REVIEW ARTICLE

Ruiyu XU, Zheren SONG, Jianguo WU, Chao WANG, Shiyu ZHOU

Change-point detection with deep learning: A review

Abstract   Recent advances in deep learning have led to the creation of various methods for change-point detection (CPD). These methods enhance the ability of CPD techniques to handle complex, high-dimensional data, making them more adaptable and less dependent on strict assumptions about data distributions. CPD methods have also demonstrated high accuracy and have been applied across various fields, including manufacturing, healthcare, activity monitoring, finance, and environmental monitoring. This review provides an overview of how these methods are applied, the data sets they use, and how their performance is evaluated. It also organizes techniques into supervised and unsupervised categories, citing key studies. Finally, we explore ongoing challenges and suggest directions for future research to improve interpretability, generalizability, and real-world implementation.

Keywords   change-point detection, deep learning, supervised learning, unsupervised learning, time-series analysis

1   Introduction

Change-point detection (CPD) has become a key technique for identifying shifts in data sequences, which provides valuable insights and enables timely interventions. CPD methods have been extensively studied across various fields, including industrial manufacturing (Chen et al., 2019), healthcare (Khan et al., 2017), human activity monitoring (Khan et al., 2016), financial data analysis (Habibi 2022), and environmental monitoring (Jaiswal et al., 2015). The concept of CPD dates back to the 1950s (Page 1954), when it was originally used to detect changes in the mean values of independent and identically distributed (IID) Gaussian variables for quality control in the industry. Over the years, many statistical model-based methods for CPD have been developed to address the needs of various applications.

Most of the above statistical methods require some form of distributive pattern in the data and estimate the efficiency of a detected change through the theoretical asymptotic presentation. Among the key statistical methods, the Bayesian change-point model embeds a priori knowledge of changes within a framework for probabilistic models (Chopin 2007; Wu et al., 2016; Wen et al., 2018; Wen et al., 2019). Other related methods for the tasks of CPD are clustering (Keogh et al., 2001), subspace models (Kawahara et al., 2007; Xu et al., 2023c), Gaussian process (Saatçi et al., 2010), CUSUM principle (Lee et al., 2020), Graph-based models (Chen and Chu 2023), distance-based methods (Matteson and James 2014), and density ratio models (Aminikhanghahi et al., 2019). For details of the theory on which all classical CPD is based, see Basseville and Nikiforov (1993).

With increased complexity and volume of data, however, the development of robust and scalable methods that could effectively detect change points has increased. Despite their theoretical robustness, applications to high-dimensional time series data have mostly demonstrated limitations of classical methods. These usually manifest as model complexity and generalization challenges across diverse, data-intensive environments (Liu et al., 2022a). But gradually, it was realized that the traditional statistical methods were insufficient in model adaptability and computational efficiency regarding certain practical applications.

Recently, deep learning technologies have been of ever-growing interest, and they have shown outstanding results in several time-series problems, such as forecasting. Deep learning models designed especially for the

Ruiyu XU, Zheren SONG, Jianguo WU (✉)
Department of Industrial Engineering and Management, Peking University, Beijing 100871, China
E-mail: j.wu@pku.edu.cn

Chao WANG
Department of Industrial and Systems Engineering, University of Iowa, Iowa City, IA 52242, USA

Shiyu ZHOU
Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

processing of sequential data, including but not limited to Recurrent Neural Networks (RNNs) and Autoencoder models, are being increasingly applied to overcome certain difficulties presented in high-dimensional data sets (Gupta et al., 2022). Compared with traditional methods, although these deep learning models do not have such a strong theoretical basis, they demonstrate great capability in the modeling of complex systems. These models are much better at handling complex change-point detection tasks, such as video monitoring. Deep learning models have great capability to deeply excavate data features and transmit temporal information within time series data. By doing so, this also provides them with an extended capability of finding out patterns and dependencies that may not be considered in traditional statistical methods. Thus, this model further develops the detecting ability of significant shifts that are difficult for statistical methods to pick up with ease, thus giving higher accuracy in practical applications. Deep learning methods can also take advantage of the learning of representation from vast amounts of data and allow models to adapt and perform with a high degree of accuracy across diverse scenarios without explicit model assumptions. Moreover, rapid development in hardware technology, such as parallelization in GPUs and CPUs, has also ensured that the computational requirements for deep learning methods remain within feasible limits for practical applications.

The surprising fact is that despite this growing trend in the adoption of deep learning for CPD, very few comprehensive literature reviews have focused on these modern methods. Previous surveys (Niu et al., 2016; Aminikhanghahi and Cook 2017; Truong et al., 2020) have tended to focus on more classical methods, paying minimal attention to deep learning advancement.

Our review aims to fill this gap by providing a systematic exploration of deep learning applications in change-point detection. First, we will discuss how these methods handle different types of data, such as high-dimensional time series and video data, and summarize some key pre- and post-processing techniques for them that ensure effective model implementation. Besides, deep learning-based CPD methods have been further divided into supervised and unsupervised frameworks, since the feasibility of different training approaches has to be taken into consideration based on data labeling. We will provide detailed expositions on the recent research advances of each category with respect to specific network architectures that each has utilized. This review also discusses challenges such as how to deal with strategies that handle few, unlabelled, and multimodality data, how to request real-time online detection algorithms, and how to provide strongly required model interpretability. We propose correspondingly some potential directions that could address these issues in the future. This review, therefore, summarizes in detail deep learning methods to CPD for the acquisition of better insight into its understanding and encouraging further research and development in this field.

The paper is structured as follows. Section 2 covers background information, including definitions, formulations, applications, and data sets. Section 3 discusses deep learning methods and related processing strategies for CPD tasks. Section 4 highlights the limitations of current methods and explores potential future research directions. Finally, Section 5 offers a brief conclusion.

2 Background of CPD

In this section, we present an overview of the foundational definitions, problem formulation, and the wide range of applications and data sets relevant to CPD methods based on deep learning.

2.1 Definitions

Our discussion begins by establishing the essential definitions of key terms and formulating the problem of CPD. A time series data are defined as a sequence comprising T elements $S = \{x_1, \ldots, x_i, \ldots, x_T\}$. Here, the element x_i can vary in its form depending on the context, such as a d-dimensional vector in the case of high-dimensional time series, or an image (matrix) when dealing with video data. We assume that the data sampling interval is even, and therefore, our review does not cover irregularly sampled data, such as general longitudinal data.

In scenarios where data are collected from multiple sources, we deal with M time series, represented as $\mathcal{S} = \{S^{(1)}, \ldots, S^{(j)}, \ldots, S^{(M)}\}$. Each series $S^{(j)} = \{x_1^{(j)}, \ldots, x_i^{(j)}, \ldots, x_T^{(j)}\}$, may consist of elements in various forms, which can complicate the analysis. However, these data series are typically collected simultaneously, ensuring that they are aligned in time. It should be noted that this review does not cover cases where data from multiple sources is not aligned. For more information on handling misaligned data from multiple sources, readers are encouraged to consult additional literature (Olsen et al., 2018; Luo and Hu 2024).

In the data collection process, the operating status of the observation system may change, resulting in one or several change points. A change point signifies a transition between different states within the time series data. Suppose there are C change-points at $\tau_1, \ldots, \tau_C$ satisfying $1 < \tau_1 < \cdots < \tau_C < T$, then the time series is divided into $C + 1$ segments. If $C > 1$, the CPD task is referred to as a multiple CPD problem. Otherwise, it is considered a single CPD problem. Within each segment, the observations $S_{\tau_i+1:\tau_{i+1}}$ exhibit the same data pattern, while abrupt transitions indicate the emergence of change-point. These transitions can take different forms, such as shifts in data distributions (Chen 2019), alterations in data correlations (Cabrieto et al., 2017), changes in graph structures

(Sulem et al., 2024), and variations in temporal characteristics (Wu and Zhou, 2024). Figure 1 displays examples of these different types of change points.
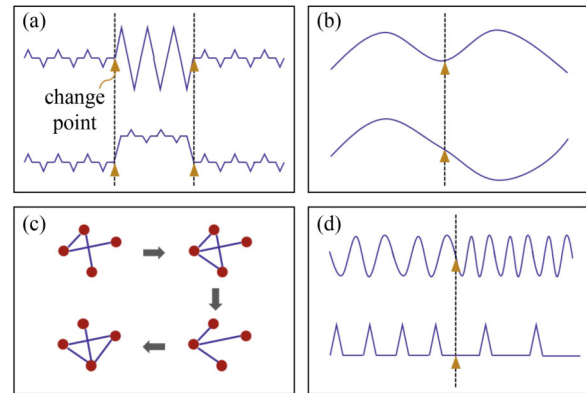
CPD methods are designed to address these types of problems by determining whether a time series contains a change point, counting the number of change points, and identifying their positions. In certain applications, CPD methods are also expected to classify each segment created by the identified change points using specific labels. This introduces an overlap between CPD tasks and labeling-based segmentation tasks, which classify each data point individually and then perform time series segmentation based on those labels. Additionally, the CPD tasks include a particular subset of anomaly detection tasks, where anomalies occur collectively and form abnormal intervals. In such cases, the last normal data point can be considered a change point, transforming the anomaly detection tasks into tasks of detecting a single change point. Therefore, this review compiles information on CPD, labeling-based segmentation, and interval-based anomaly detection methods to offer a comprehensive perspective. Figure 2 illustrates the relationships and differences among these tasks.

Generally, most CPD algorithms can be categorized into two phases: Phase I training and Phase II detection. In Phase I, historical data are used to train the model parameters and establish a baseline understanding of data patterns. In Phase II, the model is applied to new data to detect deviations or shifts that indicate potential change points. It is important to note that in some dynamic and adaptive methods, Phase I is not a discrete step, but rather an ongoing process. In such methods, the model continuously updates its understanding and adapts to new data patterns as data arrives. Therefore, the boundaries between Phase I and Phase II are intertwined and not distinctly separate.
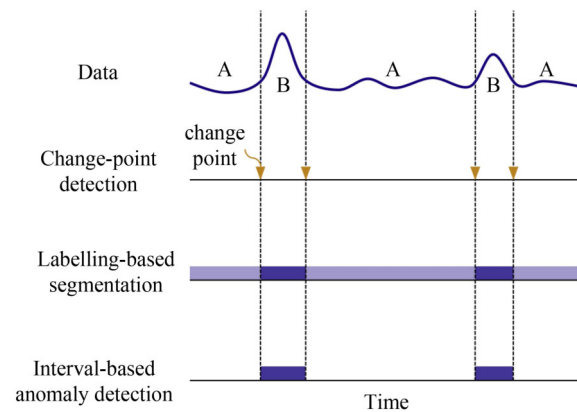
2.2   Online vs offline

CPD methods can be broadly classified as online or offline techniques. Online change detection involves real-time monitoring and detection of changes as they occur. It requires continuously analyzing incoming data streams or signals to detect immediate or recent changes. The detection process is ongoing and often sequential. On the other hand, offline change detection assumes that a sequence is available and aims to identify whether any change point(s) occurred in the time series.

For online methods, the detection delay is a key metric to evaluate the timeliness of detection. However, most literature lacks corresponding discussions and rarely compares detection delays. In fact, most neural network architectures are inherently capable of performing online detection but with some latency. For models that use time windows, the choice of window size needs careful consideration. A large window size may result in high

Fig. 1   Four representative types of change-point: (a) shifts in data distributions, (b) alterations in data correlations, (c) changes in graph structures, and (d) variations in temporal characteristics.

Fig. 2   The illustration of the change-point detection task, the labeling-based segmentation task, and the interval-based anomaly detection task.

detection delays, while a small window size could lead to a high false alarm rate.

Offline methods are not constrained by the immediacy of detection and can afford the computational time to apply more complex analysis techniques to the entire data set. This allows for a more thorough examination of the data. For example, sequence-to-sequence methods like Autoencoders and Encoder-Decoder, if designed to process the entire sequence at once, fall under offline methods. Additionally, any model that incorporates post-processing steps is inherently an offline method as these steps typically require access to the complete data set before analysis can begin.

2.3   Applications and data sets

In this subsection, we explore the wide range of applications for CPD methods, highlighting their significance across various domains. Furthermore, we enumerate the data sets frequently cited in academic literature within these areas for the convenience of researchers. Detailed

information about these data sets is summarized in Tables 1 to 3.

