


Review

Concept Drift Adaptation Methods under the Deep Learning Framework: A Literature Review

Qiuyan Xiang , Lingling Zi *, Xin Cong and Yan Wang

College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China

* Correspondence: lingling19812004@126.com

Abstract: With the advent of the fourth industrial revolution, data-driven decision making has also become an integral part of decision making. At the same time, deep learning is one of the core technologies of the fourth industrial revolution that have become vital in decision making. However, in the era of epidemics and big data, the volume of data has increased dramatically while the sources have become progressively more complex, making data distribution highly susceptible to change. These situations can easily lead to concept drift, which directly affects the effectiveness of prediction models. How to cope with such complex situations and make timely and accurate decisions from multiple perspectives is a challenging research issue. To address this challenge, we summarize concept drift adaptation methods under the deep learning framework, which is beneficial to help decision makers make better decisions and analyze the causes of concept drift. First, we provide an overall introduction to concept drift, including the definition, causes, types, and process of concept drift adaptation methods under the deep learning framework. Second, we summarize concept drift adaptation methods in terms of discriminative learning, generative learning, hybrid learning, and others. For each aspect, we elaborate on the update modes, detection modes, and adaptation drift types of concept drift adaptation methods. In addition, we briefly describe the characteristics and application fields of deep learning algorithms using concept drift adaptation methods. Finally, we summarize common datasets and evaluation metrics and present future directions.

Keywords: deep learning; time-series analysis; concept drift; data-driven decision making; discriminative learning; generative learning; hybrid learning; deep reinforcement learning; deep transfer learning



Citation: Xiang, Q.; Zi, L.; Cong, X.; Wang, Y. Concept Drift Adaptation Methods under the Deep Learning Framework: A Literature Review. *Appl. Sci.* **2023**, *13*, 6515. <https://doi.org/10.3390/app13116515>

Academic Editors: Paolo Renna, Asad Masood Khattak, Rahman Ali and Farkhund Iqbal

Received: 23 February 2023

Revised: 29 April 2023

Accepted: 24 May 2023

Published: 26 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The 2019 outbreak of coronavirus disease (COVID-19) has distinct effects on people's health and quality of life, and there is great uncertainty regarding the outbreak's evolution, duration, and scope in the future. So, in this era of the epidemic and big data, decision makers face a series of problems, such as extensive databases, rapid growth, diversified data sources, and rapid changes in data distribution. At present, deep learning technologies can solve part of the problems and provide part of the guidance to decision makers [1]. However, it does not adapt well to the changing environment. Once the environment changes, the new data does not match the distribution of the old data. With concept drift occurring [2], the deep learning model will become obsolete and invalid. How to deal with this complex situation and make timely and accurate decisions from multiple perspectives is a challenging research problem. Concept drift adaptation methods offer the possibility of solving the problem [3], helping decision makers to find the optimal or most satisfactory solution in this dynamic and complex situation. These methods can continuously capture the potential danger of events by analyzing the data stream, deal with distribution changes in the data stream on time, and help decision makers update existing decision results to prevent losses due to decision making.

Concept drift adaptation methods can be applied in various fields. By continuously upgrading the learning model to ensure its correctness and timeliness, they can provide accurate prediction information for decision makers. At the same time, they can perform active detection in combination with detection algorithms, which provides decision makers with knowledge of dynamic concepts [4], and analyze the concept drift that is occurring. For example, in the information security field, improved long short-term memory (I-LSTM) [5] can effectively detect network anomalies, analyze the causes, and strengthen protection. In the financial investment field, DeepBreath [6] can effectively minimize investment risk, and it has been experimentally proven to obtain better investment returns than current expert investment strategies. In addition, in the field of customer behavior analysis and air quality detection, ConceptExplorer [7], which combined concept drift with visualization, can help decision makers to conduct intuitive exploration and provide a basis for experts to make subsequent decisions. Therefore, concept drift adaptation methods can effectively help decision makers determine the main influencing factors, update their understanding of potential relationships, understand the nature of the data, find potential dangers, and make correct decisions.

Due to the important role of concept drift adaptation methods in applications, many researchers have reviewed them from different perspectives. For example, within nearly 10 years, Hoens et al. [8] reviewed concept drift detection methods and summarized the problem of class imbalance in the data stream. YM et al. [9] reviewed the classification problem from the perspective of a single model and an ensemble model. Gama et al. [4] reviewed concept drift adaptation methods from four aspects: memory usage, concept drift detection, model update method, and model evaluation index. Ditzler et al. [10] reviewed the existing concept drift detection methods from two aspects: active mode and passive mode. Webb et al. [11] mainly involved various types, such as drift target, drift transition mode, and drift amount, and explained their specific definitions. Krawczyk et al. [12] reviewed ensemble learning in data stream classification and regression tasks. Kamassi et al. [13] summarized concept drift adaptation methods from the aspects of data processing, adaptation mode, and the number of classifiers. Lu et al. [14] comprehensively summarized when drift occurs, to what extent, and how long it lasts, as well as the category imbalance in the classification of concept drift data streams, big data mining, and data-driven decision support systems. Ghani et al. [15] mainly introduced conceptual drift detection methods for untagged data streams. Lima et al. [16] summarized and classified concept drift adaptation methods based on different machine learning models under the regression task. Bayram et al. [17] classified and reviewed the existing detection algorithms of concept drift adaptation methods. However, to the best of our knowledge, there is no detailed introduction and summary of concept drift adaptation methods under the deep learning framework. Therefore, this paper explores the important role of concept drift adaptation methods under the deep learning framework.

The main contributions of this paper can be summarized as follows:

- (1) We review concept drift adaptation methods under the deep learning framework from four aspects—discriminative learning, generative learning, hybrid learning, and relevant others—so as to fill the gap in this area of investigation in previous work.
- (2) We reveal the general operation process of concept drift adaptive methods under deep learning frameworks and explain concept drift detection modes and update modes in detail.
- (3) We summarize the representative algorithms of each subcategory, common datasets, evaluation metrics, their application areas, and limitations.
- (4) We analyze and discuss the current problems of concept drift adaptation methods and point out the future direction.

The rest of the paper is structured as follows. Section 2 provides the definition, causes, and types of concept drift and introduces the process of concept drift adaptation methods under the deep learning framework. Section 3 classifies concept drift adaptation methods based on deep learning and reviews the existing methods in the literature. Section 4

summarizes the common datasets and evaluation metrics. Section 5 provides future research, and Section 6 concludes this paper.

2. Overview of Concept Drift

In this section, we introduce the definition and causes of concept drift, different types of concept drift, and the process of concept drift adaptation methods. Concept drift was first proposed by Schlemmer et al. [2] in 1986 and mainly refers to the fact that the underlying data stream distribution changes over time [18,19].

2.1. The Definition of Concept Drift

Assuming that P_{t_0} represents the joint probability distribution between the input variable x and the target variable y at time t_0 and P_{t_1} represents the joint probability distribution between the x and y at t_1 , then concept drift will occur if Equation (1) holds when t_0 turns to t_1 .

$$\exists x : P_{t_0}(x, y) \neq P_{t_1}(x, y) \tag{1}$$

At this time, the underlying data distribution no longer conforms to concept C_1 , and a new concept C_2 is generated. Due to the characteristics of joint probability $P_t(x, y) = P_t(x)P_t(y|x)$ if Equation (2) is satisfied when t_0 turns to t_1 , concept drift will also occur.

$$\exists x : P_{t_0}(x)P_{t_0}(y|x) \neq P_{t_1}(x)P_{t_1}(y|x) \tag{2}$$

Changes in both $P_t(x)$ and $P_t(y|x)$ can lead to concept drift.

2.2. The Causes of Concept Drift

According to the definition of concept drift and the characteristics of joint probability, it can have the following three causes:

- (1) Virtual concept drift. When the probability of x changes, but the probability of y under the condition of x does not change, i.e., $P_{t_0}(x) \neq P_{t_1}(x)$ and $P_{t_0}(y|x) = P_{t_1}(y|x)$. This case belongs to virtual concept drift, which does not affect its decision boundary and only changes the feature space.
- (2) Real concept drift. When the probability of y under the condition of x changes, the probability of x remains the same, i.e., $P_{t_0}(y|x) \neq P_{t_1}(y|x)$ and $P_{t_0}(x) = P_{t_1}(x)$. This case has a direct impact on the prediction model and is a real concept drift, which not only changes the feature space but also changes its decision-making boundary.
- (3) Hybrid concept drift. In an open environment, both real concept drift and virtual concept drift can exist in the data stream at the same time, i.e., $P_{t_0}(x) \neq P_{t_1}(x)$, $P_{t_0}(y|x) \neq P_{t_1}(y|x)$. This is a mixed concept drift, which is most common.

It is worth noting that according to the Bayesian decision theory [20], we obtain Equation (3):

$$P(y|x) = \frac{P(y) * P(x|y)}{P(x)} \tag{3}$$

It can be seen that $P_t(y)$ and $P_t(x|y)$ also affect $P_t(y|x)$, thus indirectly causing a real concept drift. The specific manifestations of the concept drift due to different causes are shown in Figure 1, in which (X_1, X_2) represents the two-digit feature space and y represents its category label.

For example, in stock trading, users can be divided into profitable and non-profitable stocks according to profitability. When a user considers purchasing stocks, a change in the channel of purchase or a small change in the number of purchases does not affect the trend of the stock. However, if affected by an outbreak, the trend of stocks may change, thus directly affecting stock returns. This situation belongs to the real concept drift, so users need to reconsider and make decisions. In real life, virtual drift tends to have less impact on the outcome of a decision. There will be no loss to decision makers. However, real concept drift tends to have a direct impact on decision outcomes due to changes in its

data relationships. It requires decision makers to discover in time and re-make decisions to avoid losses.

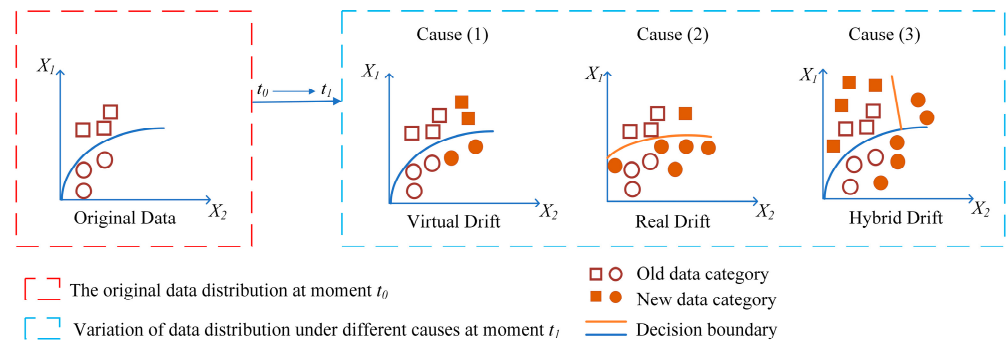


Figure 1. The causes of concept drift.

2.3. The Types of Concept Drift

The changes in concept may manifest in different forms over time. At present, the most popular types of concept drift can be divided into abrupt drift, incremental drift, gradual drift, and recurring drift [14,16,17].

Abrupt drift refers to the rapid change of concept C_1 into concept C_2 in a short period of time, and if an earthquake suddenly occurs in a certain place, its economic model changes instantaneously, as shown in Figure 2a. Incremental drift refers to the slow transformation of concept C_1 into concept C_2 in a continuous manner, as the economy gradually recovers after an earthquake, as shown in Figure 2b. Gradual drift refers to a short period of time: C_1 and C_2 repeatedly switch and eventually stabilize at C_2 , as the equipment ages, occasionally fails, and finally stops working, as shown in Figure 2c. Recurring drift refers to the fact that over time, the previous concept will reappear after a period of time; for example, the sales of down jackets meet concept C_1 in the winter, start to enter the off-season after the end of the winter, their sales will meet concept C_2 , and then the next winter concept C_1 will reappear, as shown in Figure 2d. In addition, the speed of recurring drift can be abrupt, gradual, or incremental. It can also be periodic or irregular.

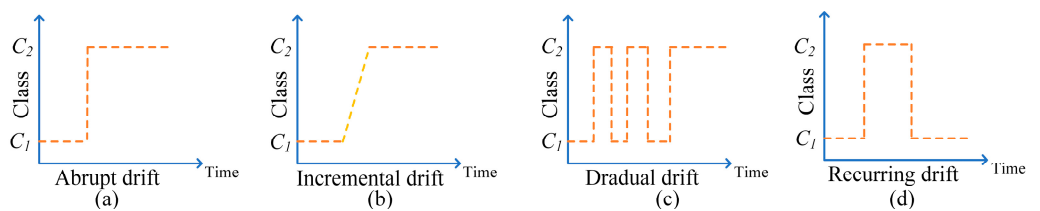


Figure 2. The types of concept drift.

