I_t = 1, Y_{t1},\dots,Y_{tn-1})$ by employing a conjugate prior suited for the data distribution parameters. The prior $P(I_t = 1|y_1,\dots,y_{t-1})$ is determined based on the predicted CP frequency. Essentially, I_t functions as a signal: if I_t equals 1, a CP is identified at time t; otherwise, it is absent. A CP gets flagged at time t when the computed probability surpasses a designated threshold, typically based on the target sensitivity and specificity. Being an online algorithm, BOCD can process data as it arrives, making it suitable for real-time applications.

3.2.3 Bayesian method based on normal-inverse gamma

Lu, Wang, Zhang, and Wang (2023) proposed a novel Bayesian method for change point detection with the Normal-Inverse Gamma Prior (BCPD-NIG). The method models a time series data as a sequence of segments, each of which follows a normal distribution with unknown mean–variance (Thies & Molnár, 2018). To estimate the mean–variance of each segment, the NIG prior is used as a conjugate prior. The joint probability distribution of a CP occurring at time t is calculated by:

$$P(C,\mu,\sigma^2|y_{t1},\dots,y_{tn}) \propto P(y_{t1},\dots,y_{tn}|Y_{tc},\mu,\sigma^2) * P(\mu,\sigma^2) * P(Y_{tc}) \tag{15}$$

Here, $P(C,\mu,\sigma^2|y_{t1},\dots,y_{tn})$ represents the posterior distribution given the observed data, $P(y_{t1},\dots,y_{tn}|K,\mu,\sigma^2)$ is the likelihood of observing the data given particular CP Y_{tc}. The prior distribution is $P(\mu,\sigma^2)$ and is likely a Normal-Inverse Gamma distribution. $P(Y_{tc})$ is the prior distribution for the CP (Y_{tc}). The BCPD-NIG algorithm computes the marginal likelihood $P(y_1,\dots,y_T|K)$ by integrating over the mean–variance:

$$P(Y_{t1},\dots,Y_{tn}|Y_{tc}) = \iint P(Y_{t1},\dots,Y_{tn}|Y_{tc},\mu,\sigma^2) * P(\mu,\sigma^2)\,d\mu\, d\sigma^2 \tag{16}$$

A change point is detected when the joint probability distribution is maximized:

$$C^* = \text{argmax}\ P(Y_{tc}|Y_{t1},\dots,Y_{tn}) \tag{17}$$

The Bayesian framework integrates prior knowledge or beliefs about parameters, facilitating a richer analysis. This approach yields outputs like posterior distributions that offer clear insights into probabilities, enabling uncertainty assessment in detecting change points.

3.2.4 Limitations of Bayesian approaches for PCD

- Bayesian methods, especially ones involving Markov chain Monte Carlo (MCMC), can be computationally intensive.
- The need to choose appropriate prior distributions can be challenging and might influence the results.
- Ensuring convergence, especially in MCMC-based methods, can be tricky and requires additional diagnostic checks.

Table 2
All Bayesian methods are arranged in chronological sequence.

Ref.	Methods	Limitations
Corradin et al. (2022)	BCPA	Choosing the wrong prior distribution in Bayesian methods can skew the outcomes, as unsuitable priors might result in inaccuracies. Additionally, these methods, particularly those using Markov chain Monte Carlo (MCMC) sampling, demand a lot of computational power, and achieving convergence with techniques like MCMC can be difficult, often necessitating thorough diagnostic processes (Jin, Yin, Zhou, & Horpibulsuk, 2019).
Altamirano et al. (2023)	BOCD	The Bayesian approach enhances pattern change detection by incorporating prior knowledge. Still, it can be computationally demanding, especially with iterative methods like MCMC. Determining the optimal change point threshold in BOCD is challenging and often requires domain expertise, with MCMC convergence sometimes requiring additional diagnostics.
Lu et al. (2023)	BCPD-NIG	The NIG prior effectively handles complex normal distribution parameters for change points but requires expertise and specific tools for Bayesian models, especially hierarchical ones. Its assumption of normality may not always apply, and the choice of NIG or change point priors greatly influences results, with poor selections leading to sub-optimal outcomes.

- Some Bayesian algorithms can be sensitive to their initial conditions, potentially affecting the outcome.

For reference, Table 2 lists various Bayesian methods and outlines their limitations.

3.3 Time–frequency analysis methods for detecting pattern changes in time series

These methods decompose the time series into time and frequency domains, allowing for a deeper understanding of how data patterns evolve. This dual analysis of time and frequency enables identifying both transient and persistent changes in the data (Aminikhanghahi & Cook, 2017). These techniques effectively identify changes in time series, particularly in complex or unstable signals where shifts might occur abruptly and gradually. The following subsections will explore several models that utilize this approach. The final section of this review will address the overarching limitations associated with these methods.

3.3.1 Models based on wavelet transformation

Bozdal, Samie, and Jennions (2021) and Zhang, Zhou, Wen, and Sun (2022) proposed Wavelet-based Change Point Detection (WCPD) by performing wavelet transform of a time-series using:

$$W_y(\alpha,\beta) = \int Y(t) * \kappa^*(\alpha,\beta;t)dt \tag{18}$$

Here, the complex conjugate $\kappa^*(\alpha,\beta;t)$ of the wavelet function $\kappa(\alpha,\beta;t)$ scaled by a factor α and translated by β. The wavelet coefficients $W_y(\alpha,\beta)$ represent the signal's local frequency content. WCPD detects CP by analyzing the time–frequency characteristics of a signal using wavelet transforms (Chen, Zhou, Qiu, & Xu, 2022). A CP is detected when the value of the wavelet coefficients crosses a threshold:

$$|W_y(\alpha,\beta)| > \tau \tag{19}$$

Here, τ represents the threshold value, which can be set based on the specific application requirements of WCPD. Wavelet-based methods offer benefits like enabling multi-resolution analysis to detect high and low-frequency changes and effectively denoising data for robust PCD. They provide a concise data representation and, with fast algorithms, ensure time-efficient computation.

3.3.2 Time–frequency models based on spectral density

Sundararajan and Pourahmadi (2018) and Jiang et al. (2020) proposed a Non-parametric technique to detect CP by calculating power spectral density using:

$$S_y(f) = |Y(f)|^2 \tag{20}$$

Here, Y(f) is the Fourier transform of Y_t, and f is the frequency. A CP is detected when the difference in power spectral density before and after a potential CP is significant:

$$|S_y(f,Y_t) - S_y(f,Y_{t+1})| > \tau \tag{21}$$

In this context, τ denotes the threshold value, adjustable by the application's particular needs. Recently, the application of time–frequency spectral density has become instrumental in detecting change points. The method utilizes a frequency reconstructor, which employs Fourier transforms to shift data into the frequency domain Fan et al. (2023). This shift allows for effective anomaly and pattern change detection by examining the phase and amplitude across different frequency segments. Spectral methods capture periodicities in the frequency domain, potentially missed in the time domain, and focus on dominant frequencies, reducing high-frequency noise for robust PCD (Shafi, Aziz, Din, & Ashraf, 2022). They are often non-parametric, demanding fewer data assumptions.

3.3.3 Limitations of time–frequency based approaches for PCD

- Time–frequency methods can be mathematically and computationally complex.
- Transformation processes, especially wavelets, can sometimes produce edge effects, which might affect PCD near the boundaries of data.
- The choice of parameters, for example, the type of wavelet or window size in a short-time Fourier transform, can influence results, making it crucial to select them appropriately.
- With the ability to analyze changes at multiple scales and frequencies, there is a risk of over-analyzing or over-interpreting insignificant fluctuations.

For reference, Table 3 lists various Time-Frequency methods and outlines their limitations.

3.4 Machine learning methods for detecting pattern changes in time series

These methods utilize sophisticated algorithms to analyze patterns and identify anomalies. These methods learn from historical data to recognize typical patterns and deviations, effectively pinpointing change points. These methods are broadly categorized into two types: supervised and unsupervised. The subsequent section provides a more in-depth analysis of these categories, exploring their distinct characteristics and applications.

3.4.1 Supervised machine learning PCD methods

These methods are effective for identifying pattern changes in sequential data. These methods depend on labeled training data, in which each occurrence is labeled with either the existence or the absence

Table 3
All time–frequency methods are arranged in chronological sequence.

Ref.	Methods	Limitations
Bozdal et al. (2021) Zhang et al. (2022) Chen et al. (2022)	WCPD	The effectiveness of these techniques heavily depends on the choice of wavelet functions, with a high chance of edge effects leading to false detections. They also require significant processing power for large datasets and demand uniform data sampling, which often means extra pre-processing is needed for datasets with uneven intervals.
Jiang et al. (2020) Shafi et al. (2022) Fan et al. (2023)	SDCPD	The effectiveness of these methods is limited by the length of the data sequence, which may lead to missed transient changes. The choice of segmenting functions, as discussed in Section 2.5.3, can impact outcomes, and the assumption of signal periodicity in these methods reduces their suitability for random or aperiodic data sequences.

Table 4
All machine learning methods are arranged in chronological sequence.

Ref.	Technique	Remark
Jin, Chen, Li, Poolla and and Sangiovanni-Vincentelli (2019)	One-Class SVM (OC-SVM)	Proposed a heuristic search approach to identify an optimal combination of input data and hyper-parameters that enhances performance using an OC classifier to detect CP.
Wen and Keyes (2019)	Convolutional Neural Networks (CNN)	Proposed a transfer learning approach that trains a model and then refines it on smaller, diverse datasets.
Kopp, Pevnỳ, and Holeňa (2020)	Random Forests (RF)	Utilizes random forests to derive rules highlighting the distinctions. These rules are then refined and showcased to the user as a unified set of classification guidelines or a corresponding rule with its conditions in a disjunctive normal form.
Barbado, Corcho, and Benjamins (2022)	One-class SVM	Computing metrics related to eXplainable Artificial Intelligence (XAI) and other extracted rules have been proposed.
Lattari, Rucci, and Matteucci (2022)	Long Short-Term Memory (LSTM)	Utilizes LSTM cells to capture the temporal relationships in the input time series and incorporates time-gated LSTM cells to factor in the sampling rate as supplementary data during the learning process.
Londschien, Bühlmann, and Kovács (2023)	Random Forest (RF)	This approach constructs a classifier log-likelihood ratio based on class probability predictions to evaluate several change point configurations
Tang, Xu, Yang, Tang, and Zhao (2023)	Gated Recurrent Units (GRU)	Proposed a GRU-based method to efficiently extract both long-term and short-term patterns, enhancing detection accuracy and aiding users in identifying the origins of anomalies in multivariate time series.
Kloska, Grmanova, and Rozinajova (2023)	Dynamic Time Warping (DTW)	Proposed an Expert Enhanced DTW (E-DTWA) approach based on DTW and integrated a human-in-the-loop feature, prioritizing expert feedback for efficient detection and adaptable retraining.

of a CP. The primary objective of supervised learning is to train a model that can generalize well to new, unobserved data and detect CP precisely (Deldari, Smith, Xue, & Salim, 2021; Pushkar et al., 2022). ML algorithms are versatile, accommodating diverse data types, and can identify intricate patterns that traditional methods might miss. They efficiently manage multivariate time series, recognize correlations across numerous variables, and can depict non-linear associations inherent in the data. These supervised methods often utilize Support Vector Machines (SVM) and Random Forests (RF), among other techniques. Their efficiency extends to managing intricate associations in multivariate contexts. To facilitate comprehension and reference, these machine learning methods are systematically organized in a Table 4, presented in chronological order.