2.3.1 High-dimensional time series

Health care. In the healthcare sector, the analysis of patient monitoring data are of utmost importance. CPD methods play a crucial role in enabling early detection of acute medical events such as seizure, arrhythmia, and sleep apnea. By tracking changes in physiologic signals like electroencephalography (EEG) and electrocardiograph (ECG) signals, CPD methods can promptly alert medical professionals to intervene. Commonly used data sets in

this domain include the CHB-MIT data set (Shoeb, 2009; Guttag, 2010), the Bonn data set (Andrzejak et al., 2001) for seizure detection, the MIT-BIH data set (Moody and Mark, 2001) for arrhythmia detection, and the Apnea-ECG data set(Penzel et al., 2000) for sleep apnea detection. For a detailed overview of this area, please refer to the works of Shoeibi et al., (2021), Xiao et al., (2023) and Ramachandran and Karuppiah (2021).

Stock trading. In the financial sector, CPD methods have become essential for monitoring financial crashes and significant global events that impact the markets. These methods analyze complex financial time series data to identify abrupt changes or anomalies before

Table 1 The commonly used high-dimensional time-series data sets for change-point detection

	Dataset	#classes	#CPs	#dimensions	#sequence	Sampling frequency	Length
Health care	CHB-MIT	2	–	23	23	256 Hz	hours
	Bonn	2	–	1	500	173.61 Hz	23.6 s
	MIT-BIH	2	–	2	47	360 Hz	~30 min
	Apnea-ECG	2	–	1	70	100 Hz	7–8 hours
Stock trading	S&P 500 index	–	9	685	1	1 month	20 years
	Apple stock	–	8	2	1	3 days	9 years
Industrial manufacturing	Amazon-CPU	2	16	1	10	5 min	14 days
	Well log	–	9	1	1	–	4050
	Hydraulic Pump	44	–	9	120	100 Hz	4–6 min
Speaker diarization	CALLHOME	2–7	–	1	500	–	~10 min
Sleep staging	Sleep-EDF Expanded	8	–	4	197	100 Hz	~20 hours
	MASS	5	–	8–24	200	256 Hz	7~8 hours
Climate monitoring	water level	2	–	1	27	10 min	8 years
	CO2 Emission	–	7	1	1	–	214
	temperature	–	4	1	1	–	1980
Others	Bee dance	3		3	6		~1000
	Opportunity	18		113	53720	30 Hz	
	HASC	6	65	3	1		39000

Table 2 The commonly used video data sets for change-point detection

	Data set	#classes	#CPs	#sequence	frame rate	Length
Action segmentation	Breakfast	48	~11 K	1712	15 fps	77 h
	50Salads	17	~0.9 K	50	30 fps	5.5 h
	ActivityNet	203	–	19994	30 fps	648 h
	GTEA	71	~0.5 K	28	15 fps	0.4 h

Table 3 The commonly used data sets in multiple data modalities for change-point detection

	Dataset	modalities	#classes	#CPs	#sequence	Length
Action segmentation	EPIC-KITCHENS-55	RGB, Audio	149	39596	432	55 hours
	EPIC-KITCHENS-100	RGB, Audio	4053	89977	700	100 hours
Speaker diarization	Switchboard1	Audio, Text	2		2400	~260 hours
	AMI meeting corpus	RGB, Audio, Text	4–5		171	100 hours

market downturns, enabling investors and analysts to proactively respond to potential risks. Detecting such change-points in financial data sets allows for a deeper understanding of market dynamics and the external factors influencing them. The data sets used typically consist of fundamental stock indices, including daily opening, high, low, and closing prices, adjusted close, and trading volume of these stocks (Au Yeung et al., 2020; Gupta et al., 2022; Sulem et al., 2024).

Industrial manufacturing. CPD methods play a crucial role in predictive maintenance. These methods are capable of detecting anomalies or changes in machinery operation data, which can serve as indicators of impending failures. By employing this approach, the amount of downtime and maintenance costs can be significantly reduced. For instance, the IGT-thermocouple data set (Maleki et al., 2021) can be utilized to detect anomalies at the burner-tip of an Industrial Gas Turbine (IGT), whereas the Amazon-CPU data set (Ahmad et al., 2017) is useful for tracking anomalies in CPU utilization of an Amazon EC2 instance. Additionally, CPD methods can be employed to monitor changes in both the operational status of systems and their external environmental conditions. This aids in the development of platforms for system observation and monitoring, thereby enhancing the perception capabilities of industrial systems. Notable data sets in this regard include the Well log data set (Ruanaidh et al., 1994) which consists of nuclear magnetic resonance measurements while drilling a well and can be used to detect changes in rock stratification, and the Hydraulic Pump End-of-Line Data set (Gaugel and Reichert, 2023), which facilitates the segmentation of operational states in the End-of-Line (EoL) testing process of hydraulic pumps.

Speaker diarization tasks. The objective is to segment audio streams in order to identify and attribute speech segments to individual speakers. This process greatly aids in the transcription and analysis of meetings, calls, and broadcasts. By incorporating CPD methods, changes between speakers can be automatically detected, thereby enhancing the accuracy of speaker diarization processes. The NIST SRE 2000 (LDC2001S97, Disk-8) data set, more commonly known as the CALLHOME data set, is widely utilized as a benchmark in recent literature concerning speaker diarization. For a comprehensive understanding of this area, one can refer to the works of Park et al., (2022) and Bai and Zhang (2021).

Sleep research. CPD methods play a pivotal role in sleep staging. These methods facilitate the segmentation of sleep data into various stages, thereby aiding in the diagnosis of sleep disorders. By analyzing polysomnography data, CPD methods excel at detecting transitions between REM and non-REM sleep stages, leading to a better understanding and treatment of sleep-related issues. Valuable sources of sleep recordings for CPD research in this domain include the Sleep-EDF Expanded data set (Kemp et al., 2000) and the Montreal Archive of

Sleep Studies (MASS) data set (O'reilly et al., 2014). For a detailed overview of this area, one can refer to the work of Imtiaz (2021).

Climate monitoring. The application of CPD in climate data analysis plays a crucial role in identifying significant shifts in environmental conditions, contributing to the study of climate change. By pinpointing points of change in temperature, precipitation, and other climatic variables, researchers can gain valuable insights into long-term climate patterns and anomalies. Various data sets, including the river water level station data set for detecting pluvial floods (Miau and Hung, 2020) the CO_2 Emission data set for monitoring carbon dioxide emissions from the burning of fossil fuels (Gupta et al., 2022), and the temperature data set for detecting anomalies in temperature data, are used in this analysis.

Others fields. CPD methods also provide valuable insights in various other fields, such as animal behavior research and human activity research. In the analysis of the Bee Dance data set, CPD methods are employed to discern complex communication patterns among bees (Oh et al., 2008). Additionally, data sets like the Opportunity data set (Roggen et al., 2010; Chavarriaga et al., 2013) and the Human Activity Sensing Consortium (HASC) data set (Kawaguchi et al., 2011) contain detailed recordings of diverse human actions, allowing for the detection of transitions among different activities like walking, running, and sitting.

2.3.2 Videos

Action segmentation. This task involves the analysis of video streams to segment and classify various actions or activities captured in the footage. CPD methods are utilized to analyze each segment of the video and detect transitions among different actions, enabling automated annotation and indexing of video content. Data sets like the Breakfast Data set (Kuehne et al., 2014), the 50Salads Data set (Stein and McKenna 2013), the ActivityNet Data set (Caba Heilbron et al., 2015), and the Georgia Tech Egocentric Activities (GTEA) Data set (Fathi et al., 2011) provide extensive examples of actions, facilitating the development of algorithms capable of recognizing a wide range of human activities with high precision. For a more detailed overview of this area, refer to the works of Herath et al., (2017), Vahdani and Tian (2023), Ding et al., (2023) and Gammulle et al., (2023).

2.3.3 Multi-modality data

To enhance the accuracy and robustness of CPD methods, there has been significant recent attention on integrating multiple data modalities and transferring knowledge across modalities. For instance, in the field of Human Action Recognition (HAR), visual modalities such as RGB, skeleton, depth, infrared sequences, as well

as non-visual modalities like audio, acceleration, radar, and WiFi, can be combined to analyze human actions. Notable data sets for CPD tasks include the EPIC-KITCHENS data set (Damen et al., 2018; Damen et al., 2022). In speaker diarization tasks, researchers have explored multimodal solutions that incorporate audio and text, and these models have shown improved performance compared to audio-only models. The Switchboard1 data set (Godfrey and Holliman 1997) and the AMI meeting corpus data set (Carletta 2007) are two commonly used data sets in this area.

2.4 Performance evaluation

In deep learning-based CPD methods, the output of CPD algorithms typically falls into two categories. The first type assigns a label to each frame in a time series, detecting change points when consecutive frames have different labels. The second type directly reports the number and positions of change points, treating the segments between these points as data of the same type.

Accordingly, performance evaluation for CPD algorithms can be categorized into three types: frame-based, CP-based, and segment-based evaluation. For frame-based evaluation, standard metrics used in supervised learning algorithms can be employed to assess performance. CP-based metrics evaluate the differences between the actual and estimated change point locations based on the detected change-points. Segment-based metrics consider both the position and length of each segment, evaluating the differences between the actual and estimated segments.

2.4.1 Frame-based metrics

Mean over Frames (*MoF*), also referred to as Accuracy, is defined as the proportion of correctly classified frames out of the total number of frames.

$$MoF = Accuracy = \frac{\# \, of \, correct \, frames}{\# \, of \, all \, frames}.$$

This metric offers a comprehensive frame-wise evaluation and is suitable for supervised scenarios. However, it may not be effective for evaluating performance in data sets with imbalanced classes, especially when there is a significant disparity in the sample sizes of different segments. Additionally, MoF fails to account for segment quality and can yield a high score even when segments are highly fragmented. This issue, known as the over-segmentation problem, arises when a segment is divided into multiple discontinuous sub-segments.

2.4.2 CP-based metrics

Mean squared error (*MSE*). This metric evaluates the precision of predicted change points in relation to the actual change points.

$$MSE = \frac{\sum_{i=1}^{\#CP}(Predicted\,(CP) - Actual\,(CP))^2}{\#CP}.$$

Similar and related metrics include the mean absolute error (MAE) and the root mean square error (RMSE).

$$MAE = \frac{\sum_{i=1}^{\#CP}|Predicted\,(CP) - Actual\,(CP)|}{\#CP},$$

$$RMSE = \sqrt{MSE}.$$

These metrics directly measure the distance between the predicted change points and the actual change points. However, they can be strongly influenced by outliers among the predicted change points, leading to higher values. Additionally, these metrics typically require that the number of predicted change points is predetermined as the ground truth, limiting the robustness of the algorithms.

2.4.3 Segment-based metrics

Edit score (*edit*) is calculated using the normalized edit distance between the ground truth label sequence (G) and the predicted label sequence (P), using the Wagner-Fischer algorithm. This score quantifies the similarity between two sequences by determining the minimum number of insertions, deletions, and replacements needed to transform one segment sequence into the other.

$$Edit\,Score = \left(1 - \frac{Edit\,distance\,(G,P)}{\max\,(|G|,|P|)}\right) \times 100,$$

where the *Edit distance* (*G*,*P*) refers to the edit distance between the two sequences, and |G|,|P| are the lengths of the two sequences respectively. The edit score allows evaluation of how well a model predicts the sequence of segment labels without requiring an exact frame-by-frame match with the ground truth. However, this metric primarily focuses on the accuracy of the order of segment labels and may overlook the accuracy of change point positions.

F1 score, another type of segment-based metric, combines the Intersection over Union (IoU) of each segment with binary classification metrics to deal with this aspect.

$$IoU = \frac{Area\,of\,Overlap}{Area\,of\,Union}.$$

First, a segment is considered a true positive if its *IoU* with the ground truth exceeds a threshold $\eta/100$, as a high *IoU* indicates that the predicted positions of change-points are close to the ground truth. If there are multiple

correct segments within a single ground truth segment, only one segment is considered a true positive, while the others are marked as false positives. Missed segments are marked as false negatives. The Precision, Recall, and *F1* score can then be calculated based on the classification results of the segments. Precision is calculated as the number of true positives (*TP*) divided by the sum of true positives and false positives (*FP*), and Recall is calculated as the number of true positives divided by the sum of true positives and false negatives (*FN*):

$$Precision = \frac{TP}{TP+FP}, \; Recall = \frac{TP}{TP+FN}.$$

The *F1* score metric can be calculated by blending the Precision and Recall into the harmonic mean:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

The threshold η is selected from a set of thresholds, such as $\{10, 25, 50\}$, to assess performance across varying degrees of overlap. The corresponding *F1* score is denoted as *F1@η*.

