Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.18. (1) 2026, pp. Comp 185–202

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

A Survey on Incremental Learning Techniques for Streaming data

Akram Asdkhan Mussa*[a], Ali Saeed Alfoudi[a], Ali Hakem Asaeedi[a]

[a]*College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwaniyah, Iraq,*

akram.asdkan@gmail.com, ali.alsaeedi@qu.edu.iq, ali.alsaeedi@qu.edu.iq

ARTICLE INFO

Article history:
Received: 26 /11/2025
Rrevised form: 13 /01/2026
Accepted : 18 /12/2026
Available online: 30 /03/2026

Keywords: Machine Learning, Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Incremental Learning, Streaming Data, Concept Drift

Each keyword to start on a new line

ABSTRACT

The rapid growth of data stream applications, such as Internet of Things (IoT) systems, smart environments, and real-time analytics, has intensified the need for learning models capable of adapting to continuously evolving data distributions. The traditional techniques of batch learning presuppose fixed data and as such they find it hard to sustain performance when concept drift occurs where the characteristics of data evolve slowly, suddenly or repeatedly with time. Incremental learning has become one of the most important solutions to this issue because it allows models to keep on updating themselves with new incoming data and retain the information previously learned and does not force them to retrain at a high cost. This survey provides an in-depth overview of incremental learning approaches that are used in the streaming data setting with concept drift. We critically review supervised, unsupervised and semi-supervised machine learning methods and deep learning and hybrid methods. The review studies the fundamental adaptation techniques, such as sliding windows, replay, prototype-based learning, methods of detecting drift, expansion of dynamic architecture, and stability-plasticity balancing. To each category we comment on the underlying mechanisms, strength, weakness, computational efficiency and their ability to adapt to various drift conditions. Overall, 38 more recent studies are critically analyzed and compared in the wide range of application areas, data sets, and indicators of assessment. Some of the challenges that are highlighted in this survey include catastrophic forgetting, scalability, interpretability, and resistance to complex and recurring concept drift. Lastly, we recognize the important research gaps and establish future plans on how to develop coherent, scalable and explainable incremental learning models of actual streaming data systems in the world.

https://doi.org/10.29304/jqcsm.2026.18.12522

Introduction

Data streaming has become the main thing that keeps many real-world applications running, like social media, smart systems, and the Internet of Things. This data comes in huge amounts and keeps changing all the time, which makes it fast and unstable. The data distribution changes over time because of different kinds of drift, sometimes gradual, sometimes sudden, and sometimes recurring, depending on different factors. On the other hand, batch learning algorithms expect stable data, so they struggle to deal with streaming data and to maintain their performance over time [1] . The main motivation behind this study comes from the need for models that can adapt to changes in real time and keep updating continuously, without needing full retraining, which is exactly what incremental learning provides This approach lets the model use partial memory to take in new data while keeping what it already learned, which helps reduce computational cost and avoid catastrophic forgetting [2].and supports

*Corresponding author: Akram Asdkhan Mussa

Email addresses: akram.asdkan@gmail.com

Communicated by 'sub etitor'

making reliable and appropriate decisions when dealing with streaming data in modern smart systems and applications. Intelligent models can rely on incremental learning as a core and essential solution, This allows models to update their weights gradually, learn new patterns and knowledge, and keep what they learned before, without needing to train from scratch [3]. This survey covers "more than 37 recent studies" and provides a clear review and critical analysis, ranging from supervised and unsupervised machine learning models to deep learning and hybrid approaches. It also points out the main challenges of each model and the key advantages that help in choosing suitable solutions and decision-making methods for handling data drift in dynamic environments.

2. Incremental Learning (IL): Incremental Learning (IL) is a learning model that allows models to update their knowledge in a sequential manner using new data and can maintain previously learned data and adapt to changing data distributions without retraining on the data..[2].

2.1 Terminology and Definitions

Incremental Learning can be defined as a learning environment whereby models are adapted based on successive batches of new data, and models are not required to be retrained. The paradigm is specifically applicable to streaming data environments, and in particular, those that are influenced by concept drift[4]

Continuous (or Continual) Learning is concerned with facilitating models to learn new tasks or concepts in a sequence over time whilst preventing disastrous forgetting. The context in which this paradigm is most actively researched is that of deep learning where an efficient balance between stability and plasticity is a major challenge[5].

Online Learning: is a model that updates as soon as a new datum is received and processes one datum at a time and requires no long-term memory and has very limited memory and computation resources. Although this can be used to adapt quickly with short-term changes, it is not as efficient as long-term knowledge[6].