3.4.2 Unsupervised machine learning PCD methods

These methods detect pattern changes in time series data without relying on labeled training data. They analyze datasets to uncover natural structures and anomalies, identifying change points through the inherent properties of the data. These techniques are particularly useful when pre-defined labels or change point examples are unavailable, making them ideal for exploratory data analysis. Such algorithms include Recurrent Neural Networks (RNN) and Dynamic Time Warping (DTW). They are well-suited for complex, multi-dimensional datasets and adept at revealing subtle and less apparent changes. The primary advantage of these methods is their capability to detect new patterns and relationships, providing valuable insights into the underlying dynamics of the data. For ease of understanding and referencing, these methods are arranged in a Table 4, displayed in a chronological sequence.

3.4.3 Limitations of machine learning based approaches for PCD

- ML models, incredibly complex, can over-fit the training data and may not generalize well.
- Hyperparameter tuning and model selection can be time-consuming.
- Training machine learning models, especially on large datasets, can be computationally expensive.
- Many ML models, particularly deep learning ones, can act as black boxes, making it hard to interpret their decisions.

3.5 Hybrid methods for detecting pattern changes in time series

These methods combine the strengths of various techniques to enhance the effectiveness of PCD. These hybrid methods aim to increase the overall performance and robustness of pattern change identification by capitalizing on the strengths of each separate method while compensating for their weaknesses (Pushkar et al., 2022). The Following subsections explore various models that implement this approach, while the final section addresses the limitations of Hybrid Methods.

3.5.1 Hybrid model based on stacked recurrent neural networks (RNN)

Shi and Chehade (2021) proposed a hybrid method based on dual-LSTM framework to detect CP using:

$$\hat{Y}_t = F(Y_t, \theta)$$

$$L_1 = -\frac{1}{tr}\sum_{t}^{tr} Y_t log(\hat{Y}_t) + (1 - Y_t) log(1 - \hat{Y}_t)$$

$$Y_{tc} = inf(t \le \alpha | \hat{Y}_{t \le \alpha} = F(Y_t, \theta) > \tau)$$

In this context, the LSTM tailored for PCD is a binary classifier with two distinct classes. The best parameters, represented by θ are chosen to reduce the binary cross entropy. The training dataset contains tr time points. The CP, denoted by Y_{tc}, is identified as the initial instance when the output $\hat{Y}_t$ exceeds the threshold τ. These techniques effectively manage multivariate time series, recognizing correlations among several variables. They are adept at capturing non-linear relationships within the data.

3.5.2 Model based on Bayesian ensemble algorithm

A Bayesian Estimator of Abrupt change, Seasonal change, and Trend (BEAST) introduced by Zhao et al. (2019) detects sudden changes by decomposing a time series using:

$$\hat{y}(t_i) = f(t_i;\theta) = S(t_i;\theta_S) + T(t_i;\theta_T), i = 1, \dots, n \tag{22}$$

Here, the parameters θ_S and θ_T represent the seasonal and trend components and encapsulate any sudden shifts. Time t is the independent variable, while data y is the dependent variable. Using the dataset $D = t_i, y_{i_{i=1,\dots,n}}$, the goal is to estimate the parameters θ_S and θ_T. These models offer the advantage of combining probabilistic inference with multiple predictive models for improved prediction accuracy and robustness.

3.5.3 Model based on mutual information for PCD

Rezvani, Barnaghi, and Enshaeifar (2021) proposed a hybrid method to detect change point by applying Lagrangian Multipliers followed by mutual information:

$$MI(q_i, q_{i+1}) = \sum_{x\in q_i}\sum_{y\in q_{i+1}} P(x,y) log\frac{P(x,y)}{P(x)P(y)} \tag{23}$$

Here, a matrix Q_{ij} of $(m \times 8)$ is formed after the sliding window application (refer 2.5.3) , where m is the row count of matrix Q_{ij}. $P(x)$ represents the probability of element x in row q_i of matrix Q_{ij}, $P(y)$ denotes the probability of element y in row q_{i+1}, and $P(x,y)$ signifies the joint probability of elements x and y. To detect change points, triangle fluctuations are identified, a decline following an increase or an ascent following a dip, within every three consecutive mutual information values, specifically among $MI(q_{i-1}, q_i)$, $MI(q_i, q_{i+1})$ and $MI(q_{i+1}, q_{i+1})$ for $i - 1, \dots, (m-2)$. These methods effectively identify dependencies between time series variables, enhancing the accuracy of detecting shifts in patterns.

3.5.4 Hybrid approach combining singular spectrum analysis and deep learning

Gupta, Wadhvani, and Rasool (2022) introduced a method combining SSA with an Auto-encoder (CPD-AE) for PCD by using:

$$T_{val} = \frac{\lambda}{100} \times \frac{\|\hat{Y}_{ti} - Y_{ti}\|}{n} \tag{24}$$

Here, n is the number of input features, and λ is a parameter that can be altered based on the implementation of PCD. The encoder's hidden layer neurons extract the components, and the Mean Absolute Error between the noise-free input and the auto-encoder's output is calculated. The threshold value is selected using the Eq. (24). This novel hybrid approach is tailored for real-time sequential data. It adeptly identifies intricate changes, ensuring reduced false alarm rates, and integrates data pre-processing to enhance real-time detection accuracy. In the pre-processing phase, a recursive form of SSA is employed on the normalized Hankel matrix to diminish noise (refer Section 2.5.4). Subsequently, auto-encoders are utilized to carry out the detection of pattern changes.

3.5.5 Hybrid approach integrating ARMA and CUSUM

Lee, Lee, and Moon (2020) proposed a Hybrid Method based on location and scale-based cumulative sum LSCUSUM to detect change point using the ARMA Model:

$$Y_t = c + \sum_{i=1}^{p}\phi_i Y_{t-i} + \sum_{j=1}^{q}\phi_j \epsilon_{t-j} + \epsilon_t \tag{25}$$

Here, ϕ_i, ϕ_j is the parameter of the Autoregressive and Moving Average part, c is the constant, and ϵ_t is the white noise error terms. Support Vector Regression (SVR), a type of SVM, is used for regression challenges. In SVR, the aim is to find the function $f(Y)$ that has at most ϵ deviation from the obtained targets Y for all the training data and, simultaneously, is as smooth as possible. The intuition behind SVR involves constructing a tube around the regression line with a radius of ϵ. As long as the predictions fall within this tube, no penalty is given to the model's loss function. A significant deviation of the actual residuals from the predicted residuals may indicate a change point. If r_t are the residuals from the ARMA model at time t, and $\hat{r}_t$ are the predicted residuals from the SVR model, then for some threshold τ, a change point is detected if:

$$|r_t - \hat{r}_t| > \tau \tag{26}$$

The choice of τ depends on the variance of the residuals and the desired sensitivity of the PCD mechanism.

3.5.6 Hybrid non-parametric, distribution-free method based on wasserstein distance

Faber, Corizzo, Sniezynski, Baron, and Japkowicz (2022) introduced LIFEWATCH, a method for detecting new and known changes in univariate and multivariate time series data. This method employs the Wasserstein distance, also known as the Earth Mover's Distance, a metric based on probability distributions in a metric space. It effectively measures the cost to transform one distribution into another (Verzelen, Fromont, Lerasle, & Reynaud-Bouret, 2023). For two distributions P and Q in a d-dimensional space, the p-Wasserstein distance is defined as the infimum across all joint distributions γ of (X, Y), where X and Y follow P and Q, respectively. This distance is calculated as:

$$W_p(p,q) = (\mathbf{inf}_{\gamma\in\Gamma(P,Q)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \mathbf{dist}(x-y)^p d\gamma(x,y))^{\frac{1}{p}} \tag{27}$$

Here, dist (x,y) usually being the Euclidean distance. In LIFEWATCH, this distance helps to determine whether new data points or mini-batches belong to the current distribution or represent a change point. A CP is detected when the distance between new data and the current distribution surpasses a set threshold. Using a similar concept, Cheng, Aeron, Hughes, Hussey, and Miller (2020) introduced a novel algorithm for PCD and time series segment clustering (TSSC) in time series analysis. This approach focuses on decomposing data streams into homogeneous segments for PCD and grouping similar nonadjacent segments for TSSC. The method is grounded in the recent theoretical advances related to the Wasserstein two-sample test, allowing for a distribution-free, unsupervised approach applicable in offline and online settings.

3.5.7 Limitations of hybrid methods for PCD

- Integrating multiple methods can make the model more complex, demanding more resources and computational power.
- Using multiple methods together could result in over-fitting, particularly if they are not properly regularized.
- Designing, implementing, and tuning hybrid methods can require more effort than using a singular method.
- If one component of the hybrid method fails, it might affect the overall performance.

For reference, Table 5 lists various Hybrid methods and outlines their limitations.

4 Performance evaluation

This section outlines how the algorithm's performance is assessed using various datasets. It specifies the performance metrics used for assessment, outlines the experimental setup, presents the results and discussions, and concludes with a computation of each algorithm's complexity.

Table 5
Overview of chronologically arranged hybrid methods with associated limitations.

Ref.	Methods	Limitations
Zhao et al. (2019)	BEAST	The BEAST method faces limitations in handling large high-resolution datasets due to its computationally intensive Monte Carlo sampling. It also struggles with data quality issues, where noisier data reduces pattern change detection probability, and its univariate approach limits simultaneous analysis of multivariate time series.
Lee et al. (2020)	LSCUSUM	The method requires full dataset pre-processing and fixed window settings, limiting real-time application and adaptability for multivariate and non-stationary time series, reducing its versatility in diverse scenarios.
Shi and Chehade (2021)	Dual-LSTM	The method needs extensive training data and risks over-fitting, requires high computational power for large datasets, and struggles with rapid changes in non-stationary series, requiring precise parameter tuning for optimal sensitivity and specificity.
Rezvani et al. (2021)	CPD-MI	The method requires an entire dataset for pre-processing, restricting it to offline use. It depends on fixed sliding window sizes, reducing flexibility, particularly for multivariate and non-stationary time-series data.
Gupta et al. (2022)	CPD-AE	The method requires a specific number of samples for training and has limited effectiveness in detecting frequency and auto-correlation changes, coupled with higher computational complexity.
Faber et al. (2022)	LIFEWATCH	The main challenge lies in effectively dealing with scenarios where numerous distributions are changing rapidly. Although robust in many situations, the technique's performance could be limited in environments with rapidly and frequently changing data distributions.