Mean average precision (*mAP*) further creates a Precision-Recall curve for each segment category based on the IoU outcomes and the supervised segment labels. For each category, a segment is considered a true positive if its IoU with the ground truth exceeds the threshold $\eta/100$ and the predicted category is correct. The average precision (AP) for each category is determined by finding the area under the Precision-Recall curve. The mAP at a specific threshold η, denoted as mAP@η, is then obtained by averaging the AP values across all categories. This metric is different from the F1 score, which relies on a single decision threshold and does not require predicted and actual segment labels. The mAP evaluates the average precision across multiple classes and decision thresholds, making it well-suited for addressing class imbalances in segments, particularly when dealing with multiple classes. By assessing the model's performance on each class separately before averaging, mAP ensures that high performance on majority classes does not overshadow underperformance on minority classes.

3 Framework for deep learning-based methods

This section discusses the integration of deep learning techniques, along with preprocessing and postprocessing methods, for CPD tasks. First, commonly used preprocessing techniques in deep learning-based methods are introduced. The deep learning methods are categorized as supervised, unsupervised, or other methods, based on the presence or absence of labeled data. Typically, these models take preprocessed data as input, learn features, and then output labels or reconstruct data through dense layers or other mechanisms. The section concludes with a discussion on postprocessing methods and real-time detection performance. Figure 3 summarizes the relationships among these components and their roles in the CPD process.

3.1 Preprocessing

3.1.1 Data transformation

In this subsection, we will examine several essential data transformation techniques that play a crucial role in preprocessing data for deep learning applications, especially in the analysis of time-series data for CPD tasks. These techniques include the Discrete Fourier Transform (DFT) (Jiang and Yin 2015; Phan et al., 2018; De Ryck et al., 2021), Fast Fourier Transform (FFT) (Thodoroff et al., 2016; San-Segundo et al., 2019; Tian et al., 2019), Short-Time Fourier Transform (STFT) (Covert et al., 2019; Nejedly et al., 2019; Yuan and Jia, 2019a), and
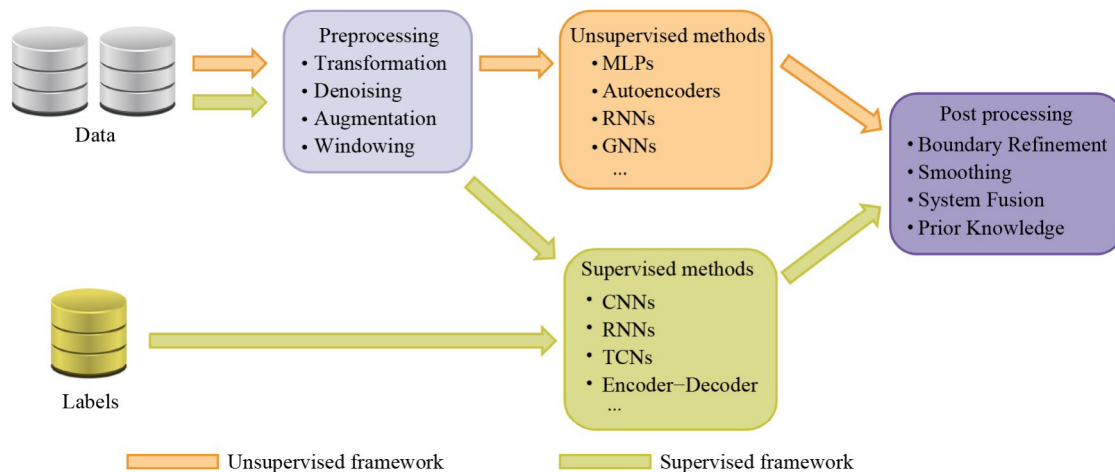

Fig. 3 Overall framework for deep learning-based change-point detection.

wavelet transforms (WTs) (Türk and Özerdem 2019; Verma and Janghel 2021).

Discrete Fourier Transform (*DFT*) is a method used to convert a sequence of time-domain data into its constituent frequencies. This transformation is beneficial for identifying periodic components and anomalies in the data by analyzing the frequency spectrum. DFT is particularly valuable for stationary time series, where frequencies do not change over time.

Fast Fourier Transform (*FFT*) is an algorithm that efficiently computes the DFT, significantly reducing computational complexity from $O(n^2)$ to $O(n\log n)$. This efficiency makes it feasible to process large data sets. The FFT technique is widely used in signal processing for filtering tasks and analyzing complex waveforms.

Short-Time Fourier Transform (*STFT*) extends the concept of Fourier transform to non-stationary signals by dividing the time series into short segments and applying the Fourier transform to each segment separately. This approach provides a time-frequency representation of the signal, enabling the analysis of how frequencies vary over time. This is crucial for detecting changes or anomalies in dynamic systems.

Wavelet Transform (*WT*) provides a multi-resolution analysis of time series, making it highly effective for capturing both frequency and location information of anomalies or changes within a signal. Unlike STFT, which uses a single analysis window, wavelet transforms utilize varying window sizes that adjust according to the frequency level. This provides more precise time and frequency localization.

These transformations are of significant importance in deep learning as they allow for the conversion of raw time-series data into image-like formats that emphasize critical features and patterns. By transforming the data into visual representations such as spectrograms (from STFT) or scalograms (from WT), we can apply image processing techniques, particularly convolutional neural networks (CNNs), to extract hierarchical features from images. Leveraging the spatial processing capabilities of CNNs not only enhances the ability of neural networks to learn from time series data, but also enables more effective detection of significant changes and anomalies.

3.1.2   Denoising

This section explores the denoising process in time-series data, which is often affected by significant noise from background disturbances and bioelectrical interferences. Denoising techniques are crucial for removing noise from the signals, enabling deep learning models to focus on the essential features of the data rather than being distracted by noise. The purpose of denoising is to preprocess the data into a more accurate and less noisy form, thereby enhancing the robustness of deep learning models, reducing training difficulty, and improving overall model performance.

Denoising methods can be broadly categorized into three groups. First, traditional filter-based methods such as lowpass, bandpass, and notch filters are widely used because they effectively separate noise from the signal when they exist in different frequency bands. Common smoothing filters include the median filter and Savitzky-Golay filters (Luo et al., 2017; Alkhodari et al., 2021; Oh and Lee 2022), as well as adaptive filters (Eltrass et al., 2021; Eltrass et al., 2022) that adjust themselves based on the characteristics of the input data.

Secondly, wavelet-based methods utilize the Discrete Wavelet Transform (DWT) to map time-series data into a time-frequency domain (Mathunjwa et al., 2021; Liu et al., 2022b). This technique assumes that important signal features align closely with selected wavelet basis functions. Noise is typically reduced by either eliminating or applying thresholds to wavelet coefficients at higher frequencies where noise is most noticeable. This approach is particularly effective in handling non-stationary noise.

Thirdly, hybrid methods integrate multiple denoising strategies to enhance effectiveness. For example, combining DWT with median or Savitzky-Golay filters can result in stronger noise removal, although it may increase processing time (Jin et al., 2021; Degirmenci et al., 2022).

The choice of a denoising method depends largely on the nature of the noise and the requirements of the subsequent analysis or application. Effective denoising not only improves the accuracy of anomaly detection but also enhances the overall performance of deep learning models by providing them with more relevant data.

3.1.3   Data augmentation

Data augmentation can significantly enhance training data sets, particularly in scenarios with imbalanced classes or limited samples. This section discusses a range of augmentation techniques used in various domains, emphasizing their importance in improving model robustness and addressing data imbalances.

Noise perturbation is a commonly used technique to prevent overfitting in deep learning models by adding noise to existing data (Snyder et al., 2018; Prabhakararao and Dandapat 2022). This simple yet effective approach can be combined with neural networks, such as Denoising Autoencoders (DAE) (Vincent et al., 2010), wherein the model learns to filter out noise and identify the underlying data structure, enhancing its ability to generalize effectively.

Synthetic generation. Creating synthetic data using methods like Generative Adversarial Networks (GAN) (Shaker et al., 2020; Du et al., 2023; Lu et al., 2022) and Variational Auto-Encoders (VAE) (Niu et al., 2020; Du et al., 2022a). This technique is particularly useful in

scenarios where minority classes lack sufficient representation, as it helps balance data sets effectively by generating realistic and complex samples from existing distributions.

Temporal and spatial transformations. For time-series and video data, temporal transformations like random clipping, frame skipping, and introducing random slopes and displacements prevent model bias toward specific segment lengths or trends (Herath et al., 2017). Additionally, randomly deleting samples or introducing variability in sampling rates (random padding or truncating) helps the model perform well under non-uniform sampling conditions (Lattari et al., 2022).

Acoustic and spectral perturbations. In the field of audio processing, various data augmentation strategies such as additive noises, reverberation, speed and pitch perturbations, and spectral augmentation are employed to train robust speaker recognition and speech recognition systems (Shahnawazuddin et al., 2020; Wang et al., 2020a; Bai and Zhang 2021). These methods simulate different acoustic environments and speaking styles, enabling models to generalize better across various auditory conditions.

Oversampling techniques. To address data imbalance, oversampling techniques like Synthetic Minority Oversampling Technique (SMOTE) and its variants are used (He et al., 2020; Singh and Sharma 2022; Ahmad et al., 2023). These techniques generate new samples by interpolating among existing minority samples in feature space, effectively increasing the representation of underrepresented classes in the training set.
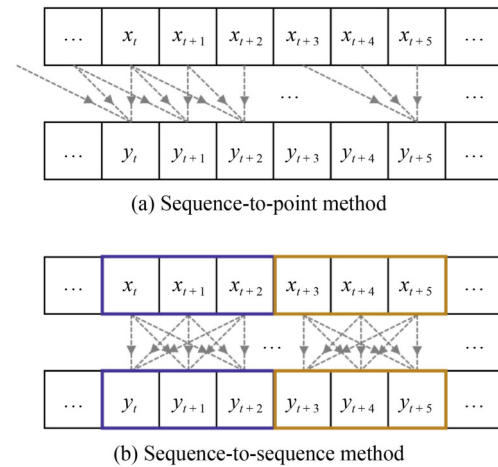
By implementing these diverse data augmentation techniques, deep learning models can acquire more comprehensive and invariant features, enhancing their capabilities in pattern recognition and CPD across different data sets and real-world conditions.

3.1.4 Windowing

Windowing, also known as the use of time windows, is a crucial technique in the analysis of time series data. It is particularly useful for extracting features and preparing inputs for sequential neural networks like Long Short-Term Memory (LSTM) models. This technique serves two primary purposes:

Feature Extraction from Time Windows. Time windows allow for the conversion of time series data into structured formats that are more suitable for analysis and machine learning models. Commonly used features include time domain features, time-frequency domain features, frequency domain features, Fourier transform based features, and wavelet transform based features.

Input Preparation for Sequential Neural Networks. Time windows are essential for training sequence-based models such as LSTMs, which require fixed-length input sequences. The configuration of the window directly affects the model's ability to capture temporal

(a) Sequence-to-point method

(b) Sequence-to-sequence method

Fig. 4 Two frameworks for the adoption of windowing in sequential neural networks.

dependencies. As illustrated in Fig. 4, depending on the neural network's output architecture, windowing can be adapted for different methodologies, such as Sequence-to-Sequence (Seq2Seq) methods and Sequence-to-Point methods. In Sequence-to-Sequence methods, each point within a time window is associated with an output, which can be a category label or latent features. In Sequence-to-Point methods, the entire window is used to predict a single output, focusing on extracting a summary or a decisive outcome from the sequence. The choice between Sequence-to-Sequence and Sequence-to-Point methods depends on the task goal and the data sampling frequency.
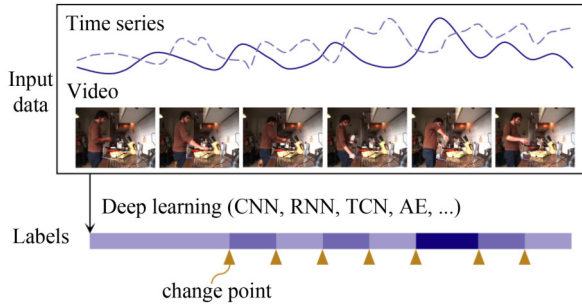
In certain applications, multiple types of windowing may be used simultaneously (Phan et al., 2019a). For example, in sleep staging tasks, an initial window may be used to transform raw signals into spectrograms, which are then treated as image sequences. These sequences can be further analyzed using sliding windows to feed into a sequential neural network, enabling dynamic and continuous learning from the data.

3.2 Supervised methods

In a supervised learning framework, the category labels for the data in the training set are predefined. Deep learning models can be trained on this labeled data to establish a mapping between the input data and the corresponding category labels. Changes in the output labels are interpreted as indications of a detected change point. The general framework is shown in Fig. 5.

3.2.1 The network structure of supervised methods

In supervised learning, the input data typically undergoes feature extraction, where deep features are learned at each time point. These features are then passed through a

Fig. 5 The illustration of supervised framework in deep learning-based CPD tasks.

softmax layer to generate the category label for each time point. The network structures can be broadly classified into several types, based on the feature extraction methods. This section discusses several prominent architectures in detail.