In the academic research of concept drift, the types are different according to the classification criteria. However, it is common to divide the types of concepts according to their transformations, and this criteria manifestation is more intuitive. In related studies, different methods adapt to solving different types of concept drift. For example, the drift detection method (DDM) algorithm [21] is more suitable for abrupt drift. In addition to adapting the four common types above, there are some methods for distinguishing real drift and virtual drift, avoiding mixing with virtual concept drift or outliers and noise. For example, the RRBM-DD [22] considers explicitly how to identify the drift of the real concept. Although concept drift adaptation methods cannot solve all types of concept drift at one time, they can still solve multiple concept drifts, which belongs to a one-to-many relationship.

In recent years, there have also been many excellent concept drift detection algorithms to detect multiple concept drifts, for example, based on sliding-window algorithms, OCDD [23], CDT_MSW [24], and KSWIN [25]. OCDD mainly has two sliding windows

to store new and old data, and the percentage of outliers detected by the classifier in the sliding window is used to send a drift signal, which is more suitable for detecting abrupt and incremental drift, but hyperparameter settings are required. CDT_MSW also has two windows, the difference being that it can identify the position and length of concept drift, so as to accurately determine the type of concept drift. KSWIN detects concept drift by applying the “Kolmogorov–Smirnov test”. These algorithms are based on supervised learning. Unsupervised concept drift algorithms include LD3 [26], STUDD [27], and CDCMS [28]. LD3 introduces the concept of label-dependent ordering for concept drift detection in multi-label classification, which is more suitable for mutation and incremental drift. STUDD mainly creates an auxiliary model (students) to mimic the behavior of the main model (teacher), uses the teacher to predict new instances, and monitors the student’s imitation loss to detect concept drift. It is more suitable for abrupt, gradual, and incremental drift. CDCMS mainly uses novel clustering and diversity-based memory management strategies in model space strategies to deal with concept drift and has good effects in dealing with abrupt and recurring drift. Finally, it is worth mentioning that most of the concept drift detection algorithms either occupy more memory or have a slow detection speed. DMDDM [29] is based on the Page–Hinkley test, which effectively improves the detection speed of concept drift and overcomes the limitations of cost and execution time but is only suitable for abrupt drift. How to achieve a cost-saving detection algorithm that covers all drift types is also a major challenge. Therefore, we will also summarize the types of conceptual drift for each method.

2.4. The Process of Concept Drift Adaptation Methods under Deep Learning Framework

The general adaptation process of concept drift under the deep learning framework when dealing with unstable state data streams is shown in Figure 3. First, the data stream input (single input or batch input) is generally trained and learned by the deep learning model (single model or ensemble model) to obtain the basic prediction results. Next, if concept drift occurs during this process, a concept drift adaptation method will be triggered to update the deep learning model to accommodate concept drift and maintain the model [16,19]. The concept drift adaptation method can be divided into two parts: concept drift detection and model update. Among them, concept drift detection contains both active and passive modes, and model updates can be divided into structure updates and parameter updates.

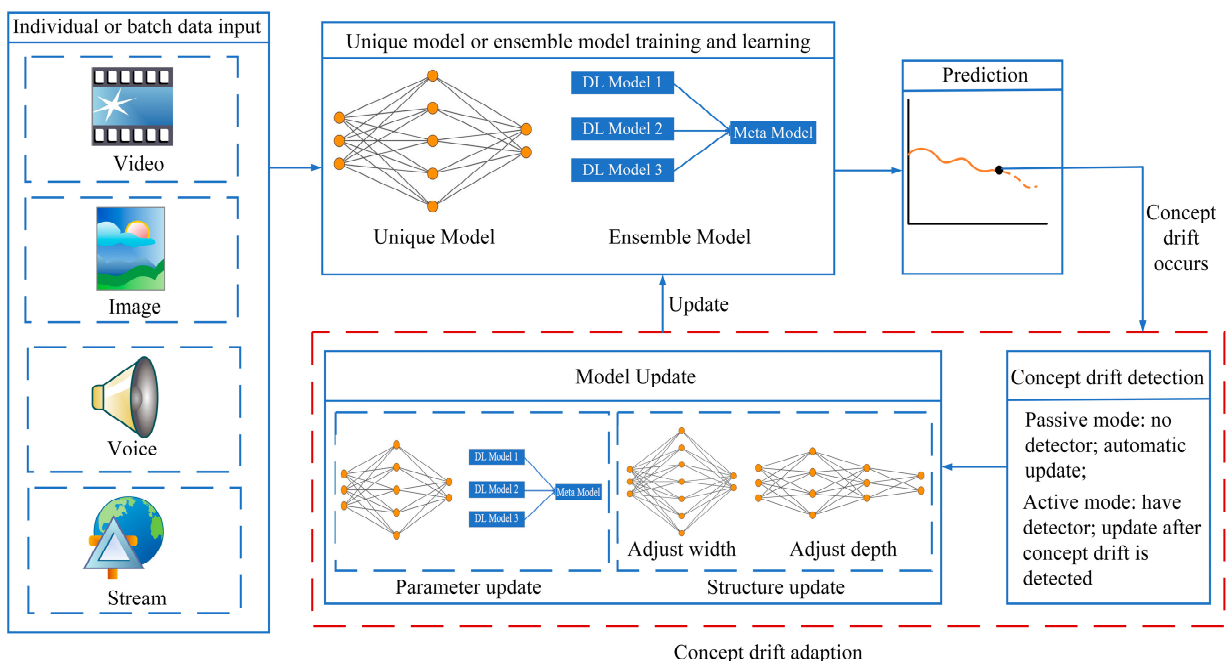


Figure 3. The general process of concept drift adaptation methods under deep learning framework.

Active modes mean that the learning process of a deep learning model contains a concept drift detection algorithm. When concept drift is detected, the concept drift adaptation method will be triggered to update the model. Passive mode means that the method continuously adjusts its model as data are continuously input during the learning process. Instead of using a drift detection algorithm, it uses a concept drift adaptation method to passively update the model continuously. After triggering the concept drift application mechanism, the deep learning model is generally updated to adapt to the concept drift through a model parameter update or a structure update [30]. Model parameter updates can be divided into full parameter updates and partial parameter updates. In particular, parameter updates also include parameter updates between ensemble models. Here, parameter updates are also weight updates. In addition, model structure updates can generally be performed by adjusting the width and depth of the network.

3. Concept Drift Adaptation Methods under Deep Learning

In this section, we will summarize concept drift adaptation methods according to the classification of deep learning [31], including discriminative learning, generative learning, hybrid learning, and others, as shown in Figure 4. For each part, we will explain the update modes, types of drift adapted, and detection modes. In addition, we will also introduce the characteristics and application fields of deep learning techniques using concept drift adaptation methods.

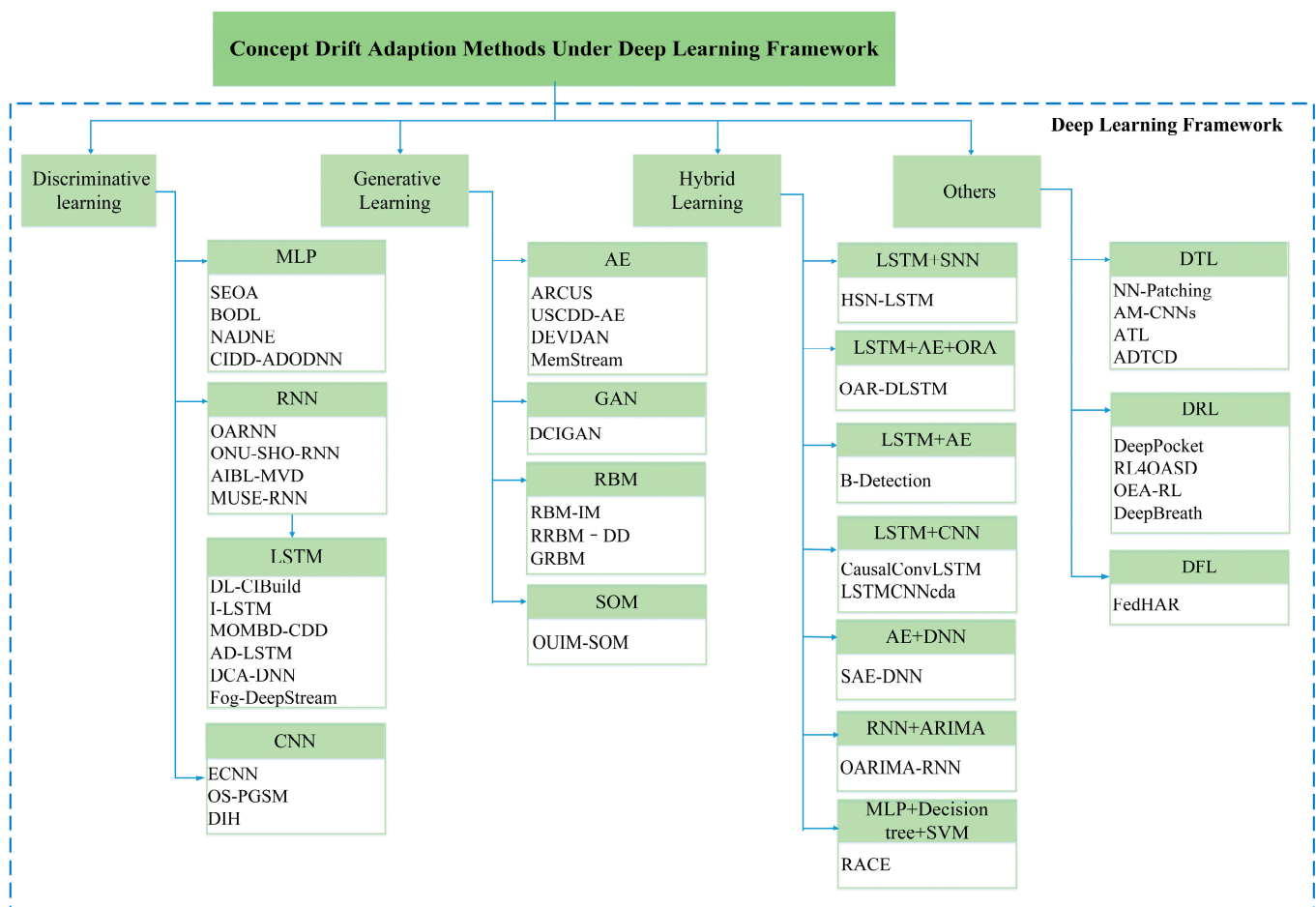


Figure 4. A taxonomy of concept drift adaption under deep learning framework.

3.1. Concept Drift Adaptation Methods Based on Discriminant Learning

This type of deep learning technique is used in supervised or classification applications by describing the posterior distributions of conditioned visible data. A discriminative

model is a model that learns the relationship between input data and output labels, and it predicts output labels by learning the characteristics of the input data. In classification problems, the main purpose is to assign each input vector a to label b . Discriminant models attempt to directly learn the function $f(a)$ that maps input vectors to labels. The classifier first learns the posterior class probability $P(b = k | a)$ from the training data and assigns a new sample a to the class with the highest posterior probability based on these probabilities, where k stands for class. The general process of the discriminant concept drift adaption method is shown in Figure 5, while the two methods of active detection and parameter update mode account for a relatively large proportion. Discriminant learning mainly includes multilayer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their variants.

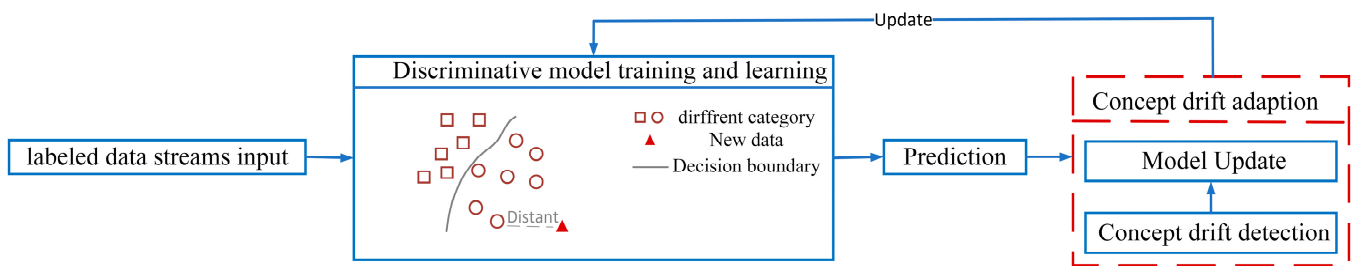


Figure 5. The general process of concept drift adaptation methods based on discriminant learning.