Incremental Learning is the main choice of learning paradigm in this survey because it is the one that best fits the context of streaming data where the data have a sequence of batches and the concept drift exists. Online Learning and Continual Learning are presented as similar but different paradigms in terms of updates occurring instance-wise with little memory or task-/class-incremental environments with a specific focus on catastrophic forgetting mitigation (in deep learning), respectively. This survey is hence focused on Incremental Learning techniques suitable in data streams with concept drift.

2.2. The benefits of continuous learning in computer science and real life

- In computers: Reduce the model forgetting: - Keep knowledge that learns from past training and learn the new patterns. Learning without full training: - no learning from scratch, which mean reduce cost of computation and less effort
- Using simple memory: used a small part of current data, it suitable for Resource-Limited Systems
- Dealing with continues and evolving data in real time: suitable for IOT devices
- Ability to detect drift such as (sudden, gradual) which help the system performance healthy over time.
- Supporting Intelligent Real-Time Systems :such as fraud detection, security monitoring, and real-time analytics. All the points explained in this section are based on the comprehensive survey [7].

All these benefits bring incremental learning to the forefront as a critical paradigm on the development of adaptive and sustainable intelligent systems, especially those operating in a constantly changing data environment, with scarce computational resources, and changing distribution of data.

2.3. The incremental learning applied in many field

- Computer vision(cv):- Tracking people or faces, recognizing new objects and images as they show up, or when new categories appear[8].

- NLP/text streaming:- Monitoring and analyzing tweets in real time, or sorting streaming data into categories like news, sports, and politics[9].
- Cyber security : - Detecting abnormal patterns or attacks in data streams continuously[10].
- Recommender systems : - Analyzing user choices and updating their preferences in real time[11].

2.4 . In real life

Real-world data is not static; it keeps changing all the time because of many factors, like changes in wind speed or gradual and sudden increases in temperature. That's why continuous learning helps systems work intelligently and adapt to these changes over time, leading to more accurate and realistic decisions Used in Medicine and healthcare such as EEG monitoring , Industries and IoT like noticing if ammonia gas readings start changing, Smart cities and transportation: indicate traffic and noisy areas and just taking alternative path[12]. These practical uses underscore that the traditional learning models are essentially ineffective in the dynamic environments that encourage a need to be using integration learning models that can adapt continuously, remain stable over long periods, and make real-time decisions.

3 . General Techniques of Incremental Learning

In real environment data changing or evolving over time, so the incremental learning strategies is used to adapt and learn from old and new data without forgetting all time ready. Most important strategies are summarized below.

- Sliding / Dynamic Window: - Sliding window is a simple incremental learning way where the model looks at the latest data and drops the old stuff that's no longer useful. This helps the model keep up with changes and deal with what's actually happening now [13].and its advantage 1-Rapidly adapt to statistical changes in input and output data in order to effectively handle data drift. 2-Reduce memory usage and computational cost by discarding old knowledge that is no longer relevant to current decision-making processes. 3- decision-making on highly relevant and up-to-date information

These benefits help improve model performance, increase efficiency, and allow fast reactions to changing and newly evident patterns in data streams under current conditions [14]

Replay: The system stores important examples of data learned by a small number of people and then displays them upon updating or when new data arrives. The purpose of these older examples is to represent the existing knowledge base and to remind the model of what it has learned previously, thus avoiding starting from scratch. Replays can be of two types: real replays, which are actual examples of important previous data, or generative replays, which are similar examples generated using a generative model to match the older examples [14].

- Prototypes / Summaries :- This technique is based on the fundamental principle of not storing data in its entirety with all details, but rather focusing on summaries. Each category or pattern has its own summary, achieved by creating concise models such as category centers, feature averages, micro clusters, which summarize the local distribution of data. With this technique, the system can retain the most important information representing the general behavior of the data without relying on storing numerous samples. This leads to reduced memory consumption and increased data storage speed in continuous learning environments, especially with large volumes of data and changing patterns. This technology makes a trade-off between efficient storage and preserving fundamental knowledge over time [15].
- Concept Drift Detection :- Drifting is detected by analyzing the change in the statistical distribution between input and output data, where the nature of the data changes. This is done by monitoring errors and observing their gradual increase, which is calculated using tools such as the kl-divergence metric. This technique also uses specialized algorithms to detect gradual or sudden drifts, such as ADWIN (adaptive windowing). When drifts occur, the system activates adaptive procedures to address the drift and adjust model weights, or adds other models that reflect the new data pattern (auxiliary sub-models). In this case, the system maintains its accuracy and stability to keep pace with changes in the data flow and detect drift in dynamic environments [16].Example: ADWIN
- Dynamic Architecture Expansion:- Whenever new knowledge or emerging patterns appear, the system adds to the weights by (expansion), where the system does several things, including increasing the number of nodes in order to represent the knowledge more accurately, or merging thin layers, or adding other thin layers or experts to represent the knowledge. The

structure expansion helps improve the model's range, keeps the current (existing) knowledge, and prevents its interference with other information, so the model becomes adaptable in dealing with flowing data in dynamic environments [17].Used in ML to add simple nodes, in DL to attach new layers or experts, and in Hybrid systems to grow while preserving prior knowledge Example: cPNN [18].