Table 6
Summary of datasets for the evaluation of the pattern change detection algorithms.

Dataset No.	Dataset	Length of sample	Features per sample	No. of sample	No. of CP	Type of series
1	Change in Mean & Variance	2000	1	200	2000	Uni-variate
2	Alter Auto-correlation	2000	1	200	2000	Uni-variate
3	Room & Occupancy	509	4	1	11	Multi-variate
4	HASC 2010 & corpus	11,852	3	18	90	Multi-variate

4.1 Datasets

To evaluate the performance of several methods, two synthetic datasets are constructed and two real world datasets have been chosen. Table 6 provides a complete summary of all the datasets for reference.

4.1.1 Dataset 1: Fluctuating average and dispersion in sequential data pattern

This artificial dataset showcases both shifts in mean and fluctuations in variance. An Auto-Regressive (AR) model generates the time series, described by the following equation, with 200 samples and 2000 observations:

$$y_i = 0.3 * y_{i-1} + 0.5 * y_{i-2} + \epsilon_i \tag{28}$$

Here, ϵ_i denotes Gaussian noise with a mean of 0.3 and a standard deviation of 0.5. Outliers are incorporated at uniform intervals of 200 timestamps. In Eq. (28), the correlation coefficients of the lagged vector of time series at lag one and lag two represented as y_{i-1} and y_{i-2} are taken as 0.3 and 0.5, respectively. The outliers are combined with random values, and the interval between subsequent shifts in the time series is determined randomly.

4.1.2 Dataset 2: Altering auto-correlation in sequential data pattern

This synthetic dataset highlights a shifting auto-correlation. An AR model is employed to create a time series comprised of 2000 observations with 200 samples having a total of 2000 CP, as depicted in the following equation:

$$y_i = \beta_1 * y_{i-1} + \beta_2 * y_{i-2} + \epsilon_i \tag{29}$$

Here, ϵ_i represents Gaussian noise with a variance ranging from 0.3 to 1.2 and a constant mean of 0.5. Outliers are integrated at regular intervals of 200 timestamps. The AR model provided in Eq. (29) is utilized to generate this dataset. The coefficients β_1 and β_2 are altered within the range of [0-1]. To generate a strongly linked time series, both coefficients are assigned to 1, and to obtain no correlation, both are set to 0. The time between two modifications is chosen at random.

4.1.3 Dataset 3: Room occupancy

The dataset comprises 18,143 room occupancy observations taken at one-hour intervals throughout the day (Van den Burg & Williams, 2020). These measurements include carbon dioxide level, temperature, light, humidity, and light, all affecting room occupancy. The data is compressed by picking every sixteen observations to minimize the size of the dataset and generate a more manageable time series. This technique yields a final time series of 509 observations, retaining the vital information while making the dataset more appropriate for analysis and processing.

4.1.4 Dataset 4: HASC2010corpus

The HASC Challenge 2010 dataset includes detailed sequences with approximately 11,852 three-axis accelerometer readings per sequence, forming a multivariate time series dataset with three features (Anguita, et al., 2013). It covers recordings of six distinct activities -'stay,' 'walk,' 'jog,' 'skip,' 'stairs up,' and 'stairs down,' captured at a 100 Hz sampling rate. This complex dataset has been used to evaluate the performance of seven individuals sequentially completing these activities. Data was gathered using sensors in devices like the iPhone/iPod Touch and ATR's WAA-series.

4.2 Performance measures

Evaluating the performance of detection methods for pattern shifts in time-series data is crucial. Table 7 details the effectiveness metrics

Table 7
Summary of all datasets chosen for the evaluation of the PCD algorithms.

Performance Measure	Formula	Description
Precision	$PRE = \frac{TP}{FP+TP}$	Precision, indicates the proportion of correctly identified change points (CP) among all detected CPs.
Recall	$REC = \frac{TP}{FN+TP}$	Recall, also known as sensitivity or True Positive Rate, measures the fraction of actual CP detected by the algorithm from all observed behavior shifts.
F1 Score	$F1 = 2 \times \frac{PRE \times REC}{PRE+REC}$	The F1 score, ranging from 0 to 1, represents the balance between precision and recall. It offers a comprehensive view of algorithm performance.
Mean Absolute Delay	$MAD = \frac{T_d}{\#TP}$ $T_d = \sum_{i \in \text{TP class}} \lvert \hat{CP_i} - CP_i \rvert$	MAD calculates the average absolute difference between actual and observed CP. It represents the expected deviation between predicted and actual CP locations from the true positive class.
Area Under the Curve	$AUC = \int_0^1 REC * f d(f)$ $f = \frac{FP}{FP+TN}$	The AUC evaluates the performance of a PCD algorithm, representing the likelihood it correctly identifies a CP over non-CPs. It plots the algorithm's sensitivity against the false positive rate.
Accuracy	$ACC = \frac{TP+TN}{TP+FP+TN+FN}$	The accuracy measure indicates the percentage of positive and negative change-points correctly categorized by the algorithm.

of the PCD algorithm. The True Positives (TP) represent the number of instances in which the algorithm successfully detected a change in pattern. This means there was, in fact, a change in the pattern within the sequential data, and the system identified the shift accurately. False Positives (FP) are the number of times the algorithm inaccurately identifies a pattern change that does not exist in the data. In other words, the sequential data do not show any pattern changes, but the algorithm mistakenly identifies one. True Negatives (TN) are the number of times the algorithm correctly determines there is no CP in the data, signifying that the sequential data has no pattern changes, and the algorithm detects this absence accurately. False Negatives (FN) are the number of actual CPs the algorithm misses, meaning there is a change in the pattern in the sequential data. Still, the algorithm fails to identify it correctly. The performance measure, Precision, is synonymous with Positive Predictive Value. A higher precision indicates that the algorithm is more adept at accurately identifying behavioral changes from quantitative data. Similarly, a greater specificity or recall suggests the algorithm excels at reducing false alarms. An algorithm that performs perfectly achieves a score of 1 on the AUC (Area Under the Curve) metric, while an algorithm that makes random guesses scores 0.5. AUC is a preferred metric for evaluating PCD algorithms due to its comprehensive assessment capabilities across diverse thresholds. Furthermore, it facilitates straightforward comparisons between different algorithms and configurations, proving invaluable for algorithm choice and fine-tuning.

4.3 Experimental setup

Different algorithms require unique types of data to detect CP accurately. Some algorithms are designed to detect change points in real-time, while others operate in an offline mode. Additionally, there are algorithms capable of real-time pattern change detection, yet they depend on batch pre-processing of data. For our study, we provided a standardized platform to ensure an equitable evaluation of these varied algorithms. We adopted an offline setup for all algorithms, including those capable of real-time operation, to maintain consistency in testing conditions. Furthermore, the data input to each algorithm has been pre-processed: it is standardized using z-score normalization and cleansed of noise through singular spectrum analysis (see Section 2.5.4), ensuring that the data is prepared for the detection process. In the course of comparison, it was discovered that specific algorithms fail to identify changes in complex datasets, such as those exhibiting heteroscedasticity or multi-modality (refer Section 2.5.1, 2.5.2) . Additionally, some algorithms cannot detect shifts within multivariate sequential data. Consequently, these algorithms were excluded from specific comparative analyses, specifically in datasets 3 and 4, because of their inability to handle such data complexities.

Each method has some unique parameters, and the algorithms we are examining have been meticulously fine-tuned. We will outline each parameter and the corresponding values for assessing the algorithms. A key parameter is the window size (W), which dictates the segmentation of the time series. The right balance is imperative: too many segments can lead to over-fitting, and too few to under-fitting. Based on the literature and our computational capacity, we have set W to 5% of the sample size, resulting in 100 segments for Datasets 1 and 2, 25 for Dataset 3, and 590 for Dataset 4. This compromise ensures sufficient data per segment for detecting changes without overloading the algorithms.

To control model complexity and prevent over-fitting in PLR, a penalty term (λ = 1.23) is implemented as a default (Muggeo et al., 2008). This regularization parameter can either constrain the size of the regression coefficients or the number of model segments. The CUSUM algorithm uses a reference value (k), which is half the dataset's standard deviation, to determine the cumulative sum of deviations. A slack variable (α), set at five times the standard deviation, adjusts the algorithm's sensitivity to variations; with a larger α, the algorithm reduces the sensitivity to minor fluctuations (De Oca et al., 2010). For the GLRT, a cap is placed on the number of change points—10 for datasets 1 and 2, 11 for dataset 3, and 90 for dataset 4 (Su et al., 2014). In kernel-based methods like KLIEP, the bandwidth (b) is set according to the median of all pairwise point distances within the dataset, guiding the smoothness of the density function estimation. The Number of Basis Functions (B) is fixed at 10, balancing the model's fitting capacity against the risks of over-fitting or under-fitting (Hushchyn & Ustyuzhanin, 2021). KCPD employs a Gaussian kernel, impacting the algorithm's PCD accuracy due to its influence on feature space mapping. Bayesian approaches use a Hazard Function (γ) set at 0.002, indicating an anticipated change approximately every 500 data points, presuming a Poisson distribution for CP occurrence (Corradin et al., 2022).

For time–frequency analysis, the Haar Wavelet is utilized due to its compatibility with the data's attributes since selecting the appropriate wavelet is crucial for change detection accuracy. Bozdal et al. (2021). ML techniques like Random Forest use 100 trees to ensure model robustness, balanced against computational efficiency, with a maximum depth of four and a minimum sample split of three to avoid

Table 8
Comparison of multiple algorithms on dataset 1 (changing mean and variance).