- MLP-based concept drift adaptation methods

MLP is a discriminant learning model widely adopted in decision making [32], which is often used in combination with concept drift adaptation methods to solve classification problems for unstable streaming data. However, concept drift adaptation methods are computationally expensive and converge slowly each time the model is updated due to the hyperparameter problem. Typical algorithms, such as selective ensemble-based online adaptive deep neural networks (SEOAs), bilevel online deep learning (BODL), neural networks with dynamically evolved capacity (NADINEs), and Adadelta optimizer-based deep neural networks with concept drift detection (CIDD-ADODNNs), are elaborated as follows.

SEOA [33] uses a deep learning model with L network layer MLPs to form L basic classifiers. It then dynamically adjusts the parameters of each basic classifier to handle concept drift and regularly selects base classifiers with different convergence and fitting abilities. It enhances the adaptability and generalization ability of the model to data distribution, which is more suitable for gradual, incremental, and recurring drift, although less suitable for dealing with high-dimensional non-linear problems. BODL [34] mainly uses the MLP model for classification prediction and detects concept drift based on the classifier's error rate. When concept drift is detected, the model's parameters are updated through a bilevel optimization strategy and the exponential gradient descent method to adapt to the abrupt concept drift, but its limitation is that the added classes cannot be identified online. In contrast, the convergence speed of the algorithm for model structure update can be slower. NADINE [35] uses a drift detection mechanism to detect concept drift actively. The drift detection mechanism mainly adds an adaptive windowing strategy to the prominent Hoeffding's bound detection algorithm. When the drift signal is detected, its network structure will be updated to adapt to concept drift, which mainly adjusts its network structure through the hidden unit growing strategy and hidden unit pruning strategy. The main advantage of NADINE over other algorithms is its elastic structure and online learning trait, but the training time of the model is relatively slow. It can be applied to classification and regression problems. Additionally, scholars have researched it for certain special data. CIDD-ADODNN [36] is adopted for the classification model of highly unbalanced data flow, which mainly uses an adaptive sliding window (ADWIN) drift detection algorithm to actively detect concept drift and then updates the network

parameters through the Adadelta optimizer, so as to adapt to the abrupt and gradual drift. This algorithm effectively improves the classification performance of highly unbalanced data streams, although its feature selection needs to be optimized.

- RNN-based concept drift adaptation methods

Compared to other neural networks, RNN has certain advantages in processing sequence data because it has at least one feedback connection [37]. To some extent, it can alleviate the problem of concept drift. However, its capacity is also limited, especially when it comes to processing long data. It is mainly used in the fields of electricity loading, weather forecasting, and anomaly detection. Typical algorithms are the online adaptive recurrent neural network (OARNN), ONU-SHO-based RNN (ONU-SHO-RNN), adaptive behavioral-based incremental batch learning malware variant detection model (AIBL-MVD), and multilayer self-evolving recurrent neural network (MUSE-RNN).

OARNN [38] mainly uses the RNN model to capture the temporal correlation and track its performance. When the performance deteriorates, the tree-structured Parzen estimator (TPE) will be used to optimize the hyperparameters of the model online. Then, the weights of the RNN model are completely updated and relearned from new data to accommodate concept drift over short periods of time. It is mainly used for energy and electricity load forecasting, although it requires a large amount of data for training and learning. In addition, Jagait et al. [39] proposed an online ARIMA-RNN integration based on OARNN, which belongs to hybrid learning. It will be further introduced later. ONU-SHO-RNN [40] determines whether to update the model by calculating its prediction accuracy and the concept drift detection of the RNN model on the data stream. It mainly uses the ONU-SHO algorithm to perform a complete parameter update and narrow the error between the target output and the measurement output. It is capable of fast convergence, adapting to incremental and gradual drift, although there are problems with update delays. In addition, AIBL-MVD [41] also adapts to incremental and gradual drift. It mainly uses the statistical process control (SPC) algorithm to actively detect the occurrence of concept drift and update all model weights through incremental learning. It is mainly used in the field of malware detection. In this process, the catastrophic forgetting problem is solved by mixing the new data with a subset of the old data. Its limitation is that labeled malware samples must be available right before updating the model. All of the above methods are based on parameter update mode. Subsequently, MUSE-RNN [42] mainly uses structural updates to update the models, and it actively detects concept drift through Hoeffding's bound detection algorithm, which is also a common method in concept drift detection algorithms. The model is updated by using growth and pruning hidden nodes and layers for the real-time classification of data streams, although it does not handle image streams.

- LSTM-based concept drift adaptation methods

Long short-term memory (LSTM) is a variant of RNN that solves problems such as vanishing gradients and is suitable for processing and forecasting important events with relatively long intervals and delays in time series [43]. LSTM-based concept drift adaptation methods are mainly used in the fields of anomaly detection, photovoltaic power generation prediction, and industrial prediction, and their typical algorithms include DL-CIBuild, I-LSTM, multi-objective metaheuristic optimization-based big data analytics with concept drift detection (MOMBD-CDD), adaptive LSTM (AD-LSTM), DCA-DNN, etc.

DL-CIBuild [44] is an algorithm based on the LSTM model to construct prediction models for continuous integration (CI) build outcome prediction. It uses the genetic algorithm (GA) to adjust the hyperparameters (including the number of hidden layers and neurons) of the LSTM model. In particular, it does not require a very large dataset size and has good robustness. However, the algorithm is relatively expensive in terms of labor and requires the construction of annotated datasets. I-LSTM [5] combines the idea of time factor with stratified sampling. Therefore, the newer the data, the higher the weight assigned to accommodate concept drift, but there are also problems with balancing old and new data. Overall, it improves multi-classification performance for anomaly detection, mainly for

IoT applications. MOMBD-CDD [45] mainly deals with high-dimensional streaming data. It mainly uses the Statistical Test of Equal Proportions method (STEPD) to detect concept drift and combines the glowworm swarm optimization (GSO) algorithm to update the bidirectional long short-term memory (Bi-LSTM) model by adjusting weights. However, it is more computationally intensive and takes up more resources. In this process, STEPD defines two windows, a recent window r and an overall window o . This is also common in deep-learning-based concept drift adaptation methods. It applies the statistical test of equal proportions to compare the accuracies between the two windows as shown in Equations (4) and (5):

$$T(v_o, v_r, n_o, n_r) = \frac{|v_o/n_o - v_r/n_r| - 0.5 \times (1/n_o + 1/n_r)}{\sqrt{\mu + (1 - \mu) \times (1/n_o + 1/n_r)}} \quad (4)$$

$$\mu = (v_o + v_r)/(n_o + n_r) \quad (5)$$

Its value is compared to the percentile of the standard normal distribution to obtain the observed significance level (p -value). p -value is equivalent to the chi-square test with Yates's continuity correction, in which v is the value of accurate predictions, and n is the number of samples for the window. The calculation formula for μ is shown in Equation (5). If p -value $< \alpha_d$, STEPD predicts a concept drift. If p -value $< \alpha_w$, STEPD predicts a warning that concept drift may occur. α_d is the concept drift significance level; α_w is the warning significance level. Fog-DeepStream [46] uses wavelet transform to reduce the dimensionality of the data and LSTM models to predict future behavior for data stream analysis on fog computing. It uses a drift detection algorithm to determine the occurrence of conceptual drift, and when a conceptual drift is detected, parameters are updated to accommodate the conceptual drift. The method tries three drift detection algorithms: cumulative sum (CUSUM), Page-Hinkley, and exponentially weighted moving average (EWMA). However, this algorithm also takes up a lot of memory.

The above algorithms are used in the Internet field. Next, we introduce algorithms in other fields. For example, AD-LSTM [47] is used for predicting photovoltaic power generation. It actively detects the occurrence of concept drift through the sliding window (SDWIN) algorithm and adopts the second stage of the two-phase adaptive learning strategy (TP-ALS) to fine-tune the prediction model. DCA-DNN [48] is mainly used for industrial prognosis and is based on the LSTM-FC model, which actively detects the occurrence of concept drift through the dendritic cell algorithm. It generates synthetic data using a kernel density estimator with drift-based bandwidth, which can be used to fine-tune the weights of the last layer to achieve faster adaptation and mitigate the problem of limited new samples. Both of the above algorithms suffer from model update delays, and their concept drift detection algorithms need to be optimized.

- CNN-based concept drift adaptation methods

CNN is a feed-forward neural network in which the connections between neurons in its convolutional layer are not fully connected, and the weights and biases of connections between some neurons in the same layer are shared [49,50]. So, the computational cost of this concept drift adaptation method is also relatively low. Typical algorithms, such as the evaluative convolutional neural network (ECNN) [51], mainly use re-weighting operation technology to dynamically update the model, so as to solve the concept drift problem in high-throughput data. ECNN overcame the "over-fitting" and "under-fitting" problems. ECNN is the first online deep learning technique to be introduced into marine data prediction research, although it is relatively computationally expensive. Online CNN-based model selection using performance gradient-based saliency maps (OS-PGSM) [52] is mainly applied to time-series prediction and uses Hoeffding's bound detection algorithm to actively detect the occurrence of concept drift. When concept drift occurs, the region of competence (ROC) of the model will be recalculated to update the weights. It has a low computational cost, using significance plots to provide an explanation for model selection,

but hyperparameter settings need to be optimized. Deep incremental hashing (DIH) [53] focuses on semantic image retrieval using a CNN model. The parameters of the CNN are updated using a point-by-point loss function guided by the similarity of the current data block keeping the target code. DIH mainly adapts to gradual, incremental drift. It also has certain limitations, such as not considering the semantic relationships between labels.

Table 1 summarizes the discriminant-learning-based concept drift adaptation methods. From this table, it can be seen that the MLP-based concept drift adaptation method focuses on the processing of streaming data samples to ensure the balance between old and new data and imbalanced data, thus improving the accuracy of prediction and reducing errors. However, it has certain limitations in dealing with high dimensionality, which is more suitable for dealing with gradual and abrupt concept drift. The RNN-based algorithm and its variants have good timeliness and can handle long-term serial data. However, it is necessary to overcome the problem of catastrophic forgetting, which is more sensitive to incremental and gradual concept drift. It is worth noting that the types of concept drift adaptation are rarely clearly specified in related studies based on LSTM and CNN. Further, most concept drift adaptation methods face the problem of slow convergence speed.

Table 1. Concept drift adaptation methods based on discriminant learning.

Types of Deep Learning	Algorithms	Concept Drift Adaptation Methods			Limitation
		Detection Modes	Update Modes	Adaptation Drift Types	
MLP	SEOA [33]	–	✓	A I G	Not suitable for high-dimensional non-linear problems
	BODL [34]	+	✓	A	New classes cannot be identified online
	NADINE [35]	+	×	A G R	Slow training time
	CIDD-ADODNN [36]	+	✓	A G	Feature selection to be optimized
RNN	OARNN [38]	–	✓	N	Requires large amounts of data to update the model
	ONU-SHO-RNN [40]	+	✓	I G	Model update delay
	AIBL-MVD [41]	+	✓	I G	Must have a marked malware sample before updating model
	MUSE-RNN [42]	+	×	A G R	Cannot handle image data streams
LSTM	DL-CIBuild [44]	–	×	N	High labor cost and need to build annotated datasets
	I-LSTM [5]	–	✓	G	Balance of old and new data
	MOMBD-CDD [45]	+	✓	N	High resource cost
	Fog-DeepStream [46]	+	✓		High memory consumption
	AD-LSTM [47]	+	✓	N	Model update latency exists
DCA-DNN [48]	+	✓	N		
CNN	ECNN [51]	–	✓	N	High computational cost
	OS-PGSM [52]	+	✓	N	Hyperparameter settings need to be optimized
	DIH [53]	–	✓	I G	No consideration of semantic relationships between labels
SNN	OeSNN [54]	+ –	×	G R	No consideration of a priori information such as speed and severity of drift

Note: + represents active mode, – represents passive mode, ✓ represents parameter update, × represents structural update, “A” represents abrupt drift, “I” represents incremental drift, “G” represents gradual drift, “R” represents recurring drift, and “N” means not mentioned in the reference.