A. Convolutional Neural Networks (CNNs)

CNNs are highly effective at capturing spatial hierarchies in data. They use convolutional kernels—small, trainable filters that slide over input data to extract local features such as edges, textures, and shapes. Based on the dimensionality of the kernels, CNNs are divided into 1D and 2D CNNs.

1D CNNs are well-suited for time-series data, where they analyze one-dimensional sequences to detect patterns and trends. Li et al. (2024) applied a 1D CNN architecture for automatic CPD in time series, consisting of multiple convolutional layers with kernel sizes optimized for capturing temporal patterns. The output is a binary classification indicating the presence or absence of a change-point at each time step. This method performed well, especially in dealing with auto-correlated or heavy-tailed noise.

In contrast, 2D CNNs are typically used for image data, where they process two-dimensional grids to identify visual features. Phan et al. (2019a) proposed a joint classification and prediction framework for automatic sleep stage classification using a 2D CNN. This approach aimed to enhance the accuracy and efficiency of sleep staging by integrating both tasks. Similarly, Aswad et al. (2021) employed a 2D CNN to classify foot gestures for controlling a collaborative robot. Time-series signals from an instrumented insole were converted into 2D images, which the CNN processed to recognize gestures. This method demonstrated high accuracy and robustness, improving the efficiency of robotic control through precise gesture detection.

Researchers have also introduced variants like Fully Convolutional Networks (FCNs), which differ from traditional CNNs by replacing fully connected layers with convolutional ones, enabling output maps that retain spatial information. To overcome issues such as label inconsistency and fixed window sizes in traditional methods, Yao et al. (2018) used an FCN to predict labels

for each time step directly, eliminating the need for window-based segmentation. Their method enables dense and accurate predictions while maintaining computational efficiency. FCNs often include Global Average Pooling (GAP) layers, which reduce model complexity and overfitting by averaging feature maps into a single vector per class. Jeong and Kim (2019) developed an FCN structure for energy-efficient human activity recognition, reducing computational complexity and energy use while maintaining high accuracy, comparable to traditional CNNs, but with significantly lower energy consumption.

B. Recurrent Neural Networks (RNNs)

RNNs, including LSTM and Gated Recurrent Units (GRUs), are designed to process sequential data by maintaining a memory of past inputs. This capability is especially useful for time-series CPD, where temporal dependency is crucial. RNNs capture long-term trends and subtle changes, allowing them to consider the entire history of data points. Bi-directional LSTMs (Bi-LSTMs) extend this by processing data in both forward and backward directions, improving accuracy by capturing both past and future contexts.

Wei et al. (2018) used an RNN-based encoder-decoder architecture to predict labels for each time step. The encoder captures both local and holistic information, while the Segment Detection Unit (SDU) in the decoder localizes segments by combining the decoder state with encoder hidden states. This approach enables dense, accurate predictions with high computational efficiency, outperforming previous methods in temporal action proposals and video summarization. Zhang et al. (2019) introduced the Unbounded Interleaved-State Recurrent Neural Network (UIS-RNN) for speaker diarization, which uses LSTMs with an unbounded interleaved-state structure. This system models each speaker with shared RNN parameters while interleaving states for different speakers over time, identifying whether a segment belongs to a known or new speaker.

In CPD for InSAR time series, Lattari et al. (2022) proposed a method combining LSTM and Time-Gated LSTM (TGLSTM) networks. This approach adds a fully connected layer to the features extracted at each time step, outputting the change-point probability. This method models temporal correlations and handles nonuniformly sampled data effectively.

In sleep staging tasks, Supratak et al. (2017) proposed the DeepSleepNet, combining CNNs for feature extraction and Bi-LSTMs for learning temporal dependencies. This model automatically learns time-invariant features and sleep stage transitions. Building on this, Phan et al. (2019b) proposed SeqSleepNet, an end-to-end hierarchical RNN for sequence-to-sequence sleep staging. Their model uses parallel filterbank layers for preprocessing, attention-based bidirectional RNNs for short-term modeling, and sequence-level bidirectional RNNs for long-term modeling, capturing both local and global temporal

dependencies directly from raw polysomnography data. Phan et al. (2021) further combined SeqSleepNet and DeepSleepNet architectures, applying deep transfer learning by pre-training on the large MASS data set and fine-tuning on smaller data sets. This approach addressed data variability, achieving improved performance in sleep staging for smaller cohorts.

C. CNN-RNN Hybrid Models

CNN-RNN hybrid models combine the strengths of CNNs and RNNs, making them highly effective for complex time-series analysis, particularly when working with image sequences or video data. In these models, 1D CNNs are used to extract sequential numerical and shape features, while 2D CNNs focus on spatial features and capture essential visual details from individual frames. The extracted features are then passed to RNNs, which model the temporal dependencies across the sequence and classify activities (Ordóñez and Roggen 2016; Thodoroff et al., 2016; Chambers and Yoder 2020; Xia et al., 2020). This combination allows the model to leverage both local spatial feature extraction and long-term temporal learning, offering a comprehensive approach to CPD. In some cases, a bidirectional LSTM (BiLSTM) replaces the unidirectional LSTM, improving the model's ability to handle temporal features (Bahrami and Forouzanfar 2022; Ahmad et al., 2023).

Further innovations in network design have expanded on this framework. For example, Hammad et al. (2021) integrated ResNet with 1D CNNs and genetic algorithms to enhance feature extraction. In this approach, CNNs learn features, which are optimized by a genetic algorithm (GA) to identify the most effective feature extraction and classification strategies. Dhekane et al. (2022) proposed an activity recognition method using RNN and LSTM structures for real-time detection and annotation of change points. Instead of directly classifying with LSTM, they used a soft classification approach that monitors change points by calculating the similarity between classification result vectors of adjacent windows. Gaugel and Reichert (2023) introduced a deep learning model named PrecTime, designed for precise time-series segmentation in industrial manufacturing. This hybrid structure consists of CNN-based feature extraction, bidirectional LSTM-based context detection, and a CNN-based prediction refinement stage, aimed at improving segmentation accuracy in multi-phase testing cycles.

D. Temporal Convolutional Networks (TCNs)

TCNs are a type of convolutional network specifically designed for sequential data, using causal convolutions to ensure that predictions depend only on past inputs, preserving temporal order. They also use dilation to capture long-range dependencies without increasing computational complexity, making them an efficient alternative to RNNs for CPD in time-series data.

Farha and Gall (2019) proposed a Multi-Stage Temporal Convolutional Network (MS-TCN) for action segmentation in videos. Their model employs multiple stages of dilated 1D convolutions to first predict and then refine the classification of each video frame, effectively handling the temporal dynamics of actions. A key feature of their approach is the use of a smoothing loss during training, which reduces over-segmentation errors and improves the accuracy of distinguishing between adjacent actions. Li et al. (2023) enhanced this architecture with MS-TCN + + , introducing a dual dilated layer that integrates both large and small receptive fields. This modification optimizes the architecture by decoupling the prediction and refinement phases, achieving better results than MS-TCN with fewer parameters.

E. Encoder-Decoder Architectures

Encoder-decoder models are widely used in sequence-to-sequence tasks. The encoder compresses the input into a latent representation, while the decoder reconstructs a sequence from these latent features. Encoder structures, like CNNs or RNNs, are chosen based on the type of data (e.g., images or sequences). The decoder generates the output sequence, with its structure designed to match the output format. This architecture is particularly effective for handling variable-length data and complex structures.

Perslev et al. (2019) developed U-Time, an encoder-decoder model based on U-Net for sleep stage segmentation. The encoder condenses the input time series into a deep feature representation, while the decoder expands it to classify each time segment. The segmentation results are aggregated using methods such as averaging or mode selection to form final predictions. U-Time can process entire polysomnography recordings in a single pass, significantly improving the efficiency of sleep stage classification without requiring task-specific adjustments or hyperparameter tuning.

F. Autoencoders (AEs)

Autoencoders follow the encoder-decoder architecture, with the main difference being that the output of an AE is a reconstruction of its input. This structure is commonly used for extracting latent features, capturing intrinsic patterns for unsupervised learning. In supervised settings, the latent features can be fed into classifiers for further categorization.

Yuan et al. (2019b) developed an advanced autoencoder for EEG seizure detection, using Short-Time Fourier Transform (STFT) to convert EEG signals into time-frequency spectrograms. These 2D spectrogram images are processed by the AE to extract latent features. Within this framework, they proposed a channel-aware seizure detection module, which directs the model to focus on relevant EEG channels, improving seizure detection accuracy.

G. Attention mechanism

The attention mechanism is a key component in modern neural networks, particularly for sequential data. It allows models to focus on different parts of the input when predicting each part of the output sequence,

enhancing the model's ability to capture dependencies over long sequences. The Transformer architecture employs this mechanism, using self-attention layers to enable parallel processing, which reduces training time and improves performance. This approach is well-suited for tasks involving time-series data, such as language and audio signal processing.

Phan et al. (2018) developed a sleep stage classification model using a bidirectional RNN enhanced with attention. The attention mechanism helps the model focus on the most discriminative parts of the EEG signal, improving feature extraction for classification. Later, Phan et al., (2022) further developed the SleepTransformer, leveraging the Transformer architecture for automatic sleep staging. This model uses self-attention to focus on relevant parts of the input sequence, enhancing the interpretability of its decisions at both the epoch and sequence levels.

H. Others

Huang et al. (2021) proposed CPD-Net, a manifold-based neural network for detecting brain state changes in fMRI data. The model combines a Symmetric Positive Definite Deep Neural Network (SPD-DNN) with a Multi-Stage Recurrent Neural Network (MS-RNN). The SPD-DNN extracts and transforms functional connectivity patterns into a low-dimensional representation on the Riemannian manifold, while the MS-RNN analyzes these patterns over time to detect changes in brain states. This approach effectively tracks dynamic shifts in functional connectivity, addressing the challenge of identifying cognitive state transitions without prior knowledge of experimental setups.

3.2.2 Loss functions

In supervised CPD, the choice of loss function is crucial for effective training and optimal model performance in deep learning models. Typically, these loss functions incorporate elements commonly used in classification tasks to assess the accuracy of predicted categories. One widely used loss function is the categorical cross-entropy loss function.

$$L_{CCE} = -\frac{1}{T} \sum_{t=1}^{T} \sum_{c=1}^{M} y_{t,c} \log p_{t,c},$$

where T is the length of the time series, M is the number of classes, $y_{t,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for observation t, and $p_{t,c}$ is the predicted probability of observation t being of class c.

Moreover, incorporating additional loss modules can enhance the training effectiveness and robustness of the model. For example, incorporating a loss function that considers the continuity of latent variables or output can help maintain the temporal coherence of the data. This

not only reduces the risk of overfitting by smoothing the predictions over time, but also results in more stable detection outcomes with fewer estimated change points. An example of such a loss function is the Kullback Leibler (KL) divergence loss of adjacent classification outputs:

$$L_{Smooth,p} = \frac{1}{T-1} \sum_{t=2}^{T} \sum_{c=1}^{M} p_{t-1,c} (\log p_{t-1,c} - \log p_{t,c}),$$

where $p_{t,c}$ is the predicted probability of observation t being of class c. This loss function optimizes the difference in label probabilities between adjacent time steps, thereby ensuring temporal stability in the model's outputs.

In models like Autoencoders (AEs), including a reconstruction error component in the loss function is essential. This ensures that the learned latent variables effectively reconstruct the input data while still allowing for dimensionality reduction. The mean squared error loss function is commonly used for this purpose.

$$L_{MSE} = \frac{1}{T} \sum_{t=1}^{T} \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2^2,$$

where $\boldsymbol{x}_t$ is the raw time series data for observation t, and $\hat{\boldsymbol{x}}_t$ is the corresponding reconstructed data.

By carefully balancing the components of the loss function, developers can create models that are sensitive to changes in the data, resistant to noise and overfitting, and more reliable for practical applications that involve complex and continuously evolving data dynamics.

3.2.3 Advantages and disadvantages

Advantages. Supervised learning methods generally achieve high accuracy in prediction tasks, particularly when trained on large, well-labeled data sets. The availability of labeled data enables models to learn explicit mappings between inputs and outputs, resulting in precise predictions. Additionally, the structured nature of supervised learning often leads to models that are easier to interpret and validate, which is valuable in applications where understanding the decision-making process is important.

Disadvantages. A significant drawback of supervised learning is its dependency on large amounts of labeled data, which can be time-consuming and expensive to acquire. In many real-world scenarios, labeled data may be scarce or inaccessible. Additionally, supervised models can easily overfit to the training data, especially if the model is overly complex or the data set is not representative of the entire population. This can result in poor generalization to new, unseen data. Lastly, supervised methods are generally less adaptable to new tasks without additional labeled data, limiting their usefulness in rapidly changing environments or novel tasks.