Category	Methods	Performance measures					
		PRE	REC	F1	MAD	AUC	ACC
Statistical Parametric Approaches	PLR-CPD	0.894	0.874	0.884	1.49	–	99.89
	CUSUM-CD	0.884	**0.953**	0.917	**1.26**	–	99.91
	GLRT-CD	0.900	0.903	0.901	1.56	0.790	99.90
Statistical Non-parametric Approaches	KLIEP	0.824	0.678	0.744	4.07	–	99.77
	uLSIF	0.696	0.849	0.765	3.67	–	99.74
	KCPD	0.863	0.628	0.727	3.51	0.791	99.76
Bayesian Methods	BCPA	0.782	0.772	0.777	3.35	–	99.78
	BOCD	0.829	0.838	0.833	5.03	–	99.83
	BCPD-NIG	0.845	0.793	0.818	3.48	–	99.82
Time-Frequency Methods	WCPD	0.835	0.902	0.867	2.43	0.912	99.86
	SDCPD	0.799	0.856	0.826	3.65	0.899	99.82
ML Supervised Methods	SVM-CD	0.647	0.823	0.724	4.91	0.721	99.69
	RF-CPD	0.852	0.879	0.865	3.16	0.879	99.86
ML Unsupervised Methods	CPLSTM	0.692	0.851	0.763	3.52	0.910	99.74
	GRU-CD	0.795	0.883	0.836	4.41	0.870	99.83
	DTW-CD	0.841	0.940	0.888	4.57	0.862	99.88
Hybrid Methods	Dual-LSTM	0.792	0.866	0.828	5.87	0.890	99.82
	BEAST	0.836	0.877	0.856	5.32	0.834	99.85
	CPD-MI	0.883	0.865	0.874	4.63	–	99.88
	CPD-AE	0.902	0.935	0.918	5.46	**0.942**	99.92
	LSCUSUM	0.860	0.928	0.893	5.35	0.891	99.89
	LIFEWATCH	**0.908**	0.931	**0.924**	4.01	0.931	**99.92**

over-fitting and accommodate noise. Neural networks, including LSTM and GRU, consist of three layers with 128, 64, and 32 neurons, respectively, using ReLU activation and a 20% dropout rate in hidden layers, optimized with Adam at a learning rate of 0.01 (Tang et al., 2023). DTW employs the Euclidean distance metric for dissimilarity measurement (Kloska et al., 2023). BEAST utilizes 1000 MCMC iterations, with a 100-period burn-in to stabilize the process and a thinning interval 10 to manage auto-correlation (Zhao et al., 2019). MI-based methods leverage Equal-Width Binning or KDE for discretization in calculating mutual information (Rezvani et al., 2021). Lastly, LSCUSUM accounts for process drift using a Drift Allowance set to three times the standard deviation, accommodating normal variations without indicating change points (Lee et al., 2020).

The evaluation of PCD algorithms was conducted on the High-Performance Computing (HPC) system at Maulana Azad National Institute of Technology (MANIT). This advanced HPC facility features multiple computing nodes with sophisticated multi-core processors. The nodes are connected through a high-speed network, ensuring efficient data handling. The HPC is capable of managing large datasets and complex computations. Additionally, the system includes specialized hardware like NVIDIA GPUs, enhancing its capability for computationally intensive tasks, making it well-suited for PCD analysis.

4.4 Results and discussions of pattern change detection algorithms

Identifying changes in datasets, especially those with fluctuating mean and variance, is a significant challenge in change point detection. The complexity of these datasets requires sophisticated algorithms to detect the moments of change accurately. The subsequent results provide a detailed analysis of the performance of various PCD algorithms on dataset 1, as depicted in Table 8, which is notable for its varying mean and variance. Each point in the analysis highlights different aspects of the results, offering a deeper insight into the strengths and weaknesses of each algorithm in this particular context.

- Hybrid and statistical methods excel in detecting change points in data with shifting mean and variance, as seen in the F1 and Accuracy matrices in Table 8. While statistical techniques like CUSUM and likelihood ratio tests harness data's probabilistic nature, hybrid methods integrate these with other approaches for a comprehensive analysis. Conversely, methods like non-parametric and Bayesian may not inherently address simultaneous changes in mean and variance or might demand more intricate assumptions, reducing their efficiency.
- The Friedman test evaluates the performance of different algorithms, with the rankings indicating that hybrid methods often perform the best, particularly LIFEWATCH, which ranks highest. Statistical methods rank second, followed by Time–Frequency methods (See in Fig. 5). Surprisingly, SVMs under-perform, as their reliance on a linear boundary in the feature space limits their ability to detect non-linear shifts in data distribution and temporal changes.
- Statistical Parametric Approaches and Hybrid Methods effectively detect change points in data with varying mean and variance. However, there is a trade-off between MAD and Accuracy for both techniques. While statistical methods might have lower accuracy than hybrid ones, they boast a smaller MAD because of their simple and specific design, excelling when data aligns with their assumptions. They adeptly handle changes in mean and variance. Conversely, hybrid methods are versatile in dealing with different data patterns, but this comes at the cost of increased complexity and computational requirements, which may result in delays in PCD.
- High AUC values over 0.9, as seen with CPD-AE, LIFEWATCH, and LSTM, suggest exemplary performance. CPD-AE stands out with the top AUC at 0.942, likely due to its enhanced feature extraction capabilities. Conversely, SVM, with an AUC of 0.721, is probably lower because it relies on linear boundaries in the feature space, hindering its capacity to discern non-linear data shifts. The AUC of multiple methods is not calculated due to the inability to determine the True Positive Rate (TPR) vs. False Positive Rate (FPR) at varying threshold levels. This underscores the importance of thorough evaluation metrics when assessing model performance.
- Techniques like SVM, LSTM, and uLSIF often have a higher rate of false alarms, as indicated in the Table 8. Although these

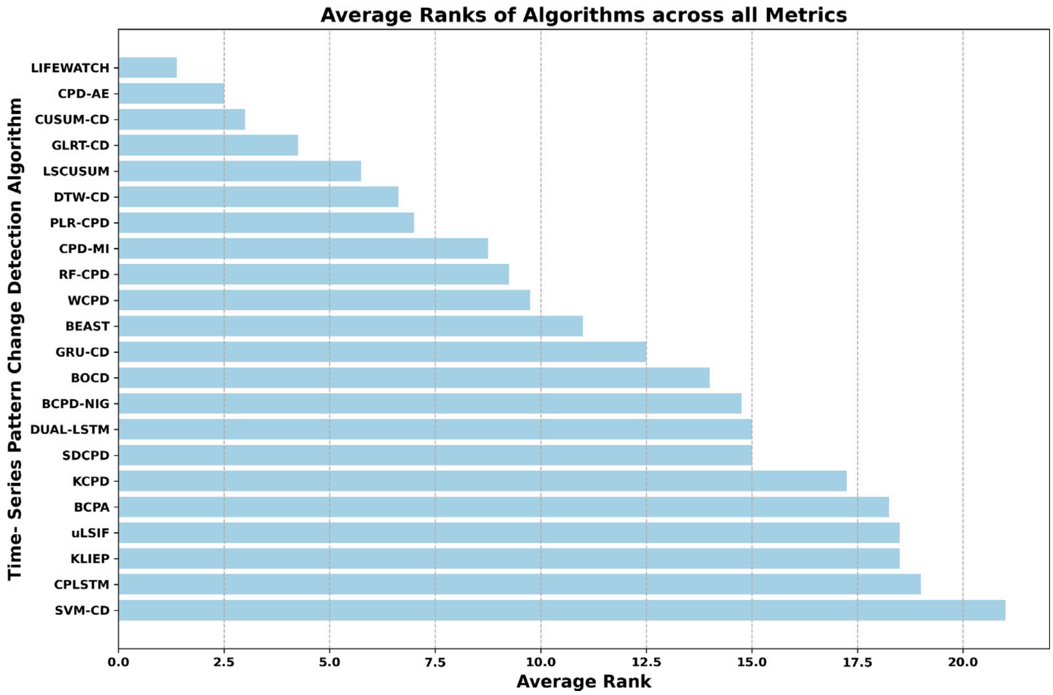

Fig. 5. Friedman test Ranks of PCD algorithms.

methods demonstrate higher F1 scores, they are not preferred over methods that commit fewer Type 1 errors, directly affecting the precision value. When selecting an algorithm, the impact of Type 1 errors should be carefully considered, mainly because of their potential importance in specific applications.

While many techniques are adept at identifying alterations in mean and variance, they often fall short when faced with more intricate shifts in data attributes, such as changes in auto-correlation. Not all techniques have the capability to recognize these subtle variations. Specifically, statistical parametric methods like PLR, CUSUM, and GLRT struggle to detect these changes because they primarily focus on abrupt or cumulative modifications and may not capture the underlying temporal dependencies intrinsic to auto-correlation. These methods generally assess how the data deviates from expected patterns in distribution. While mean and variance are directly tied to a distribution's properties, auto-correlation pertains to the relationship and dependencies between data points separated by specific time lags. Likewise, a hybrid approach based on mutual information (MI) also faces challenges. This is primarily because MI gauges the information shared between two variables and may not be finely tuned to discern subtle shifts in auto-correlation patterns within a single variable. The insights presented in Table 9 provide a comprehensive comparison of dataset 2, explicitly highlighting the alterations in auto-correlation. The following discussion will delve deeper into the key takeaways:

- The effectiveness of each algorithm decreases compared to their results on datasets with changes in mean and variance, as seen in Table 9. This reduction in performance can be attributed to the complex nature of datasets involving correlation changes and the inherent difficulties in detecting subtle differences.
- Given the subtle differences in the F1 scores across different algorithms, it is instructive to compute the geometric mean (G-mean). It acts as a comprehensive metric, especially when dealing with imbalanced datasets, as it offers a harmonized measure between two values. Mathematically, the G-mean is expressed as:

$$G_m = \sqrt{t \times f} \tag{30}$$

Here, t and f are true and false positive rates. Upon inspecting Fig. 6, it becomes evident that the Time frequency, Hybrid, and ML methods outperform other techniques, exhibiting balanced quantiles, underscoring the algorithm's stability. Conversely, methods like BCPD-NIG, and DTW display significant fluctuations in their F1 scores, suggesting they might not be the best fit for detecting shifts in such intricate datasets. Furthermore, the unpredictable patterns seen in the Figure for non-statistical parametric methods suggest inherent uncertainties, especially in specific quantiles, making them less suitable for the intended task.
- Considering all the metrics, a Friedman test is used for the overall ranking. The results reveal that the time–frequency method generally boasts the top performance with the lowest average rank, followed by the hybrid method. Notably, SDCPD stands out with the lowest value of 1.17, followed closely by CPD-AE at 3.50. Subsequent in performance are the ML methods, trailed by time–frequency and Bayesian techniques. Among the statistical non-parametric methods, DTW-CD, with a highest value of 13.42, is identified as the least performing within its category.
- In dataset 1, there is a significant discrepancy between the MAD of Hybrid and statistical methods; however, dataset 2 shows that the MAD of all algorithms is within a comparable range.