In addition to the above types of mainstream algorithms, there are some other methods of discriminant learning. For example, the OeSNN-DRT algorithm based on a spike network [54] introduces two methods: active and passive adaptation methods. It uses the data reduction technique (DRT), a selective and generative data reduction technique, to

optimize the contents of the neuronal repository and update its structure. However, it does not take into account a priori information such as the speed and severity of the drift. Currently, there are few studies related to other types of concept drift adaptation methods compared to mainstream deep learning models, so they are not listed. However, it is a worthy direction for research.

3.2. Concept Drift Adaptation Methods Based on Generative Learning

Generative learning technologies are often used to describe higher-order correlation attributes or features for pattern analysis or synthesis, as well as joint statistical distributions of visible data and their related classes [55]. Most generative learning is unsupervised learning, but sometimes it can also be used for preprocessing in supervised learning, dimensionality reduction processing, etc. [56]. A generative model learns the data generation process, learns the probability distribution of input data, and generates new samples of data. More specifically, the generative model first estimates the conditional density of the classes $P(a | b = k)$ and the prior class probability $P(b = k)$ from the training data. They tried to understand how the data for each classification was generated. Bayes' theorem is then used to estimate the posterior class probability. Generative models can also learn the joint distribution of inputs and labels $P(a, b)$ and then normalize them to obtain posterior probabilities $P(b = k | a)$. The general process of the conceptual drift adaptive method based on generative learning is shown in Figure 6, while the general parameter update mode accounts for a large proportion, and the proportion of active detection and passive adaptation is comparable. Common deep neural network technologies for unsupervised or generative learning are generative adversarial networks (GANs), autoencoders (AEs), restricted Boltzmann machines (RBMs), self-organizing mapping (SOM), and deep belief networks (DBNs) and their variants.

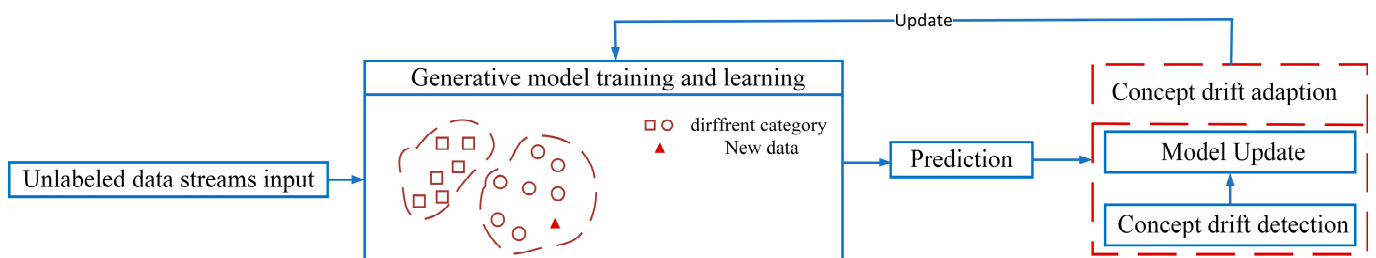


Figure 6. The general process of concept drift adaptation methods based on generative learning.

- AE-based concept drift adaptation methods

AE mainly consists of an encoder, a code, and a decoder [57]. It is combined with a concept drift adaptation method, which is mainly used for the anomaly detection of some high-dimensional data, such as the detection of the anomalous behavior of elderly people. Typical algorithms include the adaptive framework for online deep anomaly detection (ARCUS), unsupervised statistical concept drift detection (USCDD-AE), deep evolving denoising autoencoder (DEV DAN), and memory-based streaming anomaly detection (MemStream).

ARCUS [58] contains concept-driven inference and drift-aware model pool updates, where concept-driven inference focuses on evaluating the reliability of its models and giving evaluation scores when given a new data point. When a concept drift occurs, its evaluation score will drop to trigger a model pool update. Some models will then be removed and retrained to adapt to the occurrence of concept drift. In this process, the algorithm mainly uses the same structure of the AE model to form a model pool to perform anomaly detection of the data flow, which mainly has a large resource cost and cannot store the current batch of data where concept drift may occur. USCDD-AE [59] uses variational autoencoders to identify the anomalies of elderly people, which detects concept drift based

on data from families and the activity probability plot of the Kullback–Leibler divergence, as defined below.

$$D_{KL}(G \parallel Q) = \sum_{z \in Z} G(z) \log\left(\frac{G(z)}{Q(z)}\right) \quad (6)$$

where Z is the probability space, and G and Q are probability distributions defined over Z . Here G and Q are activity probability maps. Then, when concept drift occurs, the encoder will be updated to adapt to concept drift by backpropagating the reconstruction error. In this process, there are often difficulties with data collection and the possibility of false positives.

DEV DAN [60] is an incremental learning method that primarily uses the network significance formula to evaluate the predictive power of the model. Once the value in the capture formula rises, its hidden nodes are adjusted. USCDD-AE and DEV DAN are mainly based on the active concept drift adaptation method but ignore mutation oblivion when adding new layers. MemStream [61] is used for anomaly detection in multidimensional data and concept drift. It first uses a small portion of the training set and extracts features using the denoising autoencoder. Then, when a new sample arrives, the anomaly score is recalculated, and the weighting factor of AE is updated. If the anomaly score exceeds a user-set threshold, the memory is updated in a first in, first out (FIFO) manner, and the model is retrained to accommodate concept drift. This method effectively avoids noise disturbances and retrains quickly but with high resource overheads.

- GAN-based concept drift adaptation methods

GAN mainly consists of a generator and a discriminator. The former is used to create new data with similar characteristics to the original data, and the latter is used to determine the authenticity of the given data [62]. There are few examples of GAN combined with concept drift adaptation methods compared to other deep learning techniques, such as the distributed class-incremental learning method based on generative adversarial networks (DCIGAN). DCIGAN [63] uses a GAN generator to store information about past data and constantly updates GAN parameters with new data. Meanwhile, a generative fusion method (GF), which integrates multi-node local generators into a new global generator, is adopted. Particularly, a method for monitoring and evaluating GAN during continuous learning is presented, which explains the concept drift [64]. Its main purpose is to solve the problem of classifying data streams, but different hyperparameters need to be set in different environments.

- RBM-based concept drift adaptation methods

RBM is usually made up of visible and hidden nodes, each connected to every other node, which facilitates the understanding of some irregular datasets. Moreover, it is sensitive to the occurrence of concept drift because it is able to learn the probability distribution of the input [65]. RBM-I [66] and RRBM-DD [22] are two typical concept drift detection algorithms proposed by Korycki and Krawczyk, for multi-class imbalance and the presence of adversarial attack data streams, respectively. They both use gradient descent to update the weights in order to maintain the sensitivity of concept drift detection. RBM-IM is not suitable for small data streams and is prone to overfitting. RBM-DD has limitations in identifying adversarial concept drift in dynamic classes of unbalanced data streams. In addition, the Gaussian restricted Boltzmann machine (GRBM) algorithm [67] primarily uses the Kullback–Leibler divergence distance to determine whether a concept drift has occurred, thus enabling the adaptive adjustment of the sliding window and the division of the data stream. It reduces energy consumption and saves memory but only makes judgments on data from a single source and does not adaptively divide heterogeneous data.

- SOM-based concept drift adaptation methods

SOM is often applied to create low-dimensional (usually two-dimensional) representations of high-dimensional datasets, while maintaining the topology of the data [68]. The main benefit of using SOM is that it makes high-dimensional data easier to visualize and

analyze for understanding patterns. As in the case of GAN, there are few examples of SOM combined with concept drift adaptation methods. An online unsupervised incremental method based on self-organizing maps (OUIM-SOM) [69] is used for multi-label stream classification in infinite delay labeling scenarios. It adopts the online update of neuronal weight vectors and dataset label cardinality to accommodate abrupt and incremental concept drifts. However, its adaptive effect on conceptual drift is limited.

Table 2 summarizes the typical algorithms based on generative learning. Among them, AE-based algorithms are mainly used for anomaly detection, and depending on the characteristics of autoencoders, these algorithms use generally unsupervised or semi-supervised learning, which can enhance the flexibility of data flow methods in utilizing unlabeled samples. The remaining methods of combining generative learning models with concept drift algorithms, especially the concept drift adaptation methods involved in the deep belief network, have not been found, so they are not presented in this paper. However, there are some other generative learning models involved. A self-organizing incremental neural network (SOINN+) for unsupervised learning from noisy data streams [70] adapts to concept drift by adding or removing nodes, creating or deleting edges, or combining both. SOINN+ is robust to noise and is able to find topological representations that are consistent with the distribution of real data. It is worth noting that the Euclidean distance used in the node similarity metric is not suitable for high-dimensional data.

Table 2. Concept drift adaptation methods based on generative learning.

Types of Deep Learning	Algorithms	Concept Drift Adaptation Methods			Limitation
		Detection Modes	Update Modes	Adaptation Drift Types	
AE	ARCUS [58]	–	✓	A I G R	Cannot store data for the current batch where concept drift may occur
	DEV DAN [60]	+	×	A I G	Ignores mutation forgetting when adding new layers
	MemStream [61]	–	✓	N	High resource overhead
	USCDD-AE [59]	+	✓	A I G R	Difficult data collection and possible false positives
GAN	DCIGAN [63]	–	✓	N	Hyperparameter setting
RBM	RRBM-DD [22]	+	✓	A G R	Limitations in identifying adversarial conceptual drift in dynamic class imbalanced data streams
	RBM-IM [66]	+	✓	A I G	Not suitable for small data streams, prone to overfitting
	GRBM [67]	+	✓	N	Does not adaptively partition heterogeneous data
SOM	OUIM-SOM [69]	–	✓	A I	Limited adaptive effect on conceptual drift
SOINN	SOINN+ [70]	–	×	A I	Euclidean distance used in the node similarity measure is not suitable for high-dimensional data

Note: The symbol description is the same as in Table 1.

3.3. Concept Drift Adaptation Methods Based on Hybrid Learning

Hybrid deep learning models usually consist of multiple deep underlying learning models, either a free combination of discriminative or generative learning or discriminative/generative learning plus other models, such as CNN + LSTM, GAN + CNN, CNN + SVM, and other algorithms, as shown in Table 3.

The generative model and discriminant model have their own advantages. The generative learning model can learn from unlabeled data and can save labor costs. The discriminant learning model is better than the generative model in supervised tasks. Hybrid deep learning integrates discriminant or generative models according to the target task, and the framework for training deep generative models and discriminant models can enjoy

the advantages of both models to solve real-world problems. The general process of the concept drift adaptation method based on generative learning is shown in Figure 7, while the general parameter update mode accounts for a large proportion, and the proportion of active detection and passive adaptation is comparable.

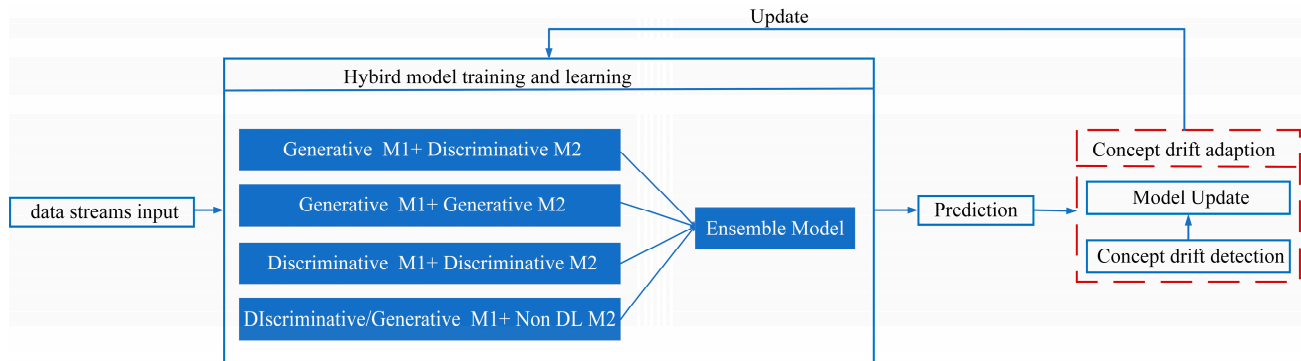


Figure 7. The general process of concept drift adaptation methods based on hybrid learning.