- Stability–Plasticity Balance:- The principle of constancy and flexibility is fundamental to incremental learning. It maintains existing knowledge while allowing for new learning (plasticity). This balance is achieved through several measures, such as adjusting learning rates to minimize the impact of new updates on prior knowledge, freezing layers with stable data, and training modules or layers responsible for processing and understanding incoming data. These measures enable the model to learn continuously without experiencing catastrophic forgetting, while maintaining its ability to adapt to the constant changes in the flow of data .used in dl[19].
- Selective Parameter Update: This refers to the use of partial parameter updates when new knowledge emerges, thus avoiding a complete model retraining. The modification is applied only to a specific set of weights or layers responsible for accommodating the new data (the new pattern), while the remaining weights or layers remain unchanged. This preserves previous knowledge, prevents catastrophic forgetting, and reduces computational effort. The larger, unchanged portion maintains the existing knowledge and stabilizes the model, while the added layers or smaller parts adapt to the influx of data Used in ML to modify only key parameters, in DL to freeze most layers and tune small modules, and in Hybrid systems for rapid, low-cost adaptation.

4. Methods used in (machine, deep learning & hybrid (deep with machine learning)

In machine learning models, incremental learning represents a practical and necessary framework as it allows these systems to acquire knowledge continuously without retraining from scratch. This increases their efficiency in handling data flow in dynamic environments through limited and selective parameter updates and retention of core knowledge, thus reducing the problem of forgetting and providing the ability to adapt to changes and drifts by relying on small data batches and low memory consumption, making them suitable for systems dealing with data flow.

4.1. machine learning models main challenges

- Shallow models weak vs complex drift. ML models (e.g., Naïve Bayes, Logistic Regression, SVM, K-Means) cannot capture evolving or complex data patterns, making them weak under complex concept drift
- Fixed thresholds Supervised ML models depend on fixed thresholds (error, confidence, loss), so when data streams change over time, these thresholds become invalid. which leads to failure under concept drift
- Fails with slow/complex drift Unsupervised models depend only on the current data and don’t have ground truth, so they usually treat small changes as noise and only notice drift after it becomes big.
- Noise/outliers break clusters Unsupervised models lack labels, so noise and outliers can easily distort clusters—shifting centers or forming false clusters—leading to unstable and low-quality clustering, especially in streaming data. These challenges are widely discussed in the literature[20]

Fig.1 illustrate the different incremental learning strategies and indicate which research papers employed each specific type. The someof studies are categorized into eight main groups based on their primary adaptation mechanism, ranging from Sliding Windows and Prototypes to Dynamic Architectures and Memory Replay