Dataset 3, which consists of multivariate time series data, focuses on room occupancy. Several algorithms, including statistical parametric methods like PLR, CUSUM, and GLRT, cannot detect change points for multivariate datasets due to their inherent limitations in handling multidimensional data. These methods primarily operate on the assumption of univariate data distributions and fail to account for the inter-dependencies and complex relationships in multivariate datasets. As a result, when applied to multivariate time series, their effectiveness diminishes, leading to sub-optimal PCD. Similarly, hybrid MI-based methods struggle to detect changes in multivariate time series due to their reliance on mutual information calculations often optimized for pairwise relationships. In multivariate datasets, the complexity increases as the number of inter-variable relationships grows exponentially. These MI-based methods may not adequately capture the intricate dependencies and interactions among multiple variables, thus

Table 9
Comparison of various algorithms on dataset 2 (Altering Auto-correlation in Sequential Data pattern).

Category	Methods	Performance measures					
		PRE	REC	F1	MAD	AUC	ACC
Statistical Non-parametric Approaches	KLIEP	0.735	0.634	0.681	3.86	–	99.70
	uLSIF	0.764	0.629	0.690	3.79	–	99.72
	KCPD	0.789	0.645	0.709	**2.44**	0.743	99.74
Bayesian Methods	BCPA	0.765	0.663	0.710	2.91	–	99.73
	BOCD	0.761	0.680	0.718	2.98	–	99.73
	BCPD-NIG	0.784	0.678	0.727	3.71	–	99.75
Time-Frequency Methods	WCPD	0.854	0.755	0.801	2.98	0.806	99.81
	SDCPD	0.883	**0.795**	**0.837**	3.15	0.832	**99.85**
ML Supervised Methods	SVM	0.735	0.677	0.705	4.53	0.789	99.72
	RF	0.779	0.684	0.728	3.68	0.812	99.75
ML Unsupervised Methods	LSTM	0.831	0.739	0.782	3.00	0.823	99.79
	GRU	0.854	0.745	0.796	4.03	0.834	99.81
	DTW	0.686	0.594	0.636	4.53	0.738	99.66
Hybrid Methods	Dual-LSTM	0.865	0.751	0.804	4.09	0.807	99.82
	BEAST	0.813	0.715	0.761	4.43	0.843	99.78
	CPD-AE	**0.892**	0.778	0.831	5.20	**0.849**	99.84
	LSCUSUM	0.828	0.562	0.669	4.45	0.816	99.72
	LIFEWATCH	0.885	0.768	0.822	5.27	0.831	99.83

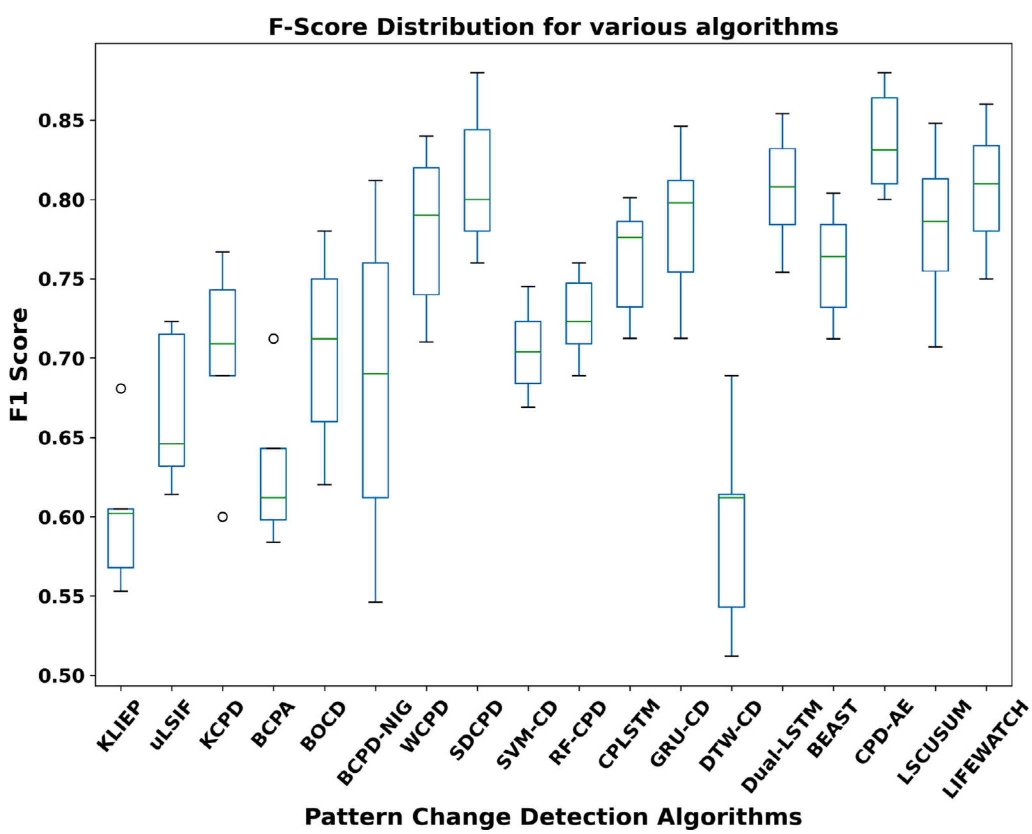

Fig. 6. Geometric mean Plot for F1 score of several PCD algorithms.

leading to oversights or inaccuracies in PCD for multivariate time series. Hence, these methods are omitted from the comparison. Only the techniques capable of processing multivariate time series are compared with the outcomes presented in Fig. 7. The insights from this comparison are as follows:

- Among all categories of algorithms, hybrid and ML-based methods consistently surpass others in performance. Fig. 7 shows that both algorithms display a balanced performance, particularly the hybrid method like CPD-AE. This method ensures that all performance indicators are well-balanced, achieving the highest accuracy. However, time–frequency methods, despite their high precision, are unstable. The potential instability could stem from an overly high precision without a corresponding high recall, which might indicate the algorithm's tendency to miss out on true positive change points, focusing excessively on avoiding false positives. This trade-off can reduce the overall effectiveness of the method in diverse scenarios.

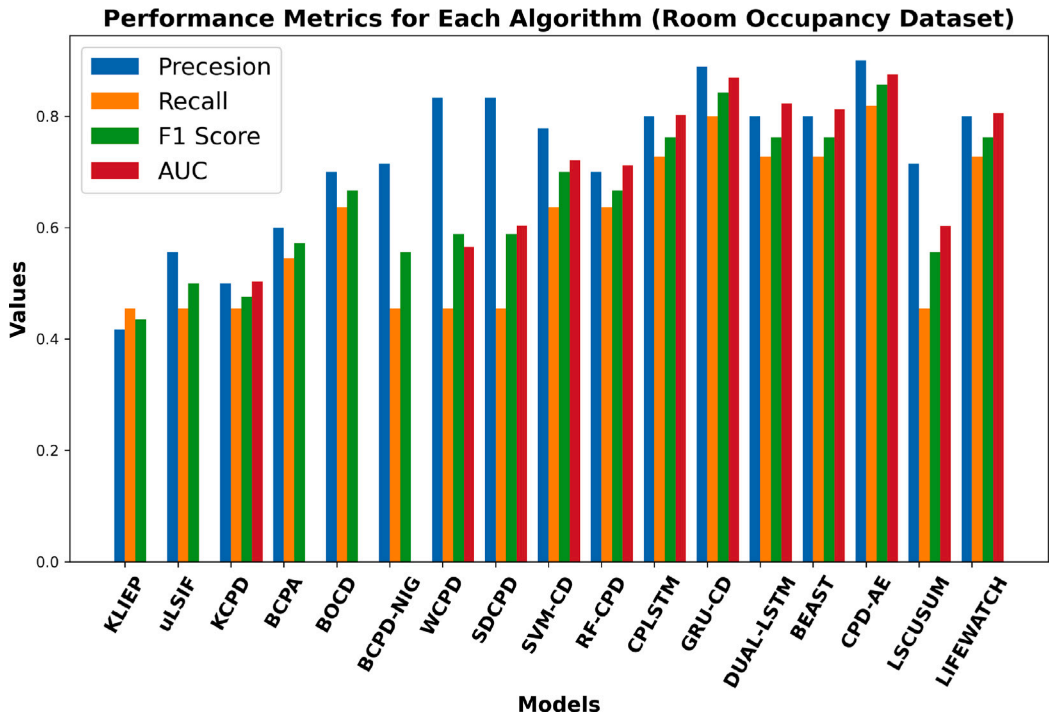

Fig. 7. Performance measure for dataset 3 (Room occupancy) of multiple Pattern change detection algorithms.

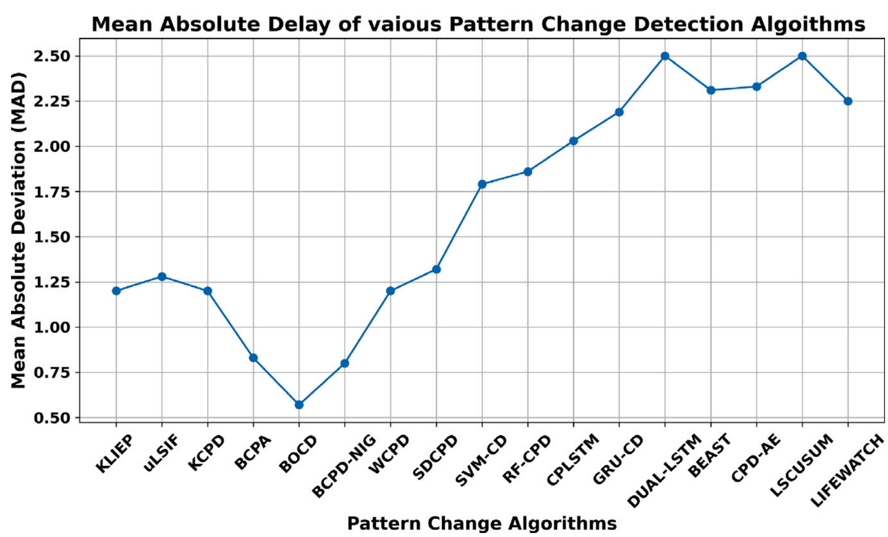

Fig. 8. Mean Absolute delay of PCD algorithms for dataset 3 (Room occupancy).