Typical algorithms combined with LSTM include HSN-LSTM, online autoregression with deep long short-term memory (OAR-DLSTM), CausalConvLSTM, and LSTMCNNcda. HSN-LSTM [71] is mainly used for multivariate time-series forecasting. It mainly embeds a novel adaptive and hybrid spiking (AHS) module into LSTM to keep the model capable of long-term prediction and alleviate its catastrophic forgetting problem. At the same time, in order to mitigate the impact of concept drift, it adopts the negative log-likelihood function in the fusion attention module to dynamically adjust the attention score and avoid noise interference. However, the resource costs are relatively high. OAR-DLSTM [72] combines a denoising autoencoder, an autoregressive model, and the deep long short-term memory (DLSTM) method, where the denoising encoder is mainly applied to feature extraction, and ORA and DLSTM are applied to target prediction. In the offline state, it divides the training data into data blocks and then pre-trains and re-trains the DLSTM model with the error rate predicted by ORA in each data block to obtain several independent sub-models. In the online state, the results of the two models are weighted using a maximum likelihood estimation to obtain the final time-series prediction output. When the dataset is too large, its performance degrades. B-Detection [73] is primarily used to detect runtime reliability anomalies in MEC services. It uses LSTM and AE models to capture the normal reliability data stream distribution characteristics. A weight-based reservoir sampling technique is then used to sample representative normal reliability data. Finally, the sampled data are used for detection model training, and the detection model is retrained to accommodate conceptual drift based on detection performance. However, the run time is relatively long.

A typical algorithm for the combination of LSTM and CNN is CausalConvLSTM [74], which utilizes CNN to extract spatial features efficiently and the LSTM model for prediction. It determines whether the model needs to be retrained based on the false-positive rate calculated from the rolling window and updates the network weights to accommodate concept drift by the backpropagation through time (BPTT) algorithm. CausalConvLSTM is primarily used for network intrusion detection but has a problem in that it is limited in the types of logs. Another example is LSTMCNNcda [75] for time-series forecasting, which focuses on actively detecting the occurrence of concept drift and updating the LSTMCNNnet model by an online parameter update when a concept drift is detected but with certain restrictions on the normalized time series and window size selection.

In addition, typical algorithms based on hybrid learning include the stacked autoencoder-deep neural network (SAE-DNN), OARIMA-RNN, and recurrent adaptive classifier ensemble (RACE). SAE-DNN [76] actively detects the occurrence of concept drift using the STEPD. If a concept drift occurs, the top level of SAE-DNN is extended by means of random vector function linking (RVFL). The parameters in the extension layer are dynamically assigned to new data through Lasso regularization and L2 regularization.

However, there is a certain amount of noise interference. Adaptive online ensemble learning with RNN and ARIMA (OARIMA-RNN) [39] uses RNN models to capture temporal dependencies and implement online learning modeling. Then, it dynamically adapts to concept drift by adding ARIMA to the set and RNN hyperparameters being optimized with each new batch. It has better accuracy than traditional offline models. However, there was no quantification of conceptual drift or performance during the drift. RACE [77] uses the concept of processing recycling, which uses an MLP, J48 decision tree, and support vector machine as basic learners to process training instances of time-series data. Then, the training instances are processed by the incremental learning algorithm, and the concept is used to detect the occurrence of concept drift. When concept drift occurs, it is updated and retrained. The algorithm requires a large amount of memory to run and slows down convergence as the size of the integration increases.

Table 3. Concept drift adaptation methods based on hybrid learning.

Types of Deep Learning	Algorithms	Concept Drift Adaptation Methods			Limitation
		Detection Modes	Update Modes	Adaptation Drift Types	
LSTM + SNN	HSN-LSTM [71]	–	✓	N	High resource overhead
LSTM + AE + ORA	OAR-DLSTM [72]	–	✓	I R	The dataset is too large, and its performance appears to degrade
LSTM + AE	B-Detection [73]	–	✓	A I G R	Long running time
LSTM + CNN	CausalConvLSTM [74]	–	✓	N	Limited log types for algorithm applications
	LSTMCNNcda [75]	+	✓	A G	Time-series data normalization issues, window size selection
AE + DNN	SAE-DNN [76]	+	×	A G	Noise interference
RNN + ARIMA	OARIMA-RNN [39]	–	✓	N	No quantification of conceptual drift or performance during drift
MLP + Decision tree + SVM	RACE [77]	+	✓	A G R	Requires large amounts of memory, increased integration size, slows convergence

Note: The symbol description is the same as in Table 1.

From the summary of typical algorithms in Table 3, it can be seen that “LSTM” + “other models” is a common hybrid approach, which is mainly applicable to long-term streaming data and can overcome the forgetting problem and improve the accuracy to a certain extent. In summary, for hybrid learning methods, multi-model integration is mainly tuned using dynamic weighting between models. So, it is essentially parameter updating, and there are also embedded model combinations that are mainly applied in process industries, such as power forecasting.

3.4. Other Concept Drift Adaptation Methods

The deep learning framework classification mentioned above is mainly divided based on the perspective of single-class models or hybrid multiple models. It is worth noting that there are cases where other deep learning technologies [31,78], such as deep reinforcement learning and deep federated learning, are used. For example, deep reinforcement learning was introduced as a combination of deep learning (DL) and reinforcement learning (RL) to better cope with the dynamic changes of unstable environments, leveraging the primary deep learning models to generate the target models we need [79,80]. The general process of this concept drift adaption method is shown in Figure 8, while the structure update mode is more common in deep transfer learning. Parameter updating is more common in deep reinforcement learning. However, relatively few research studies involving concept drift adaption methods are elaborated compared to other classes. The most popular are deep transfer learning (DTL) and deep reinforcement learning (DRL). Therefore, we mainly

introduce DTL and DRL. Table 4 summarizes concept drift adaptation methods based on DTL and DRL.

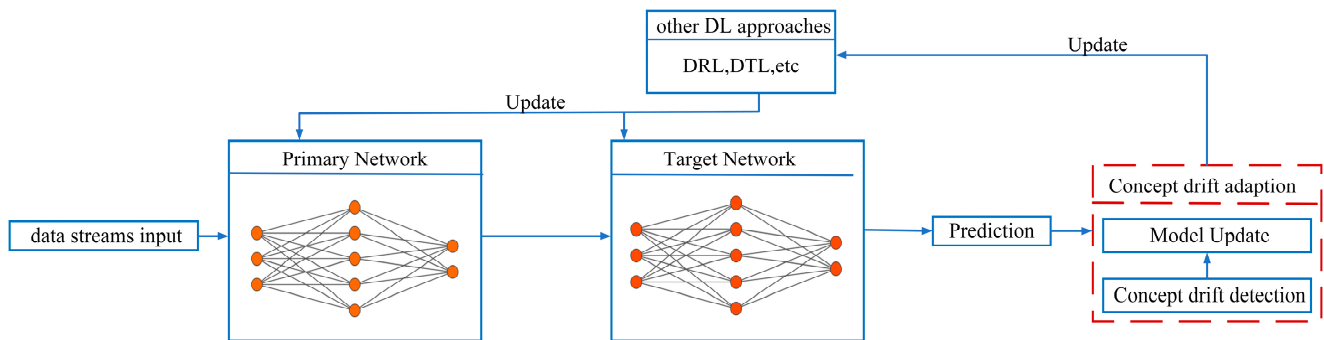


Figure 8. The general process of concept drift adaptation methods based on other deep learning.

- DTL-based concept drift adaptation methods

DTL mainly uses pre-training of deep learning models to obtain relevant knowledge. Then, by transferring the acquired knowledge to a new model, it can be adapted to a new task with minimal data and can save resources [81]. Currently, there are not many algorithms that involve concept drift. Typical methods are neural network patching (NN-Patching), adaptive mechanisms for learning CNNs (AM-CNNs), and autonomous transfer learning (ATL).

NN-Patching [82] is passively handled concept drift by an error estimator. It mainly constructs a discriminant classifier to identify the misclassified regions. Then, it trains a new classifier (called a patch network) on the misclassified data. The patch network uses the intermediate layer of the original neural network to extract features and representations that are critical to classification. This method keeps the original neural network quickly adaptable to concept drift, but its ability to handle concept drift is limited, and the hyperparameters need to be adjusted for the scene. AM-CNNs [83] uses the nonparametric CUSUM test to actively detect the occurrence of concept drift. It relies on a “transfer learning” paradigm that transfers the knowledge of the CNN running before the concept drift to the CNN running after the concept drift, but the resource overhead is relatively high. ATL [84] uses the autonomous Gaussian mixture model (AGMM) to automatically adjust the network width, which solves the concept drift problem. It is just a matter of readapting to a concept that has been there before when it reappears. An adaptive anomaly detection approach toward concept drift (ADTCD) [85] is an adaptive anomaly detection model based on knowledge distillation and DTL. It transfers knowledge from the AE-based teacher model to the student model and updates only the student model, which dynamically adjusts model weights to accommodate concept drift primarily through local inference on new samples. However, the algorithm also suffers from two limitations. Firstly, the industrial scenarios used for the experiments are relatively homogeneous, and secondly, little attention is paid to scarce anomaly data.

- DRL-based concept drift adaptation methods

DRL combines the perception ability of deep learning with the decision-making ability of reinforcement learning, which can be directly controlled based on the input information. Reinforcement learning defines the goal of optimization, and deep learning gives the mechanism by which it works (how to characterize problems and how to solve them) [86]. The algorithms using this concept drift adapting method are mainly applied in the fields of financial investment and anomaly detection. Typical algorithms include DeepPocket, RL4OASD, online ensemble aggregation using reinforcement learning (OEA-RL), and DeepBreath.

DeepPocket [87] is used in the field of financial investment. This algorithm mainly uses a restricted stacked autoencoder to extract features and uses two convolutional networks

to find the best portfolio through deep reinforcement learning. Then, it uses online training to dynamically update weights to accommodate concept drift, but it does not lend itself to a long-term investment strategy. RL4OASD [88] is mainly used for the detection of abnormal trajectories of vehicles. It includes two networks: one is responsible for learning the features of the road network and trajectory, and the other is responsible for detecting anomalous sub-traces based on the learned features. The two networks can be trained iteratively without labeled data, and they employ an online learning strategy; that is, they are trained with newly recorded trajectory data and continuously update their strategies based on current traffic conditions, but they have a longer training time. OEA-RL [89] mainly uses the deep reinforcement learning framework as a meta-learning method to learn linear weighted ensembles and actively detects the occurrence of concept drift through the Page–Hinkley (PH) test. Then, it adapts to concept drift by updating its parameters. Again, there is a certain delay in updating due to its active detection algorithm. DeepBreath [6] is mainly used for financial investment, which uses a limited superimposed autoencoder for dimensionality reduction and feature processing. Then, the SARSA algorithm and the online batch processing method are used to train CNN learning investment strategies from historical data, and after training the model, the weights are updated through the online learning scheme to adapt to the concept drift. The algorithm lacks, to some extent, the consideration of exogenous factors.

As can be seen in Table 4, for DRL and DTL, the update mode of DRL is mainly a parameter update. It interacts well with the environment to learn the sequence of its behavior. The update mode of DTL is mainly a structural update. DTL can effectively use a small amount of data to train neural networks. This characteristic can use structure updates to train better predictive models. In addition, they generally use a combination of online and offline approaches to adapt concept drift and support more complex predictions. In addition to DTL and DRL, two more popular deep learning methods, there are also concept drift adaption methods based on other deep learning technologies. Such as FedHAR [90], which is a smart human activity recognition (HAR) frame based on deep federated learning. FedHAR designs an unsupervised gradient aggregation strategy that can overcome the problem of concept drift and convergence instability in online learning, which is mainly used to summarize the gradients of all labeled clients and unlabeled clients in federated learning and then drive the parameter update of the server model by averaging the aggregate gradient to adapt to the concept drift.

Table 4. Other concept drift adaptation methods.

Types of Deep Learning	Algorithms	Concept Drift Adaptation Methods			Limitation
		Detection Modes	Update Modes	Adaptation Drift Types	
DTL	NN-Patching [82]	–	×	N	Need to adjust hyperparameters for the scene
	AM-CNNs [83]	+	×	N	High overhead
	ATL [84]	–	×	A I G	Forgetting problem
	ADTCD [85]	–	√	A I G R	Little attention is paid to scarce anomaly data
DRL	DeepPocket [87]	–	√	N	Not suitable for long-term investment strategies
	RL4OASD [88]	–	√	N	Long model training time
	OEA-RL [89]	+	√	N	Updating delay
	DeepBreath [6]	–	√	N	Lack of consideration for exogenous factors
DFL	FedHAR [90]	–	√	N	Scarcity of labels, with privacy

Note: The symbol description is the same as in Table 1.