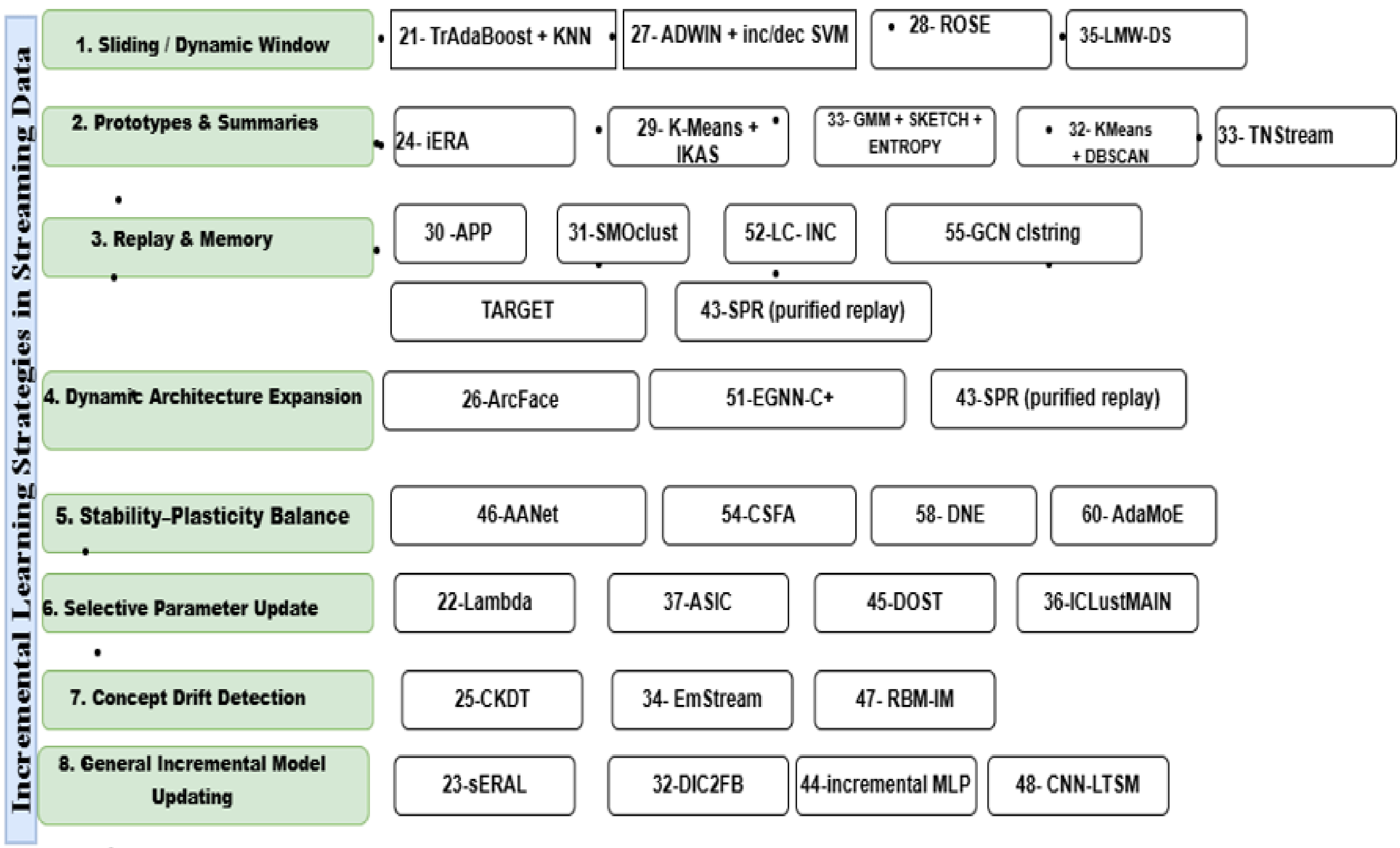

Fig 1 Increment Learning Strategic Taxonomy.

4.2. Related works

in this section explain and analysis directly related works to the research topic. Determine what are goals, methodologies, result, its ability in addition to real challenges and limitations for global understanding to the incremental learning in dynamic environment.

Researchers Gao (2022) et al, presented a sliding window-based framework for detecting changes and used the TrAdaBoost and KNN algorithms for retraining when conceptual drift occurs. The highest accuracy was achieved on fraud of 1.85 m, reaching 97.16 [21].

Ramanath et al. (2021) (Summary Alternative), Lambda Learner is incorporated inside the GAME Framework, updating only the Random-Effects component while maintaining Fixed-Effects static, therefore allowing dependable incremental learning on continuous user-interaction information. Datasets: LinkedIn Ads and MovieLens-20M. Best Result: Higher and more ste ady ROC-AUC than offline retraining [22].

Stržinar et al. (2025), introduced (sERA), based incremental alignment and clustering. Real-time adaptation to concept drift by dynamically creating, merging, or removing clusters. Achieved 81.70% accuracy on an industrial dataset . weak in high-complexity data due to fixed thresholds and simplistic alignment techniques by Stržinar et al[23]

. (2025) proposed (iERAL: incremental Evolution of Representative Alignment) it's strategy was Generates a representative prototype for every pattern and updates them in a single pass as new data arrives. applied on Pneumatic pressure and electrical current signals (Industrial sensors). And the results are V-score = 0.67; Adjusted Rand Index (ARI) = 0.39. iERAL's reliance on fixed thresholds and simplistic alignment [24].

Bartosz Krawczyk et al. (2021) proposed CKDT to deal with tensors. It uses the Rebuilding and Retraining strategy, where a drift detector indicates shifting in streaming data and destroys the feature space, then builds a new tree to adapt to data changes. The method uses Chordal Kernel to maintain the structure of the data and applies McDiarmid in order to incrementally update the decision tree before drift occurs. It achieved an accuracy of 87.92% on the SVHN dataset, but it is weak under sudden drift, since rebuilding is not fast enough to handle high shifting over time[25].