- Mean Absolute Deviation (MAD) graph (see Fig. 8) reveals that hybrid methods experience a delay in promptly detecting change points, resulting in a higher MAD value. The significance of MAD varies depending on whether the application is online or offline, with online methods inherently exhibiting some delay. In specific contexts, such as ECG analysis, a higher MAD might be acceptable or even expected due to the application-specific nature of the parameter.
- An AUC graph is presented in Fig. 9 for a more in-depth analysis. This graph displays the AUC for each algorithm category, emphasizing the top-performing algorithm within each category. The hybrid, CPD-AE, and ML methods, like GRU and RF, stand out with the highest area under the curve. Their superior performance can be attributed to their ability to adaptively learn complex patterns and capture intricate relationships in the data. Their models are more flexible and can often generalize better on unseen data points. Bayesian methods, represented by BOCD, come next in the sequence. While they provide a probabilistic framework that can be robust in certain scenarios, their assumptions might not always align with the underlying data distribution. Time–frequency methods, such as SDCPD, are observed at the tail end. Their performance might be hindered due to their reliance on fixed-frequency components, which may not always capture abrupt changes or non-stationarities in the data, making them less adaptive to rapid variations.

For an in-depth evaluation of the PCD algorithm, Dataset 4, with its complex structure featuring over 11,852 three-axis accelerometer readings for human activity tracking, has been opted for pattern change detection. As mentioned in the previous section, specific algorithms have been omitted from the comparison due to their inability to detect

Table 10
Comparison of several algorithms on dataset 4 (HASC2010corpus)

Category	Methods	Performance measures					
		PRE	REC	F1	MAD	AUC	ACC
Statistical Non-parametric Approaches	KLIEP	0.556	0.678	0.610	3.85	–	99.94
	uLSIF	0.561	0.711	0.627	3.13	–	99.96
	KCPD	0.573	0.700	0.630	2.54	0.791	99.95
Bayesian Methods	BCPA	0.586	0.722	0.647	1.85	–	99.97
	BOCD	0.557	0.756	0.642	**1.76**	–	99.96
	BCPD-NIG	0.612	0.789	0.689	2.11	–	99.97
Time-Frequency Methods	WCPD	0.557	0.867	0.678	1.74	0.912	99.97
	SDCPD	0.516	0.889	0.653	3.50	0.899	99.96
ML Supervised Methods	SVM-CD	0.556	0.833	0.667	3.73	0.721	99.95
	RF-CPD	0.569	0.822	0.673	2.64	0.879	99.97
ML Unsupervised Methods	CPLSTM	0.622	0.767	0.687	3.48	0.863	99.96
	GRU-CD	0.631	0.778	0.697	4.89	0.871	99.97
Hybrid Methods	Dual-LSTM	0.685	0.844	0.756	5.53	0.891	99.96
	BEAST	0.655	0.800	0.720	5.42	0.834	99.95
	CPD-AE	**0.757**	0.900	0.822	4.81	0.922	99.98
	LSCUSUM	0.608	0.689	0.646	5.71	0.915	99.97
	LIFEWATCH	0.739	**0.944**	**0.829**	4.12	**0.931**	**99.98**

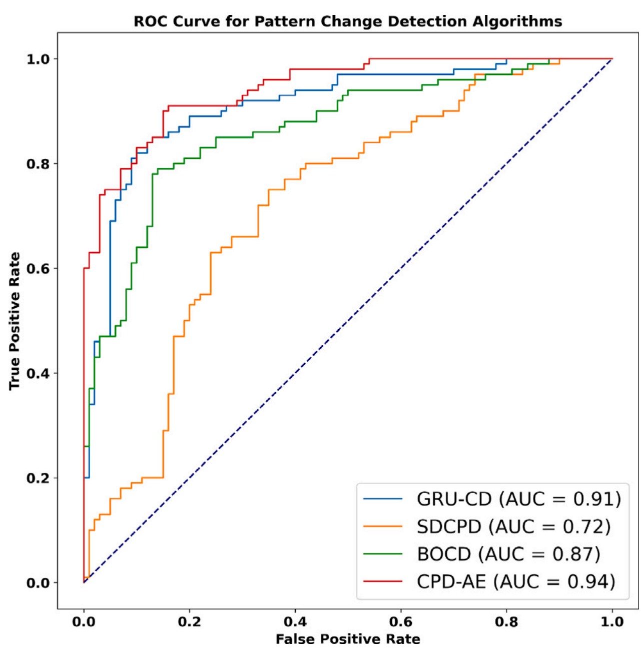

Fig. 9. AUC-ROC curve for dataset 3 (Room occupancy) of PCD algorithms.

change points in multivariate time series data. The chosen evaluation metrics and their values are detailed in Table 10. The key findings from this comparison are outlined below:

- In a large, multidimensional dataset, the performance of all algorithms decreases slightly. However, hybrid methods consistently outperform others across multiple algorithm categories. As shown in Table 10, hybrid approaches, especially those like LIFEWATCH based on Wasserstein Distance, exhibit a well-rounded performance, particularly regarding the F1 score. LIFEWATCH maintains a balance across all performance metrics, leading to superior accuracy. In contrast, while CPD-AE shows high precision, its effectiveness falls short of LIFEWATCH. This limitation may arise from CPD-AE's focus on high precision without equally high recall, potentially causing missed accurate positive detections in favor of avoiding false positives. Such a balance issue can compromise the method's utility in various scenarios.
- Fig. 10 depicts the running times of various algorithms. It highlights that ML methods such as CPLSTM and GRU-CD have longer run times. This is attributed to their intricate, multilayered structures and the significant computational demands required for training. It also shows that hybrid models like DUAL-LSTM exhibit extended run times, stemming from their complex design and the need for extensive temporal data processing over multiple iterations. In contrast, LIFEWATCH, a hybrid method, requires less running time, likely because of its effective use of the Wasserstein distance for data distribution comparison and a less demanding computational model than CPD-AE's deep learning architecture. BOCD, an online Bayesian method, requires the lowest run time to be optimized for real-time data analysis efficiency. Its speed comes from an incremental update process that sequentially processes data and updates the model as needed, reducing the need to reprocess the entire dataset or rely on complex structures, thus enabling quicker execution.
- Regarding AUC, hybrid approaches, particularly LIFEWATCH, stand out with an AUC of 0.931, attributed to its effective multidimensional analysis using the Wasserstein distance for detecting distribution changes. CPD-AE is a close second with 0.922 AUC, indicative of its precision-centric strategy, though it may compromise recall. A significant decrease in AUC is apparent when examining methods such as DUAL-LSTM, GRU, and BEAST. This decline suggests possible challenges in attaining an optimal balance between precision and recall or handling the complexities of multivariate data. Methods with lower AUCs, such as LSTM and RF, struggle more with the dataset's complexities, possibly overlooking subtle shifts in patterns that LIFEWATCH manages to identify.

Throughout the analysis of all four datasets, it becomes evident that hybrid strategies that incorporate deep learning techniques consistently outshine their counterparts in terms of F1 score. Notably, these hybrid methods register more true positives while simultaneously keeping the false alarm rate at its minimum, leading to superior accuracy. The versatility of these algorithms is underscored by their adeptness at identifying subtle changes across a diverse range of datasets. Although they tend to have a slightly higher mean absolute delay than other methods, this compromise is often acceptable for specific applications.

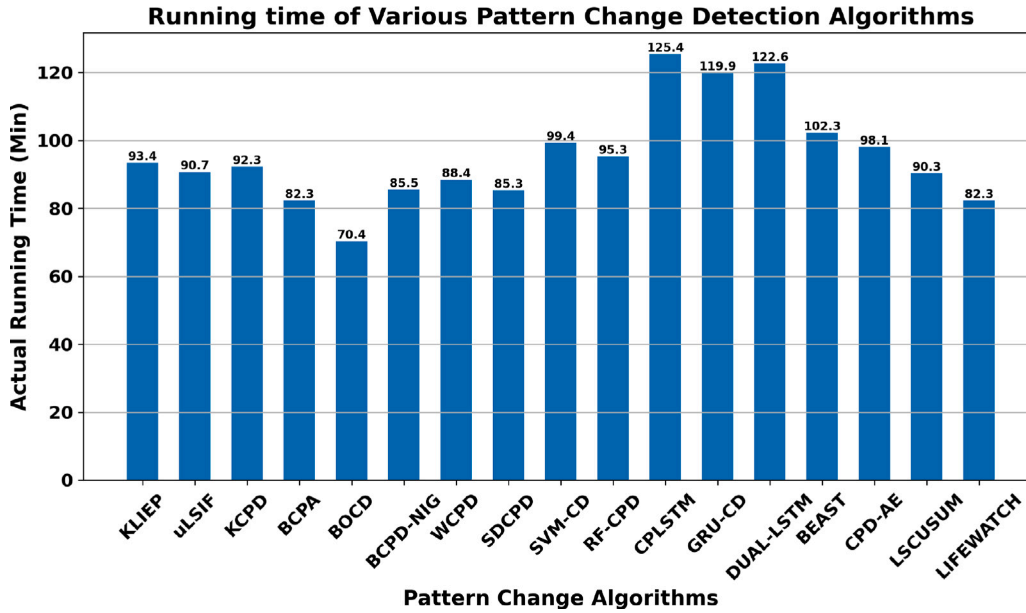

Fig. 10. Actual Running Time of PCD algorithms for HASC2010corpus.

Notably, with CUSUM standing out, parametric statistical methods excel in detecting changes in data characterized by varying mean and variation. This is particularly relevant for data that predominantly exhibit shifts in mean values rather than changes in linear trends. CUSUM's enhanced sensitivity to mean shifts, along with its strong resistance to noise, distinguishes it from others. Its effectiveness is derived from its approach of cumulatively measuring changes over time. This contrasts with methods like PLR, which respond more directly to individual data points.

Moving towards non-parametric statistical algorithms, it is evident that uLSIF outperforms KLIEP. The superior performance of uLSIF can be attributed to its flexible estimation technique, which is well-equipped to adapt to variations in probability distribution, such as mean shifts. The least-squares fitting methodology enables uLSIF to acquire probability distribution more precisely. This acute discernment is particularly beneficial when the goal is to discern shifts in the dataset's mean.

Interestingly, time–frequency methodologies have showcased impressive results for datasets exhibiting changes in auto-correlation. Their ability to discern patterns and shifts in datasets with inherent intricacies highlights their potential utility in specific applications.

In summary, selecting a Point of Change Detection (PCD) algorithm depends not only on the unique challenges and complexities of the application but also on the operational context, whether it is for real-time monitoring or batch analysis. The essential aspect is to find an equilibrium between sensitivity, accuracy, and practicality in the operational environment.

4.5 Complexity analysis of pattern change detection algorithms

The time complexity of each algorithm is assessed based on its operational mechanics. Starting with PLR, the complexity is determined by the method used to identify the CP in time series data. Table 11 shows that PLR has a nonlinear temporal complexity due to Bellman's Dynamic programming technique for any values of N in the dataset.