3.3 Unsupervised methods

Unsupervised deep learning for CPD primarily involves the utilization of various neural network architectures to predict or reconstruct time series data. Significant discrepancies, such as large reconstruction errors or substantial differences in latent features between adjacent time points, can indicate potential change points. This approach enables the detection of anomalies or shifts in data patterns without the need for prior labeling, effectively identifying substantial deviations from normal patterns.

The general framework is depicted in Fig. 6. In this framework, neural networks first process the time series data to either predict future values or reconstruct the original data from a compressed representation. By examining the prediction or reconstruction outcomes, the model identifies change points based on either the latent features extracted during the process or the errors generated during prediction/reconstruction. For example, a sudden increase in reconstruction error or a significant shift in latent features between consecutive time points may indicate a change point, highlighting an anomaly or a shift in the underlying data distribution.

3.3.1 The network structure of unsupervised methods

A. Multilayer Perceptrons (MLPs)

The MLP network is a fundamental type of feedforward neural network that consists of multiple layers of nodes, each fully connected to the nodes in the preceding layer, and typically includes at least one hidden layer. MLPs are well-suited for modeling nonlinear relationships where there is no inherent time dependency in the data.

Reznik et al. (2011) employed a multi-layer perceptron (MLP) network to detect signal changes in sensor networks. Their approach utilized MLP networks to predict sensor data based on recent history. By comparing these predictions with actual sensor readings, deviations were detected. Xu et al. (2023a) utilized an MLP to model the melt pool temperature in metal additive manu-

Fig. 6 Illustration of unsupervised framework in deep learning-based CPD tasks.
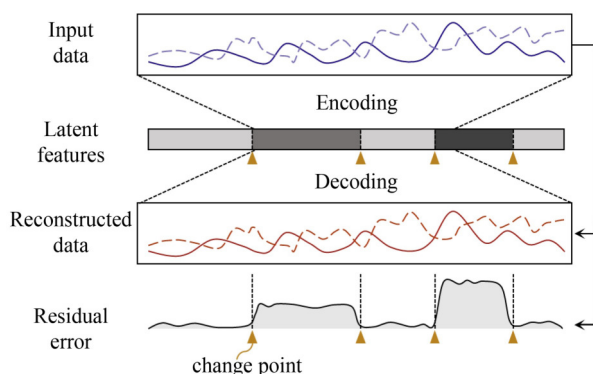
facturing processes. They established the relationship between temperature and other process data. To account for temporal dependencies, their approach integrated a mixed-effects model with a first-order autoregressive process, addressing time series autocorrelation. Based on the residuals between reconstructed and actual data, two efficient online control chart methods were proposed to facilitate anomaly detection.

B. Autoencoders (AEs)

Autoencoders are widely used in the unsupervised deep learning framework, serving as a classic example of data reconstruction neural networks. Unlike supervised tasks that utilize latent features for classification, in unsupervised tasks, autoencoders can detect change points by comparing the temporal differences in these features. When the differences between the low-dimensional space features of adjacent time points are significantly high, it indicates the occurrence of a change point. Lee et al. (2018) assumed continuity in the low-dimensional space, minimizing changes, and used a loss function that combines reconstruction and continuity. By evaluating the differences in the low-dimensional space and identifying local extremes, their method efficiently detects change points. Building upon this framework, De Ryck et al. (2021) developed a novel loss function that promotes time-invariant features, leading to more stable change point identification. This approach significantly enhances the ability to detect subtle and substantial changes in time series data by integrating features extracted from both the time and frequency domains.

Additionally, methods utilizing autoencoders can directly monitor data through the reconstruction process, potentially indicating change points when the model fails to accurately reconstruct the data. Maleki et al. (2021) utilized LSTM networks for both the encoder and decoder stages to handle sequence data effectively. This method enhances detection capabilities by calculating the probability of change points based on the magnitude of the reconstruction error. In the work of Gupta et al. (2022), a change point is detected when this error exceeds a predefined threshold, indicating a significant deviation from the expected data pattern. Atashgahi et al. (2022) also utilized a LSTM-Autoencoder-based neural network for unsupervised online CPD. In their approach, a time step is identified as a potential change point when there is a sudden and sustained increase in the model's reconstruction loss over several time steps.

Furthermore, many researchers leverage advanced techniques such as Graph Neural Networks (GNN) and GAN. Zhang et al. (2020b) developed an integrated method that combines GNNs with AEs in an encoder-decoder framework. Their approach focuses on utilizing feature differences between adjacent time points and combining them with reconstruction error from the autoencoder to detect changes. Du et al. (2023) proposed a novel GAN-based unsupervised anomaly detection

method for multivariate time series. This approach enhances anomaly detection by employing GANs for data augmentation and AEs for data reconstruction. The core technique involves detecting anomalies based on significant reconstruction errors.

C. Recurrent Neural Networks (RNNs)

In unsupervised learning, RNNs are used to predict and reconstruct time-series data by leveraging their ability to retain historical information. Unlike in supervised learning, where RNNs predict labels at each time point, in unsupervised tasks, they focus on identifying patterns and changes in the data. Au Yeung et al. (2020) used a machine learning pattern recognition model for CPD, feeding the reconstruction error into the model to detect potential change points. Wahyono et al. (2020) enhanced detection accuracy by employing stacked LSTM architectures, which provide a deeper understanding of data complexities and improve change-point identification compared to single-layer LSTMs.

Aakur and Sarkar (2019) combined a 2D CNN-based network to encode each time point's image with an RNN-based prediction network to forecast the next time point in a low-dimensional space. The difference between predicted and encoded values is used to identify large discrepancies as change points. Miau and Hung (2020) applied a similar approach using CNNs and Gated Recurrent Units (GRUs) for forecasting river flooding and detecting anomalies. Their model uses CNNs to extract detailed features from river water level data, while GRUs process these features to predict temporal sequences. Anomalies are identified by analyzing the Gaussian distribution of prediction residuals and using Mahalanobis distance to detect deviations.

D. Graph Neural Networks (GNNs)

GNNs operate on graph-structured data, making them effective for capturing patterns at both the node and graph levels. In unsupervised CPD, GNNs can capture dynamic relationships within data, such as in social networks, sensor arrays, or biological networks. By learning representations of graphs at different time points, GNNs can identify significant changes or anomalies in graph structures or node interactions. For example, a sudden shift in connectivity patterns or node features can indicate a change point.

Sulem et al. (2024) proposed a Siamese Graph Neural Network (s-GNN) model to learn a graph similarity function for detecting change points in dynamic networks. Their approach calculates the average similarity between the current graph and its recent history to detect significant deviations as change points. This method addresses the challenge of detecting distribution changes in dynamic networks and provides a robust solution for real-time CPD without delay.

E. Self-supervised methods

Self-supervised learning, a type of unsupervised learning, often improves data differentiation at various stages using contrastive learning. This technique trains models to distinguish between similar and dissimilar data points by forming pairs or groups. Some points are similar (positive examples), while others are dissimilar (negative examples), and the model learns to tell them apart. Contrastive learning uses the data's inherent features to create pseudo-labels, allowing the model to understand the data set's structure without external labels. By maximizing the distance between dissimilar pairs and minimizing it between similar pairs, it helps capture subtle details in the data.

Deldari et al. (2021) proposed a method using an autoregressive deep convolutional network, such as WaveNet, to encode each time-series window. The encoded data are further compressed into a compact embedded representation using a three-layer fully connected network. Change points are detected by calculating the cosine similarity between the embeddings of consecutive time windows, with lower similarity scores indicating potential change points. The encoder was trained with a contrastive learning approach, using consecutive time windows as positive examples and temporally separated windows as negative examples.

F. Weakly-supervised methods

Ding and Xu (2018) developed a weakly-supervised approach for segmenting human actions in untrimmed videos. Unlike fully supervised methods that rely on category labels, this approach uses only action transcripts. Their framework integrates a Temporal Convolutional Feature Pyramid Network (TCFPN) with an Iterative Soft Boundary Assignment (ISBA) strategy, which iteratively refines action boundaries. During each iteration, the network uses the previous cycle's results as labels, reclassifying each observation to refine the boundaries. This process is repeated until the results converge.

3.3.2   Loss functions

Unsupervised deep learning approaches commonly employ a reconstruction error as a primary component of the loss function. This error measures the discrepancy between the input data and its reconstructed output:

$$L_{MSE} = \frac{1}{T} \sum_{t=1}^{T} \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2^2.$$

To improve the capability of these models in capturing subtle and gradual changes in the data, some approaches incorporate an additional term in the loss function that penalizes discontinuities in the deep feature representations extracted over time. This penalty on temporal continuity ensures that the learned features evolve smoothly over time, reducing the likelihood of abrupt changes unless strongly supported by the data. By encouraging the preservation of temporal smoothness, this penalty helps differentiate genuine change points from random

fluctuations or noise in the data. The common loss function includes the l_2 norm of the difference and the cosine function of features:

$$L_{Smooth,h}^{(1)} = -\frac{1}{T-1}\sum_{t=2}^{T}\sum_{c=1}^{M}\|\boldsymbol{h}_t - \boldsymbol{h}_{t-1}\|_2^2,$$

or

$$L_{Smooth,h}^{(2)} = \frac{1}{T-1}\sum_{t=2}^{T}\sum_{c=1}^{M}\cos(\boldsymbol{h}_t, \boldsymbol{h}_{t-1}),$$

where $\boldsymbol{h}_t$ is the deep features of the data for observation t.

This combination of reconstruction error and temporal continuity penalty in the loss function effectively allows the model to balance between sensitivity to changes and stability against noise. The reconstruction error ensures that the model responds to anomalies, while the continuity penalty maintains a coherent temporal progression of the learned features, facilitating the identification and verification of significant changes in the data stream. Such an approach to the loss function is advantageous for developing robust CPD models that are sensitive to real changes and resilient against false positives.

In self-supervised models, the contrastive loss is designed to minimize the distance between embeddings of consecutive time windows, known as positive pairs $(\boldsymbol{h}_i, \boldsymbol{h}_{i+1})$, and to maximize the distance between embeddings of non-consecutive, temporally distant windows, referred to as negative pairs $(\boldsymbol{h}_i, \boldsymbol{h}_{j+1})$, where $i \neq j$.

$$L_{con} = -\frac{1}{T-1}\sum_{i=2}^{T}Sim(\boldsymbol{h}_i, \boldsymbol{h}_{i+1}) + \lambda\frac{2}{(T-1)(T-2)}\sum_{i\neq j}Sim(\boldsymbol{h}_i, \boldsymbol{h}_{j+1}),$$

where the *Sim* function is a similarity function such as the cosine function, and λ is a pre-determined parameter used to make a balance. Sometimes, sampling methods are used for the negative pairs to reduce computational costs.

3.3.3 Advantages and disadvantages

Advantages. Unsupervised learning methods do not rely on labeled data, making them suitable for scenarios where labeling is impractical or costly. This allows for the analysis of large data sets without the need for extensive manual labeling. Therefore, unsupervised learning can often scale to large data sets more easily than supervised methods, which can be hindered by the requirement for labeled data. Additionally, unsupervised methods are also great at discovering hidden structures or patterns within the data, such as clustering similar data points or detecting anomalies. This can lead to insights that were not initially apparent.

Disadvantages. Unsupervised models often yield less interpretable results compared to supervised models. The absence of labeled outcomes makes it difficult to understand the significance of discovered patterns or validate the findings. Unlike supervised learning, which allows direct performance measurement by comparing predictions to true labels, unsupervised learning lacks straightforward evaluation metrics. This can make it challenging to assess model quality and compare different unsupervised approaches. In tasks where specific labels are crucial, unsupervised methods may not perform as well as supervised methods. The indirect nature of relying on pattern discovery rather than explicit label prediction can lead to less accurate outcomes in such cases.

3.4 Post processing

Post-processing plays a crucial role in refining the outputs of CPD systems based on neural networks, significantly improving both accuracy and usability of the detections. These methods address issues such as over-segmentation, boundary refinement, and improvement of segment coherence. The following are the four primary types of post-processing techniques:

Boundary Refinement. Techniques such as pooling operations and leveraging confident predictions from later stages of models like Multi-Stage Temporal Convolutional Networks (Wang et al., 2020c) effectively mitigate over-segmentation, a common problem where too many change points are detected. Additionally, integrating dedicated boundary detection networks aids in achieving more precise determination of change points (Ishikawa et al., 2021). Advanced methods like non-maximum suppression (Gao et al., 2017; Wei et al., 2018; Park et al., 2022) in action detection are also utilized to refine outputs by merging closely neighboring proposals. These techniques reduce error rates and enhance the reliability of the model.