López-López et al;(2022) Presented an incremental face recognition method by employing ArcFace to extract deep face Numerical representations and SVM ensembles for identity Categorization with only five labeled pictures per person , Utilizing pseudo-labels, the system changes over time; controls model expansion via self-healing to remove wrong model and limiting mechanisms to remove a non-useful model and uses Extreme Value Theory (EVT) to manage unknown identities faces in open-set contexts [26].

Honorius Gâlmeanu et al. (2021.a sliding-window technique suggested that keeps new data with part of the old data, then divides the window into an old section (W0) and a new section (W1). To detect concept drift, the ADWIN test measures the disparity between W0 and W1. When drift is found, old data is eliminated by decremental SVM by slowly reducing its weights (λ); new data is integrated via incremental SVM to update the model without full training. The approach was tested on several datasets; on the COVERTYPE dataset (495,141 samples), 92.17% accuracy was recorded; however, it has a high computational cost and lacks strength beneath concept drift[27].

Cano et al(2022) developed a ROSE system with multi-little-models clustering model that gains learning from various feature subsets to increase diversity; it addresses class imbalance by allocating more weight to uncommon classes. The system also includes a sliding window to guarantee balanced representation of recent samples. The ADWIN approach is used to detect conceptual changes, which leads to the development of new clustering models that respond quickly. The finest performance was seen on the Electricity and Agrawal datasets. ROSE system with complex data , best result achieved on Agrawal (syn) with PKappa = 79.80 / PAUC = 78.38 and on Electricity (real) with PKappa = 62.48.weak with complex data [28].

Jayaratne et al(2021) . presented an unsupervised continuous learning method combining the K-Means streaming algorithm and the IKASL algorithm to track concept evolution in continuously incoming data. The method begins with cluster centers representing existing concepts and then gradually merges or splits these centers to detect any concept drift, while discarding outdated concepts over time. This method was tested on the synthetic SEA dataset and two real-world datasets: PAMAP2 and smart electricity meters. The results showed strong performance in handling sudden concept drift (F1 = 91.50%) and repeated drift (F1 = 95.57%) in the PAMAP2 dataset. However, this method cannot detect slow, gradual concept drift, relies on relatively simple algorithms, and its accuracy has been limited to a small set of pre-labeled data.[29].

Francesco Periti et al(2025) presented The APP -A-Posteriori Affinity Propagation algorithm that is a refinement of the traditional AP, which applies three simple steps to stream data. First, it uses Consolidation (summarization of each outdated cluster to a single centroid to prevent re-calculation), which allows clusters to remain constant with the arrival of new data. It then applies Stratification (processing of new data either creating new cluster, enriching an existing cluster or merging nearby clusters) to monitor cluster changes with time. Lastly, it will incorporate Forgetfulness (an aging-based system that obliterates idle clusters) in order to limit the use of memory. These enhance the stability, adaptivity, and effectiveness of APP to continually changing data. tested on the Iris data set, KDD-CUP data set and scored the highest NMI = 0.74 on the Iris data set and NMI = 0.73 on the KDD-CUP data set. APP could be lacking gradual changes due to centroid-based cluster condensation.[30].

The SMOClust algorithm proposed by Chun Wai Chiu et al(2023) .at the first arrival of the data stream addresses the imbalance that existed between large classes with many examples and a small class with few examples, which is the most important class. It takes a new example that it includes in a micro-cluster to summarize its characteristics like mean and dispersion without keeping its original data. In case it does not fit into any of the existing clusters, the algorithm forms a new cluster of it that depicts a novel pattern or a concept. In case it is similar, it is fused with the corresponding cluster. It is these clusters that are then combined to produce synthetic samples which increase the representation of data, datasets used are NOAA (16,344), Covtype (522,911), and INSECTS (128k–406k), and the evaluation was by using G-Mean, achieving the best performance at an imbalance level of ≤ 1%. SMOClust depend on micro-clusters reduces the approach ability to model minority-class changes, especially during gradual drift. [31].

Chaudhari et al. (2024) proposed DIC2FB as method a distributed incremental clustering approach . Clusters occur based on the Closeness Factor without the need to regroup from the beginning within the framework of (incremental learning). It was applied on AWS and Azure and achieved better performance than the K-means algorithm and NFICA by the Silhouette and Dunn Index metrics, which made this method suitable for analyzing data in smart counter environments that change over time. Using fixed thresholds makes the method weak to concept drift.[32].