3.5. Discussion

According to the summary of concept drift adaptation methods, we can see that the proportion of hybrid learning and discriminant learning is relatively large, and especially

discriminative learning is widely used. This phenomenon reflects the fact that having label information samples is beneficial for detecting changes in the distribution of data. In addition, parameter updates also account for a large part. Compared with structural updates, parameter updates reduce the convergence time and adapt well to abrupt concept drift. Secondly, in the algorithms of discriminant learning and generative learning, its active detection also accounts for a considerable part, and it is mainly conducive to explaining the occurrence of concept drift and reducing the computing resources of training, but to a certain extent, it requires additional memory and CPU storage. From this paper, it can be seen that dealing with concept drift, reducing the amount of computation, saving resources, and speeding up convergence are our main challenges at present.

In addition, according to the above summary of drift adaptation types, we can find that there are usually more adaptation methods for abrupt, incremental, and gradual drift types. Relatively speaking, abrupt drift occurs most frequently, and its drift speed occurs the fastest, so most detection methods can be sensitive to detection, but there will also be problems such as update delay and high computational complexity. In contrast, recurring drift occurs the least often. In the case of recurring drift, previously learned models may become relevant again in the future. Online deep learning algorithms may have to relearn previous concepts. This process has a high computational burden because it means tuning or training a new model from scratch. This is also one of the main challenges at this time. Finally, it should be added that in addition to being based on deep learning algorithms, extreme learning machines are also models based on neural networks. In recent years, the use of the concept drift algorithm of ELM has also increased, and the main algorithms include Meta-RKOS-ELM [91], SSOE-FP-ELM [92], ONLAD [93], etc., which is also a worthy research direction.

4. Performance Evaluation of Concept Drift

In this section, we summarize the common datasets and evaluation metrics. The datasets are divided into real datasets and synthetic datasets. For the former, we present its sources, learning tasks, and properties. For the latter, we show the drift types and characteristics it contains. After that, we describe the evaluation metrics and their meanings.

4.1. Datasets

Real datasets can effectively demonstrate the generality and applicability of the algorithm in the real world, for which the commonly used datasets are KDD CUP 1999, Electricity, Weather, Spam, and CoverType. The KDD CUP 1999 [94] is the dataset used in the KDD (knowledge discovery and data mining) competition. It is mainly used for network intrusion detection to distinguish between normal network connections and malicious network connections, including various attack data simulated in the military network environment. Electricity [95] is derived from the electricity market of New South Wales, Australia (1996–1998). It is mainly used to predict changes in electricity prices in the past 24 h, including the weather, user demand, supply conditions, and seasons. Weather [96] contains daily weather measurement data for a certain area from 2006 to 2016, including temperature, humidity, wind direction, wind speed, visibility, atmospheric pressure, etc., for predicting rainfall. Spam [97] is mainly used to identify spam. CoverType [94] is derived from the forest cover of a certain area in the U.S. Forest Service system.

Synthetic datasets can evaluate the performance of the algorithms under different concept drift situations and contain various types of concept drift. For detecting abrupt concept drift, R.MNIST [98], P.MNIST [60], and SEA [99] can be used. SEA contains three features and two classes in each sample. R.MNIST and P.MNIST are generated from the MNIST dataset containing 784 features and 10 classes. It is worth noting that P.MNIST also detects recurring drift. For detecting gradual concept drift, Circles [100], Hyperplane [101], and LED [94] can be used. Circles contains two features and two classes in each sample. Then, Hyperplane also detects incremental drift, containing 10 features and 2 classes

in each sample. LED also detects abrupt drift, containing 24 features and 10 classes in each sample.

In addition to the above commonly used datasets, there are some special datasets for deep learning frameworks, such as the Vxheaven dataset [102] commonly used in previous malware analysis studies, consisting of Windows binaries belonging to malware and benign portable executables, containing different types of malware families, such as trojans, ransomware, and viruses. HAR-UCI [103] was made from recordings of 30 subjects performing activities of daily living. The STL-10 dataset [104] is an image recognition dataset for the development of unsupervised feature learning, deep learning, and self-learning algorithms. The Cat-Dog dataset [105] contains two classes, cats and dogs, with 12,500 images. The CIFAR100 dataset [106] is utilized to simulate the distribution drifting situation. It has 60,000 $32 \times 32 \times 3$ RGB images. Finally, some researchers have used their own collected data, as well as datasets from the application domain. For example, the I-LSTM and ECNN algorithms are collected data, and CausalLSTM uses the HDF5 dataset [107] and the Cybersecurity's Intrusion Detection Evaluation dataset [108].

4.2. The Evaluation Metrics

For algorithms based on discriminative learning and hybrid learning, accuracy recall, precision, F_1 -score, Matthews' correlation coefficient (MCC), and Cohen's kappa k are mainly used for classification problems, and mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) are mainly used for regression problems. For algorithms based on generative learning and others, the number of hidden nodes per time step (HN), the number of hidden layers per time step (HL), parameter count (PC), and execution time (ET) are mainly used. In particular, they are unique evaluation metrics under the framework of deep learning. It is worth noting that the MCC and Cohen's kappa k evaluation metrics are mainly used for unbalanced data. The definition of MCC is shown in Equation (7).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (7)$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. The definition of Cohen's kappa is shown in Equation (8).

$$k = \frac{P_0 - P_e}{1 - P_e} \quad (8)$$

where P_0 and P_e are the success rate of the actual and random predictors.

In addition to the above basic evaluation metrics, some researchers also use some evaluation metrics for the application field of algorithms. For example, the DeepPocket algorithm, which is mainly used in the field of financial investment, mainly uses maximum drawdown (MDD), Sharpe ratio (Sr), and conditional value at risk (CVaR) to evaluate its performance.

5. Future Directions

Based on the analysis and discussion of the above algorithms, we summarize the problems that need to be solved, which require further research in the future, as described below.

5.1. Full Coverage of Concept Drift Types

According to the above-mentioned algorithms, such as ONU-SHO-RNN, DEVDAN, etc., we can find that it is not possible to adapt to all concept drifts at once, and among the four types of concept drift, the best adaptability of the algorithm is to abrupt drift. There are also some algorithms, such as ECNN and HSN-LSTM, for which the dataset used in the experimental part does not indicate the type of drift included, and there are also no

experiments on the effectiveness of different types of concept drift. Therefore, it is necessary to improve the robustness and generalization of the methods to study concept drift.

5.2. Data Processing Problem

Data processing has been a big challenge in deep learning and concept drift adaptation methods [4,21]. Firstly, when inputting samples, we may face problems such as class-imbalance data, high-dimensional data, etc. For example, when performing online anomaly detection, most of the datasets are very unbalanced, and the abnormal data account for a very small part [109]. Secondly, when the model update is performed, we will face the problem of how to balance between new data and old data and the problem that the new data samples are not enough to support the update of the deep learning model after the concept drift occurs. These will lead to poor prediction, slow model convergence, delayed model update, and other consequences that are worthy of our consideration and research.

5.3. Multi-Model Integration Problem

Our review shows that online integration methods have been more popular in concept drift adaptation methods, such as OARIMA-RNN. Ensemble algorithms can effectively prevent overfitting and provide better prediction performance. However, their computational complexity is high, and they take up more resources, so how to optimize their performance is also a question worthy of deep consideration [110].

5.4. Visualization Problem of Concept Drift

At present, there is relatively little research on concept drift visualization. Classic visualizers have DriftVis [111], which can help decision makers identify and correct concept drift in data streams. In fact, for many related fields such as air quality monitoring, financial market analysis, etc. [7,112], explaining concept drift is conducive to helping decision makers comprehensively analyze problems and make correct decisions. Finally, it is worth mentioning the application of conceptual drift type visualization.

6. Conclusions

In recent years, deep learning has become one of the core technologies of the fourth industrial revolution. So, it has also become one of the indispensable tools to assist intelligent decision making. However, in the era of the epidemic and big data, data distribution in streaming data can change very easily, which is a phenomenon known as concept drift. Once concept drift occurs, even the best-trained deep learning models become obsolete, producing poor predictions. Therefore, this paper summarizes concept drift adaptation methods under the deep learning framework. Firstly, we explain the definition and causes of concept drift. Then, we introduce the types of concept drift and the general process of a concept drift adapting method under the deep learning framework. Next, we divide the deep learning model using the concept drift adaptation method from four aspects, including discriminant learning, generative learning, hybrid learning, and others. For each aspect, we introduce in detail the update modes, detection modes, and adaptation drift types of concept drift adaptation methods. Finally, we summarize common datasets and evaluation metrics for concept drift adaptation methods and point out future directions. We hope that this paper can provide some academic help to researchers.

Author Contributions: Conceptualization, Q.X. and L.Z.; methodology, Q.X. and X.C.; formal analysis, Q.X. and Y.W.; investigation, Q.X. and X.C.; writing—original draft preparation, Q.X.; writing—review and editing, Q.X., L.Z., X.C., and Y.W.; supervision, L.Z.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Program of Chongqing Education Science Planning Project (No. K22YE205098) and the Doctoral Research Foundation of Chongqing Normal University (No.21XLB030, No.21XLB029).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this review are from previously reported studies and datasets, which have been cited.

Acknowledgments: We acknowledge the financial support provided by the Key Program of Chongqing Education Science Planning Project (No. K22YE205098) and the Doctoral Research Foundation of Chongqing Normal University (No. 21XLB030, No. 21XLB029).

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

Abbreviations

The following abbreviations are used in this manuscript:

OCDD	one-class drift detector
CDT_MSW	concept drift type identification method based on multi-sliding windows
KSWIN	Kolmogorov–Smirnov test detector
LD3	label dependency drift detector
STUDD	student–teacher approach for unsupervised drift detection
CDCMS	concept drift handling based on clustering in the model space
DMDDM	diversity measure drift detection method
I-LSTM	improved long short-term memory
DDM	drift detection method
MLP	multilayer perceptron
CNN	convolutional neural network
RNN	recurrent neural network
DNN	deep neural network
SEOA	selective ensemble-based online adaptive deep neural network
BODL	bilevel online deep learning framework
NADINE	neural network with dynamically evolved capacity
CIDD-ADODNN	Adadelta optimizer-based deep neural networks with concept drift detection
ADWIN	adaptive sliding-window drift detection technology
OARNN	online adaptive recurrent neural network
TPE	tree-structured Parzen estimator
ONU-SHO	opposition-based novel updating spotted hyena optimization
ONU-SHO-RNN	ONU-SHO-based RNN
AIBL-MVD	adaptive behavioral-based incremental batch learning malware variant detection model
SPC	statistical process control
MUSE-RNN	multilayer self-evolving recurrent neural network
LSTM	long short-term memory
CI	continuous integration
GA	genetic algorithm
MOMBD-CDD	multi-objective metaheuristic optimization-based big data analytics with concept drift detection
STEPD	Statistical Test of Equal Proportions method
GSO	glowworm swarm optimization
Bi-LSTM	bidirectional long short-term memory
CUSUM	cumulative sum
EWMA	exponentially weighted moving average
AD-LSTM	adaptive LSTM framework
SDWIN	sliding-window algorithm
TP-ALS	two-phase adaptive learning strategy
ECNN	evolutionary convolutional neural network
OS-PGSM	online CNN-based model selection using performance gradient-based saliency maps
DIH	deep incremental hashing
ROC	region of competence