Smoothing Based on Continuity. Various smoothing techniques are employed to improve the continuity of detected segments. Gaussian smoothing (Ding and Yao 2022; Du et al., 2022b) utilizes kernels to mitigate abrupt changes across sequences, while filtering methods like moving average filtering (De Ryck et al., 2021; Deldari et al., 2021) adjust the outputs to reduce noise and provide a smoother transition between detected states. For anomaly detection tasks, the point-adjust method (Du et al., 2023) assumes that anomalous observations typically occur consecutively. Therefore, if the detection model identifies an anomalous point within a time segment, all points within that segment are considered anomalous. These smoothing processes ensure the coherence of the segment sequence and help align the detection with the natural progression of the underlying data.

System Fusion: This involves integrating results from multiple systems to resolve conflicts and enhance overall detection accuracy. Techniques like DOVER (Diarization Output Voting Error Reduction) and its modifications, such as DOVER-Lap, aggregate and reconcile outputs by aiming for consensus among different systems. This approach is particularly useful in tasks like speaker diarization (Stolcke and Yoshioka 2019; Raj et al., 2021).

Integration of Prior Knowledge. Some methods incorporate prior knowledge, such as action length distributions, to guide post-processing steps. For example, to achieve temporal action localization in untrimmed videos, action length distributions are integrated as priors into a neural network architecture. This integration helps guide the temporal boundary regression process within the model (Shou et al., 2016; Gao et al., 2017). By incorporating these priors, these approaches are better able to estimate where an action is likely to start and end, resulting in more accurate action proposals.

When these post-processing techniques are combined synergistically, neural network-based CPD systems can achieve higher precision and reliability.

4 Challenges and prospects

This section outlines the limitations and challenges of current deep learning-based CPD methods. We also discuss various open research questions and potential future research directions.

4.1 Insufficient labeling

One of the primary challenges in developing effective CPD methods is the scarcity of labeled data. While unsupervised methods have made recent advancements, supervised approaches still dominate. Supervised learning algorithms heavily rely on extensive and accurately labeled data sets for achieving high performance. However, obtaining such labeled data are often labor-intensive and expensive, especially in fields that require domain-specific expertise. To mitigate the issue of insufficient labeled data, data augmentation strategies can be employed, such as introducing artificial change points in the training set (Carmona et al., 2021). These strategies enhance the diversity and quantity of training data, thereby improving the robustness and accuracy of CPD models.

In some scenarios, there is a significant mismatch between the training and deployment environments, where conditions during deployment can differ greatly from those during training (Chen et al., 2020; Gammulle et al., 2023). This mismatch, combined with the limited availability of labeled data in deployment settings, often leads to erroneous decisions due to changes in data acquisition settings, operating conditions, or sensor modalities.

To address this issue, the design of generalizable models is essential for the real-world implementation of CPD methods. One promising approach to tackle this mismatch is transfer learning, which includes meta-learning and domain generalization. Meta-learning enables models trained for a specific task to quickly adapt to new situations without extensive retraining (Coskun et al., 2023). On the other hand, domain generalization techniques allow a model trained on one domain to adapt to another domain with minimal adjustments (Zhu et al., 2018). A good transfer learning method can significantly reduce the need for large labeled data sets by utilizing pretrained models on related tasks and fine-tuning them on smaller, task-specific data sets.

Another advanced technique in this field is zero-shot learning (Zhang et al., 2020a; Zhang et al., 2022) and few-shot learning (Feng and Duarte 2019; Ben-Ari et al., 2021), which aim to transfer knowledge from well-represented classes to unseen or rarely represented classes with minimal or no training samples. For example, zero-shot temporal activity detection (ZSTAD) generalizes action detection methods to newly emerging or rare events not included in the training set, despite the inherent challenges of localizing and detecting novel action classes in untrimmed videos.

Despite these advancements, significant challenges still exist. Many methods lack robustness and struggle to generalize across different domains and data sets, highlighting the need for further research into more resilient and adaptable CPD algorithms.

4.2 Multi-modality data

The integration of multi-modal data presents both opportunities and challenges. Combining data from various sources, such as audio, visual, and sensor data, can improve detection accuracy but also increase the complexity of the models and computational requirements. Effective data fusion techniques that seamlessly integrate information from multiple modalities are necessary. For example, in speaker diarization, combining audio and visual data significantly enhances the accuracy of speaker identification (Ding et al., 2020). However, as pointed out by Wang et al. (2020b), many existing multimodal methods are not as effective as expected due to various challenges, such as overfitting.

Additionally, one major challenge in handling multimodality data are the synchronization of data from different sources. Misaligned data can result in inaccurate change detection and decreased model performance. Hardware solutions, like timestamping sensor signals, are commonly used to align data (Chen et al., 2015; Soraya et al., 2019). When timestamps are unavailable, sophisticated algorithms for temporal alignment of multi-source data are required. Techniques related to dynamic time warping (DTW) can be adapted to accurately align

time-series data (Xu et al., 2023b).

The development of standardized multimodal data sets is crucial due to the lack of diversity and complexity in existing data sets. For instance, in the domain of Human Activity Classification, there is a scarcity of publicly available data sets that involve multi-modal sensing with vision and inertial data (Majumder and Kehtarnavaz 2021). Moving forward, it is important for research to focus on creating comprehensive multimodal data sets and developing advanced fusion techniques that can effectively integrate different types of data. This will enhance the robustness and accuracy of CPD systems in various applications.

4.3 Online detection

In real-world applications like network congestion and traffic system bottlenecks, change points play a critical role in guiding timely actions (Ma et al., 2015). Therefore, the early detection of change points and minimizing detection delay is crucial. The delay in detection typically stems from computational delay and model sensitivity.

First, computational delay arises from the time required for model processing. Many CPD methods achieve high performance but at the cost of increased computational complexity. The focus on accuracy has led to the development of deep and complex architectures that are computationally expensive, making real-time online detection impractical. Hence, it is important to further research on reducing computational costs and resource consumption (e.g., CPU, GPU, and energy usage) (Lin et al., 2019; Cheng et al., 2020).

Secondly, detection delay can also be attributed to the model's sensitivity to change points. Following a change point, the model often needs to accumulate several data points before detecting the change. A model with higher sensitivity minimizes detection delay but may result in more false alarms. On the other hand, a less sensitive model may have fewer false alarms but longer detection delays. Striking a balance between computational cost and model sensitivity is crucial for achieving overall satisfactory detection accuracy and delay. Larger models with enhanced modeling capabilities tend to be more sensitive to changes but can also cause significant detection delays due to their computational cost.

Furthermore, the length of the time window used also affects detection delay. Larger time windows typically result in higher detection delays. Hence, it is vital to carefully design the length of the time window, ensuring detection accuracy while minimizing delay. One innovative approach is to dynamically adjust the time window size based on the trends in the time series. This allows the system to effectively handle diverse data characteristics without compromising accuracy (Wang et al., 2022).

Moreover, existing research often fails to consider online metrics in experimental validation. Many methods claim to be online but are primarily validated based on CPD accuracy rather than online metrics such as detection delay. Future research should place more emphasis on these aspects to improve online performance in practical applications.

4.4 Model interpretability

Interpretable models are crucial for gaining insights into detected change points, understanding their causes, and making informed decisions. Deep learning-based methods often face criticism for their lack of interpretability due to their black-box nature, compared to statistical CPD methods. While supervised methods can identify the types of change points, the specific characteristics and causes of these changes remain unknown. Unsupervised methods are even less interpretable, as they determine change points based only on the degree of change between adjacent points, without providing deeper insights.

In the financial domain, interpretability is particularly important due to the high stakes involved in decision-making processes. Financial analysts and decision-makers need to understand not only when a change point occurs but also why it happens and what it signifies for future trends. For example, a detected change point in stock prices might indicate the onset of market volatility or the impact of an external event, such as a policy change or economic report. Therefore, models used in financial CPD must be capable of providing accurate and understandable explanations to non-technical stakeholders.

Additionally, change-point phenomena may be limited to specific dimensions of the data or certain parts of the network, while most other dimensions or data structures remain unchanged (Sulem et al., 2024). Therefore, an important improvement to current frameworks is the ability to pinpoint which parts of the data have changed. The Sort-k pooling layer can enhance interpretability by selecting and ranking the most informative features, explicitly highlighting the specific data points or dimensions where changes occur (Zhang et al., 2018).

Furthermore, incorporating methods such as layer-wise relevance propagation (LRP) can further enhance model interpretability. For instance, combining ensembles of convolutional neural networks with LRP has been used to detect brain features contributing to Brain-age (Hofmann et al., 2022). The LRP algorithm highlights relevant areas in the input that support or dismiss corresponding output decisions, providing a clear visualization of the most influential input features.

Incorporating domain-specific prior knowledge into models is a promising area of research. For example, in multi-stage processes, experts can provide valuable insights on variables that are likely to be correlated across different stages. By leveraging this information, we can simplify model architecture, reduce complexity, and

improve model interpretability. Nevertheless, effectively integrating such prior knowledge into the modeling process can be challenging.

4.5   Decision-making

The integration of CPD results with decision-making is crucial. The ultimate goal of detecting change points is to enable timely and informed actions that mitigate risks, capitalize on opportunities, or adapt to new conditions. In practical applications, when a change point is detected, it triggers a decision-making process that varies depending on the specific domain. In certain domains, real-time emergency responses are immediately activated upon the detection of a change point or anomaly. For instance, in industrial monitoring, CPD can lead to predictive maintenance actions, such as replacing a component before it fails (Maleki et al., 2021). In healthcare, the detection of change points in patient monitoring data might prompt immediate interventions to prevent adverse outcomes (Ahmad et al., 2023).

However, in more complex scenarios, the decision space is larger, and the relationship between CPD and decision-making becomes more complex. For instance, in financial markets, the detection of a change point might indicate the need for portfolio reallocation or a shift in trading strategies. This requires models to provide interpretable information that supports human decision-making or to integrate CPD models with automated decision-making processes, such as those driven by reinforcement learning. After a decision is made, it is important to assess its impact, creating a feedback loop that continuously learns and improves the integration of CPD with decision-making. This iterative process helps enhance the effectiveness and reliability of CPD in guiding critical decisions across various domains.

In the field of autonomous driving, Galceran et al. (2017) utilized Bayesian changepoint detection to analyze the behavior of surrounding vehicles in real-time. They employed a multipolicy framework to simulate and predict potential future actions. Based on these predictions, they determined the optimal driving policy for the autonomous vehicle to ensure safe and efficient navigation.

In their work, Wang et al. (2019) proposed a method that integrates deep reinforcement learning, specifically a Deep Q-Network (DQN), with rule-based constraints to make lane change decisions in autonomous driving. The DQN was trained to optimize lane-changing actions by considering both safety and efficiency, while following high-level decisions informed by rule-based modifications. However, these methods typically do not directly integrate changepoint detection with decision-making processes. Effectively incorporating detection results into decision-making optimization objectives is a challenging and valuable area for further research.

5   Conclusions

CPD methods have derived great advances in a variety of areas, such as industrial manufacturing, healthcare, human activity monitoring, financial data analysis, and environmental monitoring, which improved our understanding of and response to change in these varied areas. Deep learning methods have advanced CPD methods by creating new detection processes that allow for better detection capabilities with greater efficiency and scalability, which align particularly well with the characteristics of new high-dimensional data sets.

This review examines major innovations in deep learning-based CPD methods, taking into consideration a variety of perspectives such as supervised and unsupervised learning frameworks. It provides a comprehensive examination of the key data sets, rigorous performance analysis, practical applications, and review future directions for the following research. The in-depth discussion of CPD methods in this review will be a strong motivation for continued innovation in the framework of different scientific and technology communities.