Aniket Bhanderi et al(2024), created the algorithm on the following basis -Each batch of data is modeled as a normal distribution (mean, number of points, scatter pattern) and stored in a lightweight memory. Clusters are developed or amalgamated as fresh batches come; the less the entropy, the more coherent the cluster. Mahalanobis Distance is a statistical number that is used to detect the outliers based on the distance between a point and the cluster. Along with this, compression module is employed to minimize the number of clusters and simplifies a model by making it lighter and more responsive to the data drift over time, Rand Index best scores of 0.934, 0.891 and 0.908 with simple statistics, fails under complex patterns or slow drift [33].

Alaettin Zubaroglu (2023) proposes Two integrated frameworks to manage streaming data are provided in the study: EmCStream, the initial framework, implements data embedding (embedding: reducing dimensionality without altering the closeness between similar points) then incremental clustering, which adjusts the new incoming data and detects concept drift, in case the data attributes are changing. NoCStream- the second paradigm, on the other hand, is applied in the use of density analysis to predict (Dynamic k-Prediction) the number of clusters to identify the emergence or disappearance of groups as the data streams. The two systems facilitate consistent clustering as time goes by through the collaboration that is adjusted to the constantly evolving nature of the data. In this experiment Tested on Synthetic-1, Meteo-EU, ARI = 1.000 and Purity = 1.000. It cannot be used in large-scale data because it is computationally complex.[34].

, Chen et al;(2025) proposed LMW-DS; an unsupervised window-based streaming clustering algorithm that expands and merges clusters gradually and does not maintain centers of clusters but uses the distance to weight its clusters thus able to deal with slow concept-drifts without re-clustering data already clustered. It is also characteristic of being limited in its capacity to capture subtle trends and minute changes since it is based on reduced form statistics. Best Accuracy: more than 90 percent on Mushrooms data set, since the simplified statistics is not able to cope with the minor concept drift[35].

Ahmed Al-Shammari(2023), suggested IClustMaint, a streaming clustering framework that uses combination of PCA, a dimensionality reduction method, with HAC, the initial cluster formation framework. ICM makes cluster updates incrementally performing them on Border Points only, minimizing the computational cost and execution time. On the Cuff-Less Blood Pressure Estimation data, it was faster than both HAC and BIRCH, but cannot cope with complicated and long-term concept drift[36].

Tianzhen Chen et al(2024suggested ASIC, By modeling each cluster with a Gaussian function to estimate local density from its centroid and sample count, and by updating the scatter matrix Q to capture changes in cluster shape and orientation, the proposed ASIC method performs incremental clustering for IoT data streams. As new data batches arrive, the algorithm efficiently adapts to concept drift by updating centroids, densities, and Q values with minimal computational overhead.used with AWS IoT Dataset the ARI = 0.6629 | Water Monitoring the NMI = 0.6982, ASIC degrades under noisy, dynamic data [37].

The authors adapted the distributed system proposed by Chunxiao Mu et al. (2023), which applies to the Yantai Menlou Reservoir data (China, 2019): Kafka was used to gather stream sensor information, and Flink performed parallel processing and distribution. K-Means was utilised to cluster data that was closely related and DBSCAN was utilised to filter out noise and determine actual clusters. Deniosis indicated a significant difference in NH4-N because it dropped by 1000 to 4.07, and turbidity dropped by 1000 to 61.88. There is, however, a limitation of this method because it is based on a constant Eps value (DBSCAN neighborhood radius) and therefore, it would be less flexible to dynamic streaming data [38]

Qifen Zeng et al. (2025) proposed TNStream, a streaming clustering algorithm based on shared nearest neighbors (SNN) that forms and merges micro clusters using a skeleton set and k-TNC to improve cluster quality. It was evaluated on KDD, Breast Cancer, Iris, and New-Thyroid datasets, achieving high clustering performance (e.g., Purity = 0.90510, ARI = 0.62532, NMI = 0.78798 on Breast Cancer, and Purity = 0.98002, ARI = 0.88143, NMI = 0.85946 on Iris), but requires high computational cost due to complex micro cluster processing[39]

Bi et al. 2020 presented the CODES method which introduced a semi-supervised incremental classification approach using Extreme Learning Machine (ELM). When data flows, the model is updated rapidly. It was tested on Twitter, IBM RSS, and ABC News databases. Twitter Dataset: Accuracy =0.87, Recall =0.47. IBM Dataset: Accuracy =0.86, Recall = 0.52. In also training time, it has a high adaptability to conceptual drift. Performance is affected by sudden drifts due to its reliance on a threshold of recency that needs fine adjustment[40].

Table 1. Summary of Related Works on Streaming and Incremental Learning using Machin learning models.