DRT	data reduction technique
ARCUS	adaptive framework for online deep anomaly detection under a complex evolving data stream
USCDD-AE	unsupervised statistical concept drift detection
DEVDAN	deep evolving denoising autoencoder
MemStream	memory-based streaming anomaly detection
FIFO	first in, first out
ADTCD	adaptive anomaly detection approach toward concept drift
GAN	generative adversarial network
AE	autoencoder
RBM	restricted Boltzmann machine
SOM	self-organizing mapping
DCIGAN	distributed class-incremental learning method based on generative adversarial networks
GF	generative fusion
GRBM	Gaussian restricted Boltzmann machine
OUIIM-SOM	online unsupervised incremental method based on self-organizing maps
SOINN+	self-organizing incremental neural network for unsupervised learning from noisy data streams
AHS	novel adaptive and hybrid spiking module
OAR-DLSTM	online autoregression with deep long short-term memory
HDL-OKW	hybrid deep learning classifier and optimized key windowing approach
BPTT	backpropagation through time
SAE-DNN	stacked autoencoder-deep neural network
RVFL	random vector function linking
SVM	support vector machines
OARIMA-RNN	adaptive online ensemble learning with recurrent neural network and ARIMA
RACE	recurrent adaptive classifier ensemble
DTL	deep transfer learning
NN-Patching	neural network patching
AM-CNNs	adaptive mechanism for learning CNNs
ATL	autonomous transfer learning
AGMM	autonomous Gaussian mixture model
DRL	deep reinforcement learning
OEA-RL	online ensemble aggregation using reinforcement learning
PH	Page–Hinkley
KDD	knowledge discovery and data mining
MCC	Matthews’ correlation coefficient
MAE	mean absolute error
MSE	mean squared error
RMSE	root mean squared error
HN	the number of hidden nodes per time step
HL	the number of hidden layers per time step
PC	parameter count
ET	execution time
MDD	maximum drawdown
Sr	Sharpe ratio
CVaR	conditional value at risk
FedHAR	federated learning human activity recognition frame

References

1. Shrestha, Y.R.; Krishna, V.; von Krogh, G. Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. *J. Bus. Res.* **2021**, *123*, 588–603. [[CrossRef](#)]
2. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [[CrossRef](#)]
3. Lu, J.; Liu, A.; Song, Y.; Zhang, G. Data-driven decision support under concept drift in streamed big data. *Complex Intell. Syst.* **2020**, *6*, 157–163. [[CrossRef](#)]
4. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 1–37. [[CrossRef](#)]

5. Xu, R.; Cheng, Y.; Liu, Z.; Xie, Y.; Yang, Y. Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services. *Futur. Gener. Comput. Syst.* **2020**, *112*, 228–242. [[CrossRef](#)]
6. Soleymani, F.; Paquet, E. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. *Expert Syst. Appl.* **2020**, *156*, 113456. [[CrossRef](#)]
7. Wang, X.; Chen, W.; Xia, J.; Chen, Z.; Xu, D.; Wu, X.; Xu, M.; Schreck, T. ConceptExplorer: Visual Analysis of Concept Drifts in Multi-source Time-series Data. In Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), Salt Lake City, UT, USA, 25–30 October 2020; pp. 1–11. [[CrossRef](#)]
8. Hoens, T.R.; Polikar, R.; Chawla, N.V. Learning from streaming data with concept drift and imbalance: An overview. *Prog. Artif. Intell.* **2012**, *1*, 89–101. [[CrossRef](#)]
9. Wen, Y.; Qiang, B.; Fan, Z. A survey of the classification of data streams with concept drift. *CAAI Trans. Intell. Syst.* **2013**, *8*, 95–104.
10. Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R. Learning in Nonstationary Environments: A Survey. *IEEE Comput. Intell. Mag.* **2015**, *10*, 12–25. [[CrossRef](#)]
11. Webb, G.I.; Hyde, R.; Cao, H.; Nguyen, H.L.; Petitjean, F. Characterizing concept drift. *Data Min. Knowl. Discov.* **2016**, *30*, 964–994. [[CrossRef](#)]
12. Krawczyk, B.; Minku, L.L.; Gama, J.; Stefanowski, J.; Woźniak, M. Ensemble learning for data stream analysis: A survey. *Inf. Fusion* **2017**, *37*, 132–156. [[CrossRef](#)]
13. Khamassi, I.; Sayed-Mouchaweh, M.; Hammami, M.; Ghédira, K. Discussion and review on evolving data streams and concept drift adapting. *Evol. Syst.* **2018**, *9*, 1–23. [[CrossRef](#)]
14. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. [[CrossRef](#)]
15. Ab Ghani, N.L.; Aziz, I.A.; Mehat, M. Concept Drift Detection on Unlabeled Data Streams: A Systematic Literature Review. In Proceedings of the IEEE Conference on Big Data and Analytics (ICBDA), Xiamen, China, 8–11 May 2020; pp. 61–65. [[CrossRef](#)]
16. Lima, M.; Neto, M.; Filho, T.S.; Fagundes, R.A.D.A. Learning Under Concept Drift for Regression—A Systematic Literature Review. *IEEE Access* **2022**, *10*, 45410–45429. [[CrossRef](#)]
17. Bayram, F.; Ahmed, B.S.; Kassler, A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowl. Based Syst.* **2022**, *245*, 108632. [[CrossRef](#)]
18. Desale, K.S.; Shinde, S.V. Addressing Concept Drifts Using Deep Learning for Heart Disease Prediction: A Review. In *Proceedings of Second Doctoral Symposium on Computational Intelligence*; Gupta, D., Khanna, A., Kansal, V., Fortino, G., Hassaniien, A.E., Eds.; Springer: Singapore, 2022; Volume 1374, pp. 157–167. [[CrossRef](#)]
19. Iwashita, A.S.; Papa, J.P. An Overview on Concept Drift Learning. *IEEE Access* **2018**, *7*, 1532–1547. [[CrossRef](#)]
20. R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, New York: John Wiley & Sons, 2001, pp. xx + 654, ISBN: 0-471-05669-3. *J. Classif.* **2007**, *24*, 305–307. [[CrossRef](#)]
21. Agrahari, S.; Singh, A.K. Concept Drift Detection in Data Stream Mining: A literature review. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *34*, 9523–9540. [[CrossRef](#)]
22. Korycki, Ł.; Krawczyk, B. Adversarial concept drift detection under poisoning attacks for robust data stream mining. *Mach. Learn.* **2022**, 1–36. [[CrossRef](#)]
23. Gözüaçık, Ö.; Can, F. Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artif. Intell. Rev.* **2021**, *54*, 3725–3747. [[CrossRef](#)]
24. Guo, H.; Li, H.; Ren, Q.; Wang, W. Concept drift type identification based on multi-sliding windows. *Inf. Sci.* **2022**, *585*, 1–23. [[CrossRef](#)]
25. Raab, C.; Heusinger, M.; Schleif, F.-M. Reactive Soft Prototype Computing for Concept Drift Streams. *Neurocomputing* **2020**, *416*, 340–351. [[CrossRef](#)]
26. Gözüaçık, Ö.; Büyükçakır, A.; Bonab, H.; Can, F. Unsupervised concept drift detection with a discriminative classifier. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing China, 3–7 November 2019; pp. 2365–2368.
27. Cerqueira, V.; Gomes, H.M.; Bifet, A.; Torgo, L. STUDD: A student–teacher method for unsupervised concept drift detection. *Mach. Learn.* **2022**, 1–28. [[CrossRef](#)]
28. Chiu, C.W.; Minku, L.L. A Diversity Framework for Dealing with Multiple Types of Concept Drift Based on Clustering in the Model Space. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1299–1309. [[CrossRef](#)]
29. Mahdi, O.A.; Pardede, E.; Ali, N.; Cao, J. Diversity measure as a new drift detection method in data streaming. *Knowl.-Based Syst.* **2020**, *191*, 105227. [[CrossRef](#)]
30. Yuan, L.; Li, H.; Xia, B.; Gao, C.; Liu, M.; Yuan, W.; You, X. Recent Advances in Concept Drift Adaptation Methods for Deep Learning. In Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022; International Joint Conferences on Artificial Intelligence Organization: Vienna, Austria, 2022; pp. 5654–5661.
31. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [[CrossRef](#)]

32. Leväsluoto, J.; Kohl, J.; Sigfrids, A.; Pihlajamäki, J.; Martikainen, J. Digitalization as an Engine for Change? Building a Vision Pathway towards a Sustainable Health Care System by Using the MLP and Health Economic Decision Modelling. *Sustainability* **2021**, *13*, 13007. [[CrossRef](#)]
33. Guo, H.; Zhang, S.; Wang, W. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift. *Neural Netw.* **2021**, *142*, 437–456. [[CrossRef](#)]
34. Han, Y.-N.; Liu, J.-W.; Xiao, B.-B.; Wang, X.-T.; Luo, X.-L. Bilevel Online Deep Learning in Non-stationary Environment. In Proceedings of the 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021. [[CrossRef](#)]
35. Pratama, M.; Za’In, C.; Ashfahani, A.; Ong, Y.S.; Ding, W. Automatic Construction of Multi-layer Perceptron Network from Streaming Examples. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing China, 3–7 November 2019. [[CrossRef](#)]
36. Priya, S.; Uthra, R.A. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex Intell. Syst.* **2021**, 1–17. [[CrossRef](#)]
37. Dezfouli, A.; Griffiths, K.; Ramos, F.; Dayan, P.; Balleine, B.W. Models that learn how humans learn: The case of decision-making and its disorders. *PLoS Comput. Biol.* **2019**, *15*, e1006903. [[CrossRef](#)]
38. Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [[CrossRef](#)]
39. Jagait, R.K.; Fekri, M.N.; Grolinger, K.; Mir, S. Load Forecasting Under Concept Drift: Online Ensemble Learning with Recurrent Neural Network and ARIMA. *IEEE Access* **2021**, *9*, 98992–99008. [[CrossRef](#)]
40. Singh, M.N.; Khaiyum, S. Enhanced Data Stream Classification by Optimized Weight Updated Meta-learning: Continuous learning-based on Concept-Drift. *Int. J. Web Inf. Syst.* **2021**, *17*, 645–668. [[CrossRef](#)]
41. Darem, A.A.; Ghaleb, F.A.; Al-Hashmi, A.A.; Abawajy, J.H.; Alanazi, S.M.; Al-Rezami, A.Y. An Adaptive Behavioral-Based Incremental Batch Learning Malware Variants Detection Model Using Concept Drift Detection and Sequential Deep Learning. *IEEE Access* **2021**, *9*, 97180–97196. [[CrossRef](#)]
42. Das, M.; Pratama, M.; Savitri, S.; Zhang, J. MUSE-RNN: A Multilayer Self-Evolving Recurrent Neural Network for Data Stream Classification. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 110–119. [[CrossRef](#)]
43. Badri, A.K.; Heikal, J.; Terah, Y.A.; Nurjaman, D.R. Decision-Making Techniques using LSTM on Antam Mining Shares before and during the COVID-19 Pandemic in Indonesia. *Aptisi Trans. Manag.* **2021**, *6*, 167–180. [[CrossRef](#)]
44. Saidani, I.; Ouni, A.; Mkaouer, M.W. Improving the prediction of continuous integration build failures using deep learning. *Autom. Softw. Eng.* **2022**, *29*, 21. [[CrossRef](#)]
45. Mansour, R.F.; Al-Otaibi, S.; Al-Rasheed, A.; Aljuaid, H.; Pustokhina, I.V.; Pustokhin, D.A. An Optimal Big Data Analytics with Concept Drift Detection on High-Dimensional Streaming Data. *Comput. Mater. Contin.* **2021**, *68*, 2843–2858. [[CrossRef](#)]
46. Alencar, B.M.; Canário, J.P.; Neto, R.L.; Prazeres, C.; Bifet, A.; Rios, R.A. Fog-DeepStream: A new approach combining LSTM and Concept Drift for data stream analytics on Fog computing. *Internet Things* **2023**, *22*, 100731. [[CrossRef](#)]
47. Luo, X.; Zhang, D. An adaptive deep learning framework for day-ahead forecasting of photovoltaic power generation. *Sustain. Energy Technol. Assess.* **2022**, *52*, 16. [[CrossRef](#)]
48. Diez-Olivan, A.; Ortego, P.; Del Ser, J.; Landa-Torres, I.; Galar, D.; Camacho, D.; Sierra, B. Adaptive Dendritic Cell-Deep Learning Approach for Industrial Prognosis Under Changing Conditions. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7760–7770. [[CrossRef](#)]
49. Cheng, M.-M.; Jiang, P.-T.; Han, L.-H.; Wang, L.; Torr, P. Deeply Explain CNN Via Hierarchical Decomposition. *Int. J. Comput. Vis.* **2023**, *131*, 1091–1105. [[CrossRef](#)]
50. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [[CrossRef](#)]
51. Li, X.; Zhang, Z.; Zhao, Z.; Wu, L.; Huo, J.; Zhang, J.; Wang, Y. ECNN: One Online Deep Learning Model for Streaming Ocean Data Prediction. In Proceedings of the ACM International Conference on Intelligent Computing and Its Emerging Applications, Jinan, China, 28–29 December 2021; p. 6.
52. Saadallah, A.; Jakobs, M.; Morik, K. Explainable Online Deep Neural Network Selection Using Adaptive Saliency Maps for Time Series Forecasting. In *Machine Learning and Knowledge Discovery in Databases. Research Track*; Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 12975, pp. 404–420. [[CrossRef](#)]
53. Tian, X.; Ng, W.W.Y.; Xu, H. Deep Incremental Hashing for Semantic Image Retrieval with Concept Drift. *IEEE Trans. Big Data* **2023**, 1–13. [[CrossRef](#)]
54. Lobo, J.L.; Laña, I.; Del Ser, J.; Bilbao, M.N.; Kasabov, N. Evolving Spiking Neural Networks for online learning over drifting data streams. *Neural Netw.* **2018**, *108*, 1–19. [[CrossRef](#)]
55. Da’u, A.; Salim, N. Recommendation system based on deep learning methods: A systematic review and new directions. *Artif. Intell. Rev.* **2020**, *53*, 2709–2748. [[CrossRef](#)]
56. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Trans. Signal Inf. Process.* **2014**, *3*, e2. [[CrossRef](#)]
57. Bank, D.; Koenigstein, N.; Giryas, R. Autoencoders. *arXiv* **2021**, arXiv:2003.05991.