CUSUM finds noteworthy changes by computing the cumulative sum of deviations from a reference value or mean. While each data point is processed only once, the algorithm's temporal complexity is proportional to the number of data points (as shown in Table 11). CUSUM's computational cost can increase if post-processing, such as a sliding window or thresholding, is required to identify the CP. Depending on the specific post-processing method chosen, the total time complexity may increase in such situations.

KLIEP and uLSIF are non-parametric methods for calculating density ratios between two probability distributions with similar methodologies but differing time complexities. KLIEP has a higher complexity partly due to the necessity of inverting an $(M \times M)$ matrix during optimization, adding an M^3 term to its complexity. uLSIF, not requiring this matrix inversion, generally has a lower time complexity. The complexity of the KCPD algorithm is mainly dependent on the rank approximation of the kernel matrix, often achieved through Nystrom approximation or random subspace methods. Additionally, the efficiency of the chosen eigensolver algorithm significantly influences KCPD's computational burden, affecting its suitability for real-time or large-scale applications.

Bayesian methods like BCPA, which require the entire dataset before the algorithm begins, have a complexity as shown in Table 11, where C is the total number of CP and M is the cost of assessing the likelihood function for a specific segment. For BOCD, where data points are processed as they come, the complexity can be $\mathcal{O}(N \times K)$ per data point, assuming the likelihood calculation has constant complexity. It is important to note that various optimizations and approximations can be applied to BCPA to reduce the time complexity, such as utilizing dynamic programming or efficient approximations to compute the likelihoods (Lu et al., 2023).

For time–frequency methods used in detecting pattern changes in sequential data, such as WCPD, the time complexity depends on the specific implementation and the wavelet transform used, as shown in Table 11. The PCD algorithm applied to the wavelet coefficients also contributes to the overall complexity of the WCPD technique (Chen et al., 2022). Different PCD algorithms have varying complexities, and the total complexity of the WCPD method is a combination of the wavelet transform and the chosen PCD algorithm.

The time complexity of the RF algorithm is primarily influenced by the number of trees in the forest often denoted as and the depth of these trees. Since RF works by evaluating multiple bootstrapped samples (trees) for potential change points, the overall complexity can be significant, especially with many trees. Similarly, in SVM for PCD, the time complexity is influenced by the number of support vectors and the type of kernel used.

In machine learning, methods like LSTM and GRU have complexity affected by the number of layers influenced by forward and backward passes. More layers increase complexity. Factors like batch size,

Table 11
Complexity summary of PCD algorithms.

S. No.	Method	Type	Complexity	Remark
1	PLR-CPD	Parametric	$\mathcal{O}(N^2 \times C)$	N: # Data-points, C: #CP
2	CUSUM-CD	Parametric	$\mathcal{O}(N) + PP$	PP: Post Processing Cost
3	GLRT-CD	Parametric	$\mathcal{O}(N)$	–
4	KLIEP	Non-parametric	$\mathcal{O}(N \times B + B^3)$	B: Number of Basis function(fn)
5	uLSIF	Non-parametric	$\mathcal{O}(N \times B)$	uLSIF < KLIEP
6	KCPD	Non-parametric	$\mathcal{O}(k \times F \times N^3)$	k: # iterations, F: #Features
7	BCPA	Offline Bayesian	$\mathcal{O}(N \times C \times M)$	M: Cost of Likelihood fn
8	BOCD	Online Bayesian	$\mathcal{O}(N)$	–
9	BCPD-NIG	Online Bayesian	$\mathcal{O}(N^2)$	–
10	WCPD	TF Methods	$\mathcal{O}(N \times log(N))$	–
11	SDCPD	TF Methods	$\mathcal{O}(N \times W^2)$	W: Window Length
12	SVM-CD	Supervised	$\mathcal{O}(N^2 \times F) - \mathcal{O}(N^3 \times F)$	–
13	RF-CPD	Supervised	$\mathcal{O}(T \times N \times F \times log(N))$	# trees
14	CPLSTM	Unsupervised	$\mathcal{O}(S \times N \times H^2)$	S: Sequence length H: #Hidden units
15	GRU-CD	Unsupervised	$\mathcal{O}(S \times N \times H^2)$	GRU<LSTM
16	DTW-CD	Unsupervised	$\mathcal{O}(N \times K)$	K: Length of time series
17	Dual-LSTM	Unsupervised	$\mathcal{O}(2 \times S \times N \times H^2)$	–
18	BEAST	Unsupervised	$\mathcal{O}(N^2)$	–
19	CPD-MI	Unsupervised	$\mathcal{O}(N^2 log(N))$	–
20	CPD-AE	Unsupervised	$\mathcal{O}(D^2)$	D: Component of SVD
21	LSCUSUM	Unsupervised	$\mathcal{O}(N \times W)$	–
22	LIFEWATCH	Unsupervised	$\mathcal{O}(N^2)$	–

optimization algorithms, and epoch numbers also impact time complexity (Lattari et al., 2022). For PCD, LSTM suits small to medium-sized datasets and sequences but struggles with large datasets, long sequences, or deep architectures due to high computational demands. Optimization techniques such as parallelization, pruning, or using more efficient architectures like GRU can help. When using DTW for PCD, complexity depends on the frequency of change point checks and the window size used for comparison.

The complexity of hybrid methods, like CPD-MI, is primarily influenced by the estimation technique used. In PCD, calculating mutual information across different data windows is essential for identifying change points. In LIFEWATCH, the complexity is contingent upon the selected number of data points and the window size. Bayesian methods such as BEAST exhibit time complexities ranging from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$ or more, based on the specific implementation. Due to matrix multiplication, the CPD-AE algorithm has a complexity of $\mathcal{O}(N^3)$, with N representing the number of data points. Meanwhile, the testing complexity of the PCD algorithm is $\mathcal{O}(D^2)$, where D denotes the recursive SSA component size in SVD. For LSCUSUM, the complexity varies based on the approach to segmenting the time series into locally stationary sections, typically using a sliding window method or wavelet decomposition.

5 Conclusion and future work

This review article thoroughly examines diverse pattern change algorithms in sequential data, encompassing simple and complex types. Moreover, the paper delves into an analysis of cutting-edge techniques used to detect these pattern changes, referencing existing literature. The survey emphasizes the challenges that algorithms encounter in identifying various pattern changes and assesses the advantages and limitations of these algorithms. The survey describes numerous algorithms that utilize both supervised and unsupervised approaches. Additionally, it examines the algorithms' functionalities in both online and offline modes.

After a comprehensive review of different algorithms such as Statistical Parametric, Non-parametric, Bayesian, Time–Frequency Methods, Machine Learning, and Hybrid Methods, it becomes clear that no single approach fits all scenarios for identifying pattern changes in sequential data. The effectiveness of a particular strategy is significantly influenced by the attributes of the dataset being analyzed. However, it has been observed that hybrid methods often provide superior outcomes in terms of performance across a variety of datasets. These approaches can adapt to different data features and identify complicated changes efficiently. While hybrid approaches have a higher computational cost, they provide a suitable combination of accuracy and versatility, making them useful for diverse applications. It is vital to highlight that the selection of the suitable PCD algorithm should be driven by the unique needs of the situation at hand as well as the characteristics of the data. While determining the best approach for a given job, domain expertise, dataset size, computing resources, and the required degree of accuracy should all be considered.

The survey raised several scientific questions, which have been effectively resolved through performance evaluation. Regarding performance and accuracy, the evaluations revealed that the nature of the dataset plays a crucial role in determining an algorithm's accuracy. While some algorithms are adept at spotting sudden changes, others are more effective in recognizing gradual variations. Hybrid and machine learning-based methods generally show higher accuracy across various datasets.

When considering complexity and efficiency, it is clear that statistical methods like CUSUM-CD or PLR-CPD, though computationally efficient, might not be complex enough for detailed datasets. Conversely, Bayesian algorithms offer a good balance, managing complexity with moderate computational needs.

For robustness and reliability, statistical methods and machine learning, particularly ensemble approaches, demonstrate enhanced resilience to noise and incomplete data. In terms of interpretability and transparency, machine learning models such as random trees provide valuable insights into their decision-making processes.

Scalability depends mainly on the specific field, but algorithms designed for parallel processing or those utilizing efficient data structures generally handle large-scale data more effectively. Real-time or online algorithms also show notable Scalability.

Regarding adaptability, online algorithms with real-time parameter updates excel in adjusting to evolving data trends. Models like reinforcement learning or those using a sliding window technique also display significant adaptability.

In intervention and actionability, algorithms that produce probabilistic forecasts or are integrated with decision-making frameworks, such as reinforcement learning, are helpful for actionable steps. Moreover, models that offer confidence intervals or prediction bands, like some Bayesian models, provide valuable, actionable information.

Advancements in pattern change detection still face challenges such as detecting multiple change points, managing missing or high-dimensional data, and processing large datasets efficiently. Ongoing research is vital for improving the performance of PCD algorithms for applications like health monitoring and speech and image analysis. Future efforts should develop algorithms capable of addressing large-scale data and integrating PCD with concept drift. Hybrid methods and optimized models could improve accuracy and reduce the occurrence of false positives and negatives. Additionally, refining the algorithms to better handle data with variable frequency and auto-correlation is needed. Reducing the initialization data for models could speed up response times. Improving the computational efficiency of PCD algorithms is crucial, as it would make adaptive pre-processing more viable in real-world applications. These improvements will make PCD algorithms more applicable across diverse fields, from environmental monitoring to medical surveillance.

CRediT authorship contribution statement

Muktesh Gupta: Investigation, Methodology, Validation, Writing – original draft. **Rajesh Wadhvani:** Conceptualization, Review, Editing, Supervision, Validation, Writing – review & editing. **Akhtar Rasool:** Conceptualization, Review, Supervision, Validation, Visualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The Alan Institute deserves our thanks for making their Turing CP work accessible to the public. The dataset in their collection was obtained expressly to evaluate the performance of PCD algorithms.

References

Altamirano, M., Briol, F.-X., & Knoblauch, J. (2023). Robust and scalable Bayesian online changepoint detection. arXiv preprint arXiv:2302.04759.

Aminikhanghahi, S., & Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, *51*(2), 339–367.

Aminikhanghahi, S., Wang, T., & Cook, D. J. (2018). Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, *31*(5), 1010–1023.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. (2013). A public domain dataset for human activity recognition using smartphones. In *Esann*: *vol. 3*, (p. 3).

Banesh, D., Petersen, M., Wendelberger, J., Ahrens, J., & Hamann, B. (2019). Comparison of piecewise linear change point detection with traditional analytical methods for ocean and climate data. *Environmental Earth Sciences*, *78*, 1–16.