Ref	Method	Type	Dataset	Metrics,&score	Advantage	Limitation
[21]	slidingWindow + TrAdaBoost + KNN	Supervised / Hybrid	Fraud (1.85M)	Acc = 97.16%	Strong drift handling with delayed labels	It needs delayed labels
[22]	Lambda Learner (GAME)	Supervised	LinkedIn Ads, MovieLens-20M	Stable ROC-AUC	Reliable incremental updates	Works only With limited interaction data
[23]	sERAL	Unsupervised	Industrial Time Series	Acc = 81.70%	Real-time drift adaptation	Noise-sensitive
[24]	iERAL	Unsupervised	Industrial Signals	V-score = 0.67, ARI = 0.39	Single-pass prototype update	Moderate accuracy
[25]	CKDT	Supervised	SVHN	pACC = 87.92%, pmAUC = 83.02%	Preserves tensor structure	Weak under sudden drift
[26]	ArcFace + SVM Ensembles	Supervised	Face Recognition Streams	Not reported	Few-shot & open-set handling	Highly drift-sensitive
[27]	Sliding Window + ADWIN + inc/dec SVM	Supervised	COVERTYPE	Acc = 92.17%	No full retraining	Computationally expensive
[28]	ROSE	Unsupervised	Agrawal, Electricity	PKappa = 79.80	Handles imbalance and drift	Weak with complex data
[29]	K-Means + IKASL	Unsupervised	SEA, PAMAP2	F1 = 95.57%	Handles sudden & repeated drift	Cannot detect gradual drift
[30]	APP (A-Posteriori AP)	Unsupervised	Iris, KDD-CUP	NMI = 0.74	Stable and adaptive clustering	Misses slow changes
[31]	SMOClust	Unsupervised	Streaming Imbalanced Data	Not reported	Handles class imbalance	Primarily designed for imbalance handling, not drift-centric
[32]	DIC2FB	Unsupervised / Distributed	AWS, Azure	Silhouette, Dunn 1	No re-clustering required	Relies on fixed thresholds
[33]	GMM + Sketch + Entropy	Unsupervised	Real & Synthetic Streams	RI ≈ 0.93	Lightweight memory usage	Limited capability under highly complex drift
[34]	EmCStream / NoCStream	Unsupervised	Synthetic, Meteo-EU	ARI = 1.00, Purity = 1.00	ARI = 1.00, Purity = 1.00	High computational cost

[35]	LMW-DS	Unsupervised	Mushrooms	Acc > 90%	Tracks slow drift	Misses subtle patterns
[36]	IClustMaint	Unsupervised	BP Estimation (UCI)	Faster execution	Incremental updates	Not good with long-term drift
[37]	ASIC	Unsupervised	AWS IoT, Water Monitoring	ARI = 0.66, NMI = 0.69	Low computational overhead	Performance drops when data is noisy
[32]	Kafka + Flink + KMeans + DBSCAN	Unsupervised / Distributed	Water Monitoring	NH4-N ↓ to 4.07	Scalable, noise removal	Uses fixed DBSCAN parameters
[39]	TNStream	Unsupervised	KDD, Breast Cancer, Iris	Purity = 0.98	High cluster quality	High computational cost
[40]	CODES (ELM-based)	Semi-supervised	Twitter, IBM RSS	Acc= 0.87	Fast drift adaptation	Very sensitive to threshold tuning

4.3 Deep learning

Deep learning: Deep Learning is a branch of machine learning based on multi-layer artificial neural networks to learn intricate representations by training on massive data. It is also a vital part of streaming analytics, which allows real-time pattern identification, prediction and decision-making, and it is also vital in computer vision, natural language processing, healthcare and autonomous system[41].

4.4. deep learning main challenges

- High computational cost: deep learning models requires for training and updating a lot computing power and time ,high energy consummation especially when deals with a large scale dataset or complex pattern of streaming data[41]
- Catastrophic Forgetting: When the model learns new data, it may forget what it learned before. As a result, it becomes good at the new task but loses its ability to perform the old tasks. [2].
- Weak drift adaptation:Standard deep learning models find it hard to notice and adapt to data changes over time. They do not work well when the data distribution keeps changing[41]
- Memory limits (buffers / replay): deep learning models store an example from previous data that learned during training to reduce forgetting , but the limited sized of memory it saves a small samples but still not enough to represent the old knowledge and make the model less effeminacy in long term [42]
- Poor generalization across datasets: Models trained on one dataset often perform poorly on other datasets[41]

4.5. related works of deep learning approaches

In this section, we review significant recent research on Incremental Learning within the framework of Deep Learning. We highlight the methodologies used, datasets, achieved results, and limitations

Kim et al. (2021) introduced the Self-Purified Replay (SPR) model of learning under noisy labels with the use of self-supervised learning to minimize the error of labeling and a selective replay buffer that is used to preserve reliable samples. The method eliminated about 96 percent of label noise and had an 86.7 percent accuracy on the MNIST dataset. The SPR, however, is constrained to the data of statical image but not adaptation to the continuous data streams. [43].