References

Aakur S N, Sarkar S (2019). A perceptual prediction framework for self supervised event segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1197–1206

Ahmad I, Wang X, Javeed D, Kumar P, Samuel O W, Chen S (2023). A hybrid deep learning approach for epileptic seizure detection in eeg signals. IEEE Journal of Biomedical and Health Informatics, in press

Ahmad S, Lavin A, Purdy S, Agha Z (2017). Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262: 134–147

Alkhodari M, Apostolidis G, Zisou C, Hadjileontiadis L J, Khandoker

A H (2021). Swarm decomposition enhances the discrimination of cardiac arrhythmias in varied-lead ECG using ResNet-BiLSTM network activations. 2021 Computing in Cardiology (CinC), IEEE. 1–4

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 (2019). Real-time change point detection with application to smart home time series data. IEEE Transactions on Knowledge and Data Engineering, 31(5): 1010–1023

Andrzejak R G, Lehnertz K, Mormann F, Rieke C, David P, Elger C E (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 64(6): 061907

Aswad F E, Djogdom G V T, Otis M J D, Ayena J C, Meziane R (2021). Image generation for 2D-CNN using time-series signal features from foot gesture applied to select cobot operating mode. Sensors, 21(17): 5743

Atashgahi Z, Mocanu D C, Veldhuis R N, Pechenizkiy M (2022). Memory-free online change-point detection: A novel neural network approach

Au Yeung J F, Wei Z, Chan K Y, Lau H Y K, Yiu K F C (2020). Jump detection in financial time series using machine learning algorithms. Soft Computing, 24(3): 1789–1801

Bahrami M, Forouzanfar M (2022). Sleep apnea detection from single-lead ECG: A comprehensive analysis of machine learning and deep learning algorithms. IEEE Transactions on Instrumentation and Measurement, 71: 1–11

Bai Z, Zhang X L (2021). Speaker recognition based on deep learning: An overview. Neural Networks, 140: 65–99

Basseville M, Nikiforov I V (1993). Detection of abrupt changes: Theory and application. Prentice hall Englewood Cliffs

Ben-Ari R, Nacson M S, Azulai O, Barzelay U, Rotman D (2021). TAEN: temporal aware embedding network for few-shot action recognition. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2786–2794

Caba Heilbron F, Escorcia V, Ghanem B, Carlos Niebles J (2015). Activitynet: A large-scale video benchmark for human activity understanding. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 961–970

Cabrieto J, Tuerlinckx F, Kuppens P, Grassmann M, Ceulemans E (2017). Detecting correlation changes in multivariate time series: A comparison of four non-parametric change point detection methods. Behavior Research Methods, 49(3): 988–1005

Carletta J (2007). Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. Language Resources and Evaluation, 41(2): 181–190

Carmona C U, Aubet F X, Flunkert V, Gasthaus J (2021). Neural contextual anomaly detection for time series. arXiv preprint arXiv:2107.07702

Chambers R D, Yoder N C (2020). FilterNet: A many-to-many deep learning architecture for time series classification. Sensors, 20(9): 2498

Chavarriaga R, Sagha H, Calatroni A, Digumarti S T, Tröster G, Millán J R, Roggen D (2013). The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, 34(15): 2033–2042

Chen C, Jafari R, Kehtarnavaz N (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. 2015 IEEE International conference on image processing (ICIP), IEEE. 168–172

Chen G, Lu G, Liu J, Yan P (2019). An integrated framework for statistical change detection in running status of industrial machinery under transient conditions. ISA Transactions, 94: 294–306

Chen H (2019). Sequential change-point detection based on nearest neighbors. Annals of Statistics, 47(3): 1381–1407

Chen H, Chu L (2023). Graph-based change-point analysis. Annual Review of Statistics and Its Application, 10(1): 475–499

Chen M H, Li B, Bao Y, AlRegib G, Kira Z (2020). Action segmentation with joint self-supervised temporal domain adaptation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 9454–9463

Cheng K, Zhang Y, He X, Chen W, Cheng J, Lu H (2020). Skeleton-based action recognition with shift graph convolutional network. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 183–192

Chopin N (2007). Dynamic detection of change points in long time series. Annals of the Institute of Statistical Mathematics, 59(2): 349–366

Coskun H, Zia M Z, Tekin B, Bogo F, Navab N, Tombari F, Sawhney H S (2023). Domain-specific priors and meta learning for few-shot first-person action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(6): 6659–6673

Covert I C, Krishnan B, Najm I, Zhan J, Shore M, Hixson J, Po M J (2019). Temporal graph convolutional networks for automatic seizure detection. Machine learning for healthcare conference, PMLR. 160–180

Damen D, Doughty H, Farinella G M, Fidler S, Furnari A, Kazakos E, Moltisanti D, Munro J, Perrett T, Price W (2018). Scaling egocentric vision: The epic-kitchens dataset. Proceedings of the European Conference on Computer Vision (ECCV). 720–736

Damen D, Doughty H, Farinella G M, Furnari A, Kazakos E, Ma J, Moltisanti D, Munro J, Perrett T, Price W, Wray M (2022). Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision, 130(1): 33–55

De Ryck T, De Vos M, Bertrand A (2021). Change point detection in time series data using autoencoders with a time-invariant representation. IEEE Transactions on Signal Processing, 69: 3513–3524

Degirmenci M, Ozdemir M A, Izci E, Akan A (2022). Arrhythmic heartbeat classification using 2d convolutional neural networks. IRBM, 43(5): 422–433

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

Dhekane S G, Tiwari S, Sharma M, Banerjee D S (2022). Enhanced annotation framework for activity recognition through change point detection. In: 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE. 397–405

Ding G, Sener F, Yao A (2023). Temporal action segmentation: An analysis of modern techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(2): 1011–1030

Ding G, Yao A (2022). Temporal action segmentation with high-level complex activity labels. IEEE Transactions on Multimedia, 25: 1928–1939

Ding L, Xu C (2018). Weakly-supervised action segmentation with iterative soft boundary assignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 6508–6516

Ding Y, Xu Y, Zhang S X Cong Y, Wang L (2020). Self-supervised learning for audio-visual speaker diarization. ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. 4367–4371

Du B, Sun X, Ye J, Cheng K, Wang J, Sun L (2023). GAN-based anomaly detection for multivariate time series using polluted training set. IEEE Transactions on Knowledge and Data Engineering, 35(12): 12208–12219

Du C, Liu P X, Zheng M (2022a). Classification of imbalanced electro-cardiosignal data using convolutional neural network. Computer Methods and Programs in Biomedicine, 214: 106483

Du Z, Wang X, Zhou G, Wang Q (2022b). Fast and unsupervised action boundary detection for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 3323–3332

Eltrass A S, Tayel M B, Ammar A I (2021). A new automated CNN deep learning approach for identification of ECG congestive heart failure and arrhythmia using constant-Q non-stationary Gabor transform. Biomedical Signal Processing and Control, 65: 102326

Eltrass A S, Tayel M B, Ammar A I (2022). Automated ECG multi-class classification system based on combining deep learning features with HRV and ECG measures. Neural Computing & Applications, 34(11): 8755–8775

Farha Y A, Gall J (2019). Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 3575–3584

Fathi A, Ren X, Rehg J M (2011). Learning to recognize objects in egocentric activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE. 3281–3288

Feng S, Duarte M F (2019). Few-shot learning-based human activity recognition. Expert Systems with Applications, 138: 112782

Galceran E, Cunningham A G, Eustice R M, Olson E (2017). Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. Autonomous Robots, 41(6): 1367–1382

Gammulle H, Ahmedt–Aristizabal D, Denman S, Tychsen-Smith L, Petersson L, Fookes C (2023). Continuous human action recognition for human–machine interaction: A review. ACM Computing Surveys, 55(13s): 1–38

Gao J, Yang Z, Chen K, Sun C, Nevatia R (2017). Turn tap: Temporal unit regression network for temporal action proposals. Proceedings of the IEEE international conference on computer vision. 3628–3636

Gaugel S, Reichert M (2023). PrecTime: A deep learning architecture for precise time series segmentation in industrial manufacturing operations. Engineering Applications of Artificial Intelligence, 122: 106078

Godfrey J J, Holliman E (1997). Switchboard-1 Release 2. Linguistic Data Consortium, Philadelphia, 926: 927

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: 118260

Guttag J (2010). CHB-MIT Scalp EEG Database (version 1.0.0). PhysioNet

Habibi R (2022). Bayesian online change point detection in finance. Financial Internet Quarterly, 17(4): 27–33

Hammad M, Iliyasu A M, Subasi A, Ho E S, El-Latif A A A (2021). A multitier deep learning model for arrhythmia detection. IEEE Transactions on Instrumentation and Measurement, 70: 1–9

He J, Rong J, Sun L, Wang H, Zhang Y, Ma J (2020). A framework for cardiac arrhythmia detection from IoT-based ECGs. World Wide Web, 23(5): 2835–2850

Herath S, Harandi M, Porikli F (2017). Going deeper into action recognition: A survey. Image and Vision Computing, 60: 4–21

Hofmann S M, Beyer F, Lapuschkin S, Goltermann O, Loeffler M, Müller K R, Villringer A, Samek W, Witte A V (2022). Towards the interpretability of deep learning models for multi-modal neuroimaging: Finding structural changes of the ageing brain. NeuroImage, 261: 119504

Huang Z, Cai H, Dan T, Lin Y, Laurienti P, Wu G (2021). Detecting brain state changes by geometric deep learning of functional dynamics on Riemannian manifold. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VII 24, Springer. 543–552

Imtiaz S A (2021). A systematic review of sensing technologies for wearable sleep staging. Sensors, 21(5): 1562

Ishikawa Y, Kasai S, Aoki Y, Kataoka H (2021). Alleviating over-segmentation errors by detecting action boundaries. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2322–2331

Jaiswal R, Lohani A, Tiwari H (2015). Statistical analysis for change detection and trend assessment in climatological parameters. Environmental Processes, 2(4): 729–749

Jeong C Y, Kim M (2019). An energy-efficient method for human activity recognition with segment-level change detection and deep learning. Sensors (Basel), 19(17): 3688

Jiang W, Yin Z (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In: Proceedings of the 23rd ACM international conference on Multimedia. 1307–1310

Jin Y, Liu J, Liu Y, Qin C, Li Z, Xiao D, Zhao L, Liu C (2021). A novel interpretable method based on dual-level attentional deep neural network for actual multilabel arrhythmia detection. IEEE Transactions on Instrumentation and Measurement, 71: 1–11

Kawaguchi N, Yang Y, Yang T, Ogawa N, Iwasaki Y, Kaji K, Terada T, Murao K, Inoue S, Kawahara Y (2011). In: HASC2011corpus: Towards the common ground of human activity recognition. In: Proceedings of the 13th International Conference on Ubiquitous Computing. 571–572

Kawahara Y, Yairi T, Machida K (2007). Change-point detection in time-series data based on subspace identification. Seventh IEEE International Conference on Data Mining (ICDM 2007), IEEE.

559–564

Kemp B, Zwinderman A H, Tuk B, Kamphuisen H A, Oberye J J (2000). Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. IEEE Transactions on Biomedical Engineering, 47(9): 1185–1194

Keogh E, Chu S, Hart D, Pazzani M (2001). An online algorithm for segmenting time series. In: Proceedings 2001 IEEE International Conference on Data Mining, IEEE. 289–296

Khan F A, Haldar N A H, Ali A, Iftikhar M, Zia T A, Zomaya A Y (2017). A continuous change detection mechanism to identify anomalies in ECG signals for WBAN-based healthcare environments. IEEE Access : Practical Innovations, Open Solutions, 5: 13531–13544

Khan N, McClean S, Zhang S, Nugent C (2016). Optimal parameter exploration for online change-point detection in activity monitoring using genetic algorithms. Sensors (Basel), 16(11): 1784

Kuehne H, Arslan A, Serre T (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 780–787

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

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: 106101

Lee W H, Ortiz J, Ko B, Lee R (2018). Time series segmentation through automatic feature learning. arXiv preprint arXiv:1801.05394

Li J, Fearnhead P, Fryzlewicz P, Wang T (2024). Automatic change-point detection in time series via deep learning. Journal of the Royal Statistical Society. Series B, Statistical Methodology, 86(2): 273–285

Li S, Farha Y A, Liu Y, Cheng M M, Gall J (2023). Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(6): 6647–6658

Lin J, Gan C, Han S (2019). Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7083–7093.