58. Yoon, S.; Lee, Y.; Lee, J.-G.; Lee, B.S. Adaptive Model Pooling for Online Deep Anomaly Detection from a Complex Evolving Data Stream. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2347–2357. [[CrossRef](#)]
59. Friedrich, B.; Sawabe, T.; Hein, A. Unsupervised statistical concept drift detection for behaviour abnormality detection. *Appl. Intell.* **2022**, *53*, 2527–2537. [[CrossRef](#)]
60. Ashfahani, A.; Pratama, M.; Lughofer, E.; Ong, Y.-S. DEVDAN: Deep evolving denoising autoencoder. *Neurocomputing* **2020**, *390*, 297–314. [[CrossRef](#)]
61. Bhatia, S.; Jain, A.; Srivastava, S.; Kawaguchi, K.; Hooi, B. MemStream: Memory-Based Streaming Anomaly Detection. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 610–621. [[CrossRef](#)]
62. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
63. Guan, H.; Wang, Y.; Ma, X.; Li, Y. DCIGAN: A Distributed Class-Incremental Learning Method Based on Generative Adversarial Networks. In Proceedings of the IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Rio de Janeiro, Brazil, 20–24 May 2019; pp. 768–775. [[CrossRef](#)]
64. Guzy, F.; Wozniak, M.; Krawczyk, B. Evaluating and Explaining Generative Adversarial Networks for Continual Learning under Concept Drift. In Proceedings of the International Conference on Data Mining Workshops (ICDMW), Auckland, New Zealand, 7–10 December 2021; pp. 295–303. [[CrossRef](#)]
65. Zhang, N.; Ding, S.; Zhang, J.; Xue, Y. An overview on Restricted Boltzmann Machines. *Neurocomputing* **2018**, *275*, 1186–1199. [[CrossRef](#)]
66. Korycki, L.; Krawczyk, B. Concept Drift Detection from Multi-Class Imbalanced Data Streams. In Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 1068–1079.
67. Wang, W.; Zhang, M. Data Stream Adaptive Partitioning of Sliding Window Based on Gaussian Restricted Boltzmann Machine. In *Artificial Intelligence in China*; Liang, Q., Wang, W., Mu, J., Liu, X., Na, Z., Chen, B., Eds.; Springer: Singapore, 2020; Volume 572, pp. 220–228. [[CrossRef](#)]
68. Miljkovic, D. Brief review of self-organizing maps. In Proceedings of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 1061–1066.
69. Cerri, R.; Junior, J.D.C.; Faria, E.R.; Gama, J. A new self-organizing map based algorithm for multi-label stream classification. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event Republic of Korea, 22–26 March 2021. [[CrossRef](#)]
70. Wiwatcharakoses, C.; Berrar, D. SOINN+, a Self-Organizing Incremental Neural Network for Unsupervised Learning from Noisy Data Streams. *Expert Syst. Appl.* **2020**, *143*, 113069. [[CrossRef](#)]
71. Zheng, W.; Zhao, P.; Chen, G.; Zhou, H.; Tian, Y. A Hybrid Spiking Neurons Embedded LSTM Network for Multivariate Time Series Learning under Concept-drift Environment. *IEEE Trans. Knowl. Data Eng.* **2022**, 1–14. [[CrossRef](#)]
72. Sun, L.; Ji, Y.; Zhu, M.; Gu, F.; Dai, F.; Li, K. A new predictive method supporting streaming data with hybrid recurring concept drifts in process industry. *Comput. Ind. Eng.* **2021**, *161*, 107625. [[CrossRef](#)]
73. Wang, L.; Chen, S.; Chen, F.; He, Q.; Liu, J. B-Detection: Runtime Reliability Anomaly Detection for MEC Services with Boosting LSTM Autoencoder. *IEEE Trans. Mob. Comput.* **2023**, 1–14. [[CrossRef](#)]
74. Yen, S.; Moh, M.; Moh, T.-S. CausalConvLSTM: Semi-Supervised Log Anomaly Detection Through Sequence Modeling. In Proceedings of the 18th IEEE International Conference on Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1334–1341.
75. Perera, P.P.B. Anomaly Detection on Single Variate Time Series with Concept drifts. Ph.D. Thesis, University of Colombo School of Computing, Colombo, Sri Lanka, 2021.
76. Zhang, X.; Zou, Y.; Li, S. Enhancing incremental deep learning for FCCU end-point quality prediction. *Inf. Sci.* **2020**, *530*, 95–107. [[CrossRef](#)]
77. Museba, T.; Nelwamondo, F.; Ouahada, K.; Akinola, A. Recurrent Adaptive Classifier Ensemble for Handling Recurring Concept Drifts. *Appl. Comput. Intell. Soft Comput.* **2021**, *2021*, 5533777. [[CrossRef](#)]
78. Massaoudi, M.; Chihi, I.; Abu-Rub, H.; Refaat, S.S.; Oueslati, F.S. Convergence of Photovoltaic Power Forecasting and Deep Learning: State-of-Art Review. *IEEE Access* **2021**, *9*, 136593–136615. [[CrossRef](#)]
79. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
80. Zhang, D.; China Electric Power Research Institute; Han, X.; Deng, C. Taiyuan University of Technology. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370. [[CrossRef](#)]
81. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In Proceedings of the 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018. [[CrossRef](#)]
82. Kauschke, S.; Lehmann, D.H.; Furnkranz, J. Patching Deep Neural Networks for Nonstationary Environments. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019.
83. Disabato, S.; Roveri, M. Learning Convolutional Neural Networks in presence of Concept Drift. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; p. 8. [[CrossRef](#)]

84. Pratama, M.; de Carvalho, M.; Xie, R.; Lughofer, E.; Lu, J. ATL: Autonomous Knowledge Transfer from Many Streaming Processes. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019.
85. Xu, L.; Ding, X.; Peng, H.; Zhao, D.; Li, X. ADTCD: An Adaptive Anomaly Detection Approach Towards Concept-Drift in IoT. *IEEE Internet Things J.* **2023**, 1–13. [[CrossRef](#)]
86. Yang, T.; Tang, H.; Bai, C.; Liu, J.; Hao, J.; Meng, Z.; Liu, P.; Wang, Z. Exploration in Deep Reinforcement Learning: A Comprehensive Survey. *arXiv* **2022**, arXiv:2109.06668.
87. Soleymani, F.; Paquet, E. Deep graph convolutional reinforcement learning for financial portfolio management—DeepPocket. *Expert Syst. Appl.* **2021**, *182*, 115127. [[CrossRef](#)]
88. Zhang, Q.; Wang, Z.; Long, C.; Huang, C.; Yiu, S.M.; Liu, Y.; Cong, G.; Shi, J. Online Anomalous Subtrajectory Detection on Road Networks with Deep Reinforcement Learning. *arXiv* **2022**, arXiv:2211.08415. [[CrossRef](#)]
89. Saadallah, A.; Morik, K. Online Ensemble Aggregation using Deep Reinforcement Learning for Time Series Forecasting. In Proceedings of the IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), Porto, Portugal, 6–9 October 2021; pp. 1–8. [[CrossRef](#)]
90. Yu, H.; Chen, Z.; Zhang, X.; Chen, X.; Zhuang, F.; Xiong, H.; Cheng, X. FedHAR: Semi-Supervised Online Learning for Personalized Federated Human Activity Recognition. *IEEE Trans. Mob. Comput.* **2021**, *22*, 3318–3332. [[CrossRef](#)]
91. Liu, Z.; Loo, C.K.; Pasupa, K.; Seera, M. Meta-cognitive recurrent kernel online sequential extreme learning machine with kernel adaptive filter for concept drift handling. *Eng. Appl. Artif. Intell.* **2020**, *88*, 103327. [[CrossRef](#)]
92. da Silva, C.A.S.; Krohling, R.A. Semi-Supervised Online Elastic Extreme Learning Machine with Forgetting Parameter to deal with concept drift in data streams. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
93. Tsukada, M.; Kondo, M.; Matsutani, H. A Neural Network-Based On-device Learning Anomaly Detector for Edge Devices. *IEEE Trans. Comput.* **2020**, *69*, 1027–1044. [[CrossRef](#)]
94. Asuncion, A.; Newman, D. *UCI Machine Learning Repository*; University of California: Irvine, CA, USA, 2007.
95. Harries, M.; Wales, N.S. *Splice-2 Comparative Evaluation: Electricity Pricing*; Research Gate: Berlin, Germany, 1999.
96. Elwell, R.; Polikar, R. Incremental Learning of Concept Drift in Nonstationary Environments. *IEEE Trans. Neural Netw.* **2011**, *22*, 1517–1531. [[CrossRef](#)]
97. Katakis, I.; Tsoumakas, G.; Vlahavas, I. Tracking recurring contexts using ensemble classifiers: An application to email filtering. *Knowl. Inf. Syst.* **2010**, *22*, 371–391. [[CrossRef](#)]
98. Lopez-Paz, D.; Ranzato, M.A. Gradient Episodic Memory for Continual Learning. *arXiv* **2022**, arXiv:1706.08840.
99. Street, W.N.; Kim, Y. A streaming ensemble algorithm (SEA) for large-scale classification. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, 26–29 August 2001; ACM Press: New York, NY, USA, 2001; pp. 377–382.
100. Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. Learning with Drift Detection. In *Advances in Artificial Intelligence—SBIA 2004*; Bazzan, A.L.C., Labidi, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3171, pp. 286–295. [[CrossRef](#)]
101. Fan, W.; Yu, P.; Han, J. Mining Concept-Drifting Data Streams using Ensemble Classifiers. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; p. 10.
102. Kakisim, A.G.; Nar, M.; Sogukpinar, I. Metamorphic malware identification using engine-specific patterns based on co-opcode graphs. *Comput. Stand. Interfaces* **2020**, *71*, 103443. [[CrossRef](#)]
103. Anguita, D.; Ghio, A.; Oneto, L.; Parra-Llanas, X.; Reyes-Ortiz, J. *Proceedings of the International Conference on European Symposium on Artificial Neural Networks (ESANN)*; ESAN: Bruges, Belgium, 2013; Volume 3, p. 3.
104. Coates, A.; Ng, A.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 215–223.
105. Elson, J.; Douceur, J.R.; Howell, J.; Saul, J. Asirra: A CAPTCHA that exploits interest-aligned manual image categorization. *CCS* **2007**, *7*, 366–374.
106. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
107. Xu, W.; Huang, L.; Fox, A.; Patterson, D.; Jordan, M.I. Detecting large-scale system problems by mining console logs. In Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, Big Sky, MT, USA, 11–14 October 2009; ACM Press: New York, NY, USA, 2009.
108. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
109. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep learning for anomaly detection: A review. *ACM Comput. Surv.* **2021**, *54*, 1–38. [[CrossRef](#)]
110. Ganaie, M.A.; Hu, M.; Malik, A.; Tanveer, M.; Suganthan, P. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [[CrossRef](#)]

111. Yang, W.; Li, Z.; Liu, M.; Lu, Y.; Cao, K.; Maciejewski, R.; Liu, S. Diagnosing concept drift with visual analytics. In Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), Salt Lake City, UT, USA, 25–30 October 2020; pp. 12–23.
112. Stiglic, G.; Kokol, P. Interpretability of Sudden Concept Drift in Medical Informatics Domain. In Proceedings of the IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, Canada, 11–14 December 2011; pp. 609–613. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.