Barbado, A., Corcho, Ó., & Benjamins, R. (2022). Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM. *Expert Systems with Applications*, *189*, Article 116100.

Bozdal, M., Samie, M., & Jennions, I. K. (2021). WINDS: A wavelet-based intrusion detection system for controller area network (CAN). *IEEE Access*, *9*, 58621–58633. http://dx.doi.org/10.1109/ACCESS.2021.3073057.

Van den Burg, G. J. J., & Williams, C. K. I. (2020). An evaluation of change point detection algorithms. arXiv preprint arXiv:2003.06222.

Chang, W.-C., Li, C.-L., Yang, Y., & Póczos, B. (2019). Kernel change-point detection with auxiliary deep generative models. arXiv preprint arXiv:1901.06077.

Chatterjee, S., & Qiu, P. (2009). Distribution-free cumulative sum control charts using bootstrap-based control limits.

Chen, J., Zhou, S., Qiu, Y., & Xu, B. (2022). An anomaly detection method of time series data for cyber-physical integrated energy system based on time-frequency feature prediction. *Energies*, *15*(15), 5565.

Cheng, K. C., Aeron, S., Hughes, M. C., Hussey, E., & Miller, E. L. (2020). Optimal transport based change point detection and time series segment clustering. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 6034–6038). IEEE.

Corradin, R., Danese, L., & Ongaro, A. (2022). Bayesian nonparametric change point detection for multivariate time series with missing observations. *International Journal of Approximate Reasoning*, *143*, 26–43.

De Oca, V. M., Jeske, D. R., Zhang, Q., Rendon, C., & Marvasti, M. (2010). A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. *Journal of Systems and Software*, *83*(7), 1288–1297.

Deldari, S., Smith, D. V., Xue, H., & Salim, F. D. (2021). Time series change point detection with self-supervised contrastive predictive coding. In *Proceedings of the web conference 2021* (pp. 3124–3135).

El-Sappagh, S., Abuhmed, T., Islam, S. R., & Kwak, K. S. (2020). Multimodal multitask deep learning model for Alzheimer's disease progression detection based on time series data. *Neurocomputing*, *412*, 197–215.

Faber, K., Corizzo, R., Sniezynski, B., Baron, M., & Japkowicz, N. (2022). Lifewatch: Lifelong wasserstein change point detection. In *2022 international joint conference on neural networks* (pp. 1–8). IEEE.

Fan, J., Wang, Z., Wu, H., Sun, D., Wu, J., & Lu, X. (2023). An adversarial time–frequency reconstruction network for unsupervised anomaly detection. *Neural Networks*, *168*, 44–56.

Guo, S., Guo, W., Abolhassani, A., Kalamdani, R., Puchala, S., Januszczak, A., et al. (2019). Manufacturing process monitoring with nonparametric change-point detection in automotive industry. *Journal of Manufacturing Science and Engineering*, *141*(7), Article 071013.

Gupta, M., Wadhvani, R., & Rasool, A. (2022). Real-time change-point detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data. *Expert Systems with Applications*, *209*, Article 118260.

Gupta, M., Wadhvani, R., & Rasool, A. (2023). A real-time adaptive model for bearing fault classification and remaining useful life estimation using deep neural network. *Knowledge-Based Systems*, *259*, Article 110070.

Hushchyn, M., & Ustyuzhanin, A. (2021). Generalization of change-point detection in time series data based on direct density ratio estimation. *Journal of Computer Science*, *53*, Article 101385.

Iqbal, W., Berral, J. L., Carrera, D., et al. (2020). Adaptive sliding windows for improved estimation of data center resource utilization. *Future Generation Computer Systems*, *104*, 212–224.

Jiang, J., Zheng, Y., Pang, T., Wang, B., Chachan, R., & Tian, Y. (2020). A comprehensive study on spectral analysis and anomaly detection of river water quality dynamics with high time resolution measurements. *Journal of Hydrology*, *589*, Article 125175.

Jin, B., Chen, Y., Li, D., Poolla, K., & Sangiovanni-Vincentelli, A. (2019). A one-class support vector machine calibration method for time series change point detection. In *2019 IEEE international conference on prognostics and health management* (pp. 1–5). IEEE.

Jin, Y.-F., Yin, Z.-Y., Zhou, W.-H., & Horpibulsuk, S. (2019). Identifying parameters of advanced soil models using an enhanced transitional Markov chain Monte Carlo method. *Acta Geotechnica*, *14*, 1925–1947.

Khusna, H., Mashuri, M., Ahsan, M., Suhartono, S., & Prastyo, D. D. (2020). Bootstrap-based maximum multivariate CUSUM control chart. *Quality Technology & Quantitative Management*, *17*(1), 52–74.

Kloska, M., Grmanova, G., & Rozinajova, V. (2023). Expert enhanced dynamic time warping based anomaly detection. *Expert Systems with Applications*, *225*, Article 120030.

Kopp, M., Pevnỳ, T., & Holeňa, M. (2020). Anomaly explanation with random forests. *Expert Systems with Applications*, *149*, Article 113187.

Kurt, M. N., Yılmaz, Y., & Wang, X. (2020). Real-time nonparametric anomaly detection in high-dimensional settings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(7), 2463–2479.

Lang, C. I., Sun, F.-K., Lawler, B., Dillon, J., Al Dujaili, A., Ruth, J., et al. (2022). One class process anomaly detection using kernel density estimation methods. *IEEE Transactions on Semiconductor Manufacturing*, *35*(3), 457–469.

Lattari, F., Rucci, A., & Matteucci, M. (2022). A deep learning approach for change points detection in InSAR time series. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–16. http://dx.doi.org/10.1109/TGRS.2022.3155969.

Lee, S., Lee, S., & Moon, M. (2020). Hybrid change point detection for time series via support vector regression and CUSUM method. *Applied Soft Computing*, *89*, Article 106101.

Li, Y., Lin, G., Lau, T., & Zeng, R. (2019). A review of changepoint detection models. arXiv preprint arXiv:1908.07136.

Li, L., Liu, G., Zhang, L., & Li, Q. (2019). Sensor fault detection with generalized likelihood ratio and correlation coefficient for bridge SHM. *Journal of Sound and Vibration*, *442*, 445–458.

Li, E., Zeng, L., Zhou, Z., & Chen, X. (2019). Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communication*, *19*(1), 447–457.

Liu, S., Wright, A., & Hauskrecht, M. (2018). Change-point detection method for clinical decision support system rule monitoring. *Artificial Intelligence in Medicine*, *91*, 49–56.

Liu, S., Yamada, M., Collier, N., & Sugiyama, M. (2013). Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, *43*, 72–83.

Londschien, M., Bühlmann, P., & Kovács, S. (2023). Random forests for change point detection. *Journal of Machine Learning Research*, *24*(216), 1–45.

Lu, J., Wang, C., Zhang, J., & Wang, X. (2023). A sequential Bayesian changepoint detection procedure for aberrant behaviours in computerized testing. *British Journal of Mathematical and Statistical Psychology*.

Muggeo, V. M., et al. (2008). Segmented: An R package to fit regression models with broken-line relationships. *R News*, *8*(1), 20–25.

Pourhabibi, T., Ong, K.-L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, *133*, Article 113303.

Pushkar, A., Gupta, M., Wadhvani, R., & Gyanchandani, M. (2022). A comparative study on change-point detection methods in time series data. In *2022 2nd international conference on intelligent technologies* (pp. 1–7). IEEE.

Rezvani, R., Barnaghi, P., & Enshaeifar, S. (2021). A new pattern representation method for time-series data. *IEEE Transactions on Knowledge and Data Engineering*, *33*(7), 2818–2832. http://dx.doi.org/10.1109/TKDE.2019.2961097.

Sgueglia, A., Di Sorbo, A., Visaggio, C. A., & Canfora, G. (2022). A systematic literature review of iot time series anomaly detection solutions. *Future Generation Computer Systems*.

Shafi, I., Aziz, A., Din, S., & Ashraf, I. (2022). Reduced features set neural network approach based on high-resolution time-frequency images for cardiac abnormality detection. *Computers in Biology and Medicine*, *145*, Article 105425.

Shi, Z., & Chehade, A. (2021). A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, *205*, Article 107257.

Su, X., Deledalle, C.-A., Tupin, F., & Sun, H. (2014). Change detection and classification of multi-temporal SAR series based on generalized likelihood ratio comparing-and-recognizing. In *2014 IEEE geoscience and remote sensing symposium* (pp. 1433–1436). http://dx.doi.org/10.1109/IGARSS.2014.6946705.

Sundararajan, R. R., & Pourahmadi, M. (2018). Nonparametric change point detection in multivariate piecewise stationary time series. *Journal of Nonparametric Statistics*, *30*(4), 926–956.

Tang, C., Xu, L., Yang, B., Tang, Y., & Zhao, D. (2023). GRU-based interpretable multivariate time series anomaly detection in industrial control system. *Computers & Security*, *127*, Article 103094.

Thies, S., & Molnár, P. (2018). Bayesian change point analysis of bitcoin returns. *Finance Research Letters*, *27*, 223–227.

Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, *167*, Article 107299.

Verzelen, N., Fromont, M., Lerasle, M., & Reynaud-Bouret, P. (2023). Optimal change-point detection and localization. *The Annals of Statistics*, *51*(4), 1586–1610.

Wang, X., Borsoi, R. A., Richard, C., & Chen, J. (2023). Change point detection with neural online density-ratio estimator. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Warwicker, J. A., & Rebennack, S. (2023). Generating optimal robust continuous piecewise linear regression with outliers through combinatorial benders decomposition. *IISE Transactions*, *55*(8), 755–767.

Wen, T., & Keyes, R. (2019). Time series anomaly detection using convolutional neural networks and transfer learning. arXiv preprint arXiv:1905.13628.

Wen, X., Lu, G., Liu, J., & Yan, P. (2020). Graph modeling of singular values for early fault detection and diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, *145*, Article 106956.

Yamada, M., Kimura, A., Naya, F., & Sawada, H. (2013). Change-point detection with feature selection in high-dimensional time-series data. In *Twenty-third international joint conference on artificial intelligence*.

Zhang, C., Zhou, T., Wen, Q., & Sun, L. (2022). TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 2497–2507).

Zhao, K., Wulder, M. A., Hu, T., Bright, R., Wu, Q., Qin, H., et al. (2019). Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm. *Remote Sensing of Environment*, *232*, Article 111181.

Zhou, R., Gueaieb, W., & Spinello, D. (2023). A model-free Kullback–Leibler divergence filter for anomaly detection in noisy data series. *Journal of Dynamic Systems, Measurement, and Control*, *145*(2), Article 024501.