Liu B, Zhang X, Liu Y (2022a). High dimensional change point inference: Recent developments and extensions. Journal of Multivariate Analysis, 188: 104833

Liu Z, Zhou B, Jiang Z, Chen X, Li Y, Tang M, Miao F (2022b). Multiclass arrhythmia detection and classification from photo-plethysmography signals using a deep convolutional neural network. Journal of the American Heart Association, 11(7): e023555

Lu H, Du M, Qian K, He X, Wang K (2022). GAN-based data augmentation strategy for sensor anomaly detection in industrial robots. IEEE Sensors Journal, 22(18): 17464–17474

Luo K, Li J, Wang Z, Cuschieri A (2017). Patient-specific deep architectural model for ECG classification. Journal of Healthcare Engineering, 2017(1): 4108720

Luo X, Hu Y (2024). Temporal misalignment in intensive longitudinal data: consequences and solutions based on dynamic structural equation models. Structural Equation Modeling, 31(1): 118–131

Ma X, Yu H, Wang Y, Wang Y (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS One, 10(3): e0119044

Majumder S, Kehtarnavaz N (2021). Vision and inertial sensing fusion for human action recognition: A review. IEEE Sensors Journal, 21(3): 2454–2467

Maleki S, Maleki S, Jennings N R (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. Applied Soft Computing, 108: 107443

Mathunjwa B M, Lin Y T, Lin C H, Abbod M F, Shieh J S (2021). ECG arrhythmia classification by using a recurrence plot and convolutional neural network. Biomedical Signal Processing and Control, 64: 102262

Matteson D S, James N A (2014). A nonparametric approach for multiple change point analysis of multivariate data. Journal of the American Statistical Association, 109(505): 334–345

Miau S, Hung W H (2020). River flooding forecasting and anomaly detection based on deep learning. IEEE Access: Practical Innovations, Open Solutions, 8: 198384–198402

Moody G B, Mark R G (2001). The impact of the MIT-BIH arrhythmia database. IEEE Engineering in Medicine and Biology Magazine, 20(3): 45–50

Nejedly P, Kremen V, Sladky V, Nasseri M, Guragain H, Klimes P, Cimbalnik J, Varatharajah Y, Brinkmann B H, Worrell G A (2019). Deep-learning for seizure forecasting in canines with epilepsy. Journal of Neural Engineering, 16(3): 036031

Niu Y S, Hao N, Zhang H (2016). Multiple change-point detection: a selective overview. Statistical Science, 31: 611–623

Niu Z, Yu K, Wu X (2020). LSTM-based VAE-GAN for time-series anomaly detection. Sensors (Basel), 20(13): 3738

O'reilly C, Gosselin N, Carrier J, Nielsen T (2014). Montreal Archive of Sleep Studies: an open–access resource for instrument benchmarking and exploratory research. Journal of Sleep Research, 23(6): 628–635

Oh S, Lee M (2022). A shallow domain knowledge injection (sdk-injection) method for improving cnn-based ecg pattern classification. Applied Sciences, 12(3): 1307

Oh S M, Rehg J M, Balch T, Dellaert F (2008). Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. International Journal of Computer Vision, 77(1–3): 103–124

Olsen N L, Markussen B, Raket L L (2018). Simultaneous inference for misaligned multivariate functional data. Applied Statistics, 67(5): 1147–1176

Ordóñez F J, Roggen D (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1): 115

Page E S (1954). Continuous inspection schemes. Biometrika, 41(1/2): 100–115

Park T J, Kanda N, Dimitriadis D, Han K J, Watanabe S, Narayanan S (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72: 101317

Penzel T, Moody G B, Mark R G, Goldberger A L, Peter J H (2000). The apnea-ECG database. Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163), IEEE. 255–258

Perslev M, Jensen M, Darkner S, Jennum P J, Igel C (2019). U-time: A fully convolutional network for time series segmentation applied to sleep staging. Advances in Neural Information Processing Systems, 32

Phan H, Andreotti F, Cooray N, Chén O Y, De Vos M (2018). Automatic sleep stage classification using single-channel? Learning sequential features with attention-based recurrent neural networks. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE. 1452–1455

Phan H, Andreotti F, Cooray N, Chén O Y, De Vos M (2019a). Joint classification and prediction CNN framework for automatic sleep stage classification. IEEE Transactions on Biomedical Engineering, 66(5): 1285–1296

Phan H, Andreotti F, Cooray N, Chén O Y, De Vos M (2019b). SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(3): 400–410

Phan H, Chén O Y, Koch P, Lu Z, McLoughlin I, Mertins A, De Vos M (2021). Towards more accurate automatic sleep staging via deep transfer learning. IEEE Transactions on Biomedical Engineering, 68(6): 1787–1798

Phan H, Mikkelsen K, Chén O Y, Koch P, Mertins A, De Vos M (2022). Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification. IEEE Transactions on Biomedical Engineering, 69(8): 2456–2467

Prabhakararao E, Dandapat S (2022). Multi-scale convolutional neural network ensemble for multi-class arrhythmia classification. IEEE Journal of Biomedical and Health Informatics, 26(8): 3802–3812

Raj D, Garcia-Perera L P, Huang Z, Watanabe S, Povey D, Stolcke A, Khudanpur S (2021). Dover-lap: A method for combining overlap-aware diarization outputs. 2021 IEEE Spoken Language Technology Workshop (SLT), IEEE. 881–888

Ramachandran A, Karuppiah A (2021). A survey on recent advances in machine learning based sleep apnea detection systems. Healthcare, MDPI. 914

Reznik L, Von Pless G, Al Karim T (2011). Distributed neural networks for signal change detection: On the way to cognition in sensor networks. IEEE Sensors Journal, 11(3): 791–798

Roggen D, Calatroni A, Rossi M, Holleczek T, Förster K, Tröster G, Lukowicz P, Bannach D, Pirkl G, Ferscha A (2010). Collecting complex activity datasets in highly rich networked sensor environments. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS), IEEE. 233–240

Ruanaidh J O, Fitzgerald W J, Pope K J (1994). Recursive Bayesian location of a discontinuity in time series. Proceedings of ICASSP'94. In: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE. IV/513-IV/516 vol. 514

Saatçi Y, Turner R D, Rasmussen C E (2010). Gaussian process change point models. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). 927–934

San-Segundo R, Gil-Martín M, D'Haro-Enríquez L F, Pardo J M (2019). Classification of epileptic EEG recordings using signal transforms and convolutional neural networks. Computers in Biology and Medicine, 109: 148–158

Shahnawazuddin S, Ahmad W, Adiga N, Kumar A (2020). In-domain and out-of-domain data augmentation to improve children's speaker verification system in limited data scenario. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. 7554–7558

Shaker A M, Tantawi M, Shedeed H A, Tolba M F (2020). Generalization of convolutional neural networks for ECG classification using generative adversarial networks. IEEE Access: Practical Innovations, Open Solutions, 8: 35592–35605

Shoeb A H (2009). Application of machine learning to epileptic seizure onset detection and treatment, Massachusetts Institute of Technology

Shoeibi A, Khodatars M, Ghassemi N, Jafari M, Moridian P, Alizadehsani R, Panahiazar M, Khozeimeh F, Zare A, Hosseini-Nejad H, Khosravi A, Atiya A F, Aminshahidi D, Hussain S, Rouhani M, Nahavandi S, Acharya U R (2021). Epileptic seizures detection using deep learning techniques: A review. International Journal of Environmental Research and Public Health, 18(11): 5780

Shou Z, Wang D, Chang S F (2016). Temporal action localization in untrimmed videos via multi-stage cnns. Proceedings of the IEEE conference on computer vision and pattern recognition. 1049–1058

Singh P, Sharma A (2022). Interpretation and classification of arrhythmia using deep convolutional network. IEEE Transactions on Instrumentation and Measurement, 71: 1–12

Snyder D, Garcia-Romero D, Sell G, Povey D, Khudanpur S (2018). X-vectors: Robust dnn embeddings for speaker recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE. 5329–5333

Soraya S I, Chuang S P, Tseng Y C, İk T U, Ching Y T (2019). A comprehensive multisensor dataset employing RGBD camera, inertial sensor and web camera. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE. 1–4

Stein S, McKenna S J (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 729–738

Stolcke A, Yoshioka T (2019). DOVER: A method for combining diarization outputs. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE. 757–763

Sulem D, Kenlay H, Cucuringu M, Dong X (2024). Graph similarity learning for change-point detection in dynamic networks. Machine Learning, 113(1): 1–44

Supratak A, Dong H, Wu C, Guo Y (2017). DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(11): 1998–2008

Thodoroff P, Pineau J, Lim A (2016). Learning robust features using deep learning for automatic seizure detection. Machine Learning for Healthcare Conference, PMLR. 178–190

Tian X, Deng Z, Ying W, Choi K S, Wu D, Qin B, Wang J, Shen H, Wang S (2019). Deep multi-view feature learning for EEG-based epileptic seizure detection. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(10): 1962–1972

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

Türk Ö, Özerdem M S (2019). Epilepsy detection by using scalogram based convolutional neural network from EEG signals. Brain Sciences, 9(5): 115

Vahdani E, Tian Y (2023). Deep learning-based action detection in untrimmed videos: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4): 4302–4320

Verma A, Janghel R R (2021). Epileptic seizure detection using deep recurrent neural networks in EEG signals. Advances in Biomedical Engineering and Technology: Select Proceedings of ICBEST 2018, Springer. 189–198

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P A, Bottou L (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11: 3371–3408

Wahyono T, Heryadi Y, Soeparno H, Abbas B S (2020). Anomaly detection in climate data using stacked and densely connected long short-term memory model. Journal of Computers, 31(4): 42–53

Wang J, Zhang Q, Zhao D, Chen Y (2019). Lane change decision-making through deep reinforcement learning with rule-based constraints. In: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE. 1–6

Wang S, Rohdin J, Plchot O, Burget L, Yu K, Černocký J (2020a). Investigation of specaugment for deep speaker embedding learning. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. 7139–7143

Wang W, Tran D, Feiszli M (2020b). What makes training multi-modal classification networks hard? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 12695–12705

Wang Z, Gao Z, Wang L, Li Z, Wu G (2020c). Boundary-aware cascade networks for temporal action segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, Springer. 34–51

Wang Z, Wang Y, Gao C, Wang F, Lin T, Chen Y (2022). An adaptive sliding window for anomaly detection of time series in wireless sensor networks. Wireless Networks, 28(1): 393–411

Wei Z, Wang B, Nguyen M H, Zhang J, Lin Z, Shen X, Mech R, Samaras D (2018). Sequence-to-segment networks for segment detection. Advances in Neural Information Processing Systems, 31

Wen Y, Wu J, Das D, Tseng T L B (2018). Degradation modeling and RUL prediction using Wiener process subject to multiple change points and unit heterogeneity. Reliability Engineering & System Safety, 176: 113–124

Wen Y, Wu J, Zhou Q, Tseng T L (2019). Multiple-change-point modeling and exact Bayesian inference of degradation signal for prognostic improvement. IEEE Transactions on Automation Science and Engineering, 16(2): 613–628

Wu H T, Zhou Z (2024). Frequency detection and change point estimation for time series of complex oscillation. Journal of the American Statistical Association, 119(547): 1945–1956

Wu J, Chen Y, Zhou S (2016). Online detection of steady-state operation using a multiple-change-point model and exact Bayesian inference. IIE Transactions, 48(7): 599–613

Xia K, Huang J, Wang H (2020). LSTM-CNN architecture for human activity recognition. IEEE Access: Practical Innovations, Open Solutions, 8: 56855–56866

Xiao Q, Lee K, Mokhtar S A, Ismail I, Pauzi A, Zhang Q, Lim P Y (2023). Deep learning-based ECG arrhythmia classification: A systematic review. Applied Sciences, 13(8): 4964

Xu R, Huang S, Song Z, Gao Y, Wu J (2023a). A deep mixed-effects modeling approach for real-time monitoring of metal additive manufacturing process. IISE Transactions, 1–15

Xu R, Wang C, Li Y, Wu J (2023b). Generalized Time Warping Invariant Dictionary Learning for Time Series Classification and Clustering. arXiv preprint arXiv:2306.17690

Xu R, Wu J, Yue X, Li Y (2023c). Online structural change-point detection of high-dimensional streaming data via dynamic sparse subspace learning. Technometrics, 65(1): 19–32

Yao R, Lin G, Shi Q, Ranasinghe D C (2018). Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. Pattern Recognition, 78: 252–266

Yuan Y, Jia K (2019a). FusionAtt: deep fusional attention networks for multi-channel biomedical signals. Sensors, 19(11): 2429

Yuan Y, Xun G, Jia K, Zhang A (2019b). A multi-view deep learning framework for EEG seizure detection. IEEE Journal of Biomedical and Health Informatics, 23(1): 83–94

Zhang A, Wang Q, Zhu Z, Paisley J, Wang C (2019). Fully supervised speaker diarization. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. 6301–6305

Zhang L, Chang X, Liu J, Luo M, Li Z, Yao L, Hauptmann A (2022). TN-ZSTAD: Transferable network for zero-shot temporal activity detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(3): 3848–3861

Zhang L, Chang X, Liu J, Luo M, Wang S, Ge Z, Hauptmann A (2020a). Zstad: Zero-shot temporal activity detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 879–888

Zhang M, Cui Z, Neumann M, Chen Y (2018). An end-to-end deep learning architecture for graph classification. In: Proceedings of the AAAI conference on artificial intelligence

Zhang R, Hao Y, Yu D, Chang W C, Lai G, Yang Y (2020b). Correlation-aware unsupervised change-point detection via graph neural networks. arXiv preprint arXiv:2004.11934

Zhu Y, Long Y, Guan Y, Newsam S, Shao L (2018). Towards universal representation for unseen action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 9436–9445