A framework proposed by Sarmas et al. (2022) on the concept of incremental learning framework to predict photovoltaic (PV) power generation and build electricity load in energy microgrids, basing on the squencess update of an MLP model a real streaming time-series data acquired in an Italian energy network. Only MLP model was used to test the framework with 11.9% performance gains in PV forecasting and load forecasting performance yielded 8.6% performance improvement[44].

The DOST frame learning articles that Wang et al. (2024) presented. It is also partially updated with VIA Adapter in which the model gathers local data in urban data. Simple memory is employed to avoid forgetting, in which prior knowledge is held. DOST was experimented on Chicago-T database where the result produced MAE = 0.72, RMSE = 2.06, WMAPE = 36.80% which was effective in adapting gradual drift and weak in sudden drift[45]

.To minimize forgetting, Liu et al. (2021) proposed the AANets model that employs a structure incorporating two paths (fixed and flexible). Old knowledge is held onto through the fixed path whereas new information is taken in through the flexible path. The trade-off between the two directions is manipulated with the help of two-level optimization of the adaptive weights. This approach was evaluated using CIFAR-100 where it had a high accuracy of 67.59, which suggests that it is capable of adjusting to new classes. Its efficiency is diminished by high flowing wide area data when the two levels of computational complexity poses a problem limiting the efficiency of the model operation[46].

Korycki and Krawczyk (2021) have suggested using the RBM-IM method to detect concept drift in stream data. The technique trains the data pattern with the help of RBM and identifies the drift as changes in the reconstruction error where a large drift is noted. The model then performs the update process of its weights that does not completely retrain it. The method demonstrated high performance (pmAUC = 91.03% on Poker Hand dataset), but it is less efficient in the case of abrupt and rapid drifts.study no 26 The TARGET framework. a continuous distributed learning method is produced by Zhang et al. (2023) , (where machines are trained on native data). only the weights of this data are sent to the server, without sharing the actual data to maintain privacy[47].

A hybrid model of energy consumption prediction is suggested by Kim et al. (2020), which is based on CNN to extract local trends in data and LSTM to model the temporal relationship and works on an Edge-Cloud platform. The system is based on incremental learning in which the model is refined over time by adjusting the training batch size to new changes in data distribution, thus avoiding the necessity to refit the model entirely. The performance of the experiments was strong and the results were RMSE = 0.0630 when the large-scale dataset of KEPCO AM was used. The study, however, emphasizes that incorrect estimation of drift may incite incorrect model revision and this can impact unfavorably on the accuracy of prediction and model stability.[48]

The method is ICLA, a cognitively motivated framework (simulating how the brain applies its memory consolidation capabilities without the need to develop raw inputs), introduced by Rostami and Galstyan (2023). The approach is based on the use of Gaussian Mixture Models (GMM) (a statistical algorithm that represents the essence of classes as abstract groups of data, instead of real images) to generate generative replay (generating synthetic images that are perceived as imagined to store past knowledge). Though it cultivates 97.2% accuracy in MNIST, the method is restricted to the quality of the autoencoders generated images such that bad performance of the synthetic data reconstruction results in poor performance[49].

Prashanth et al. (2025) designed a system that is intelligent to respond to concept drift, which is the abrupt shift in the pattern of health-data, by using a combination of adaptive storage a selective memory that stores only rare and significant cases to updates its immediate model and a Bi-LSTM network that reads data forward and backward to capture context better. This combined method provided the most consistent and the best performance on important measures like Accuracy, F1 -score, and AUC when used on NHANES 20172018 data. Irrespective of the mentioned strengths, the system still has its weaknesses because of its high-computational burden and its use of simulated data shifts as opposed to real-world one[50].

Leite et al. (2024), the proposed EGNN-C+ model is an advanced granular neural network, the nature of which is a, intelligent system that divides knowledge into small units that can be adjusted and expanded online to classify emotions based on EEG (electroencephalography) signals. The mechanism is based on the unceasing update of these units or on the formation of new units as a stream of data flows in, which allows adapting to the immediate change of the state of the brain. Although the model reaches an accuracy of 81.7 percent on the GAMEEMO dataset, it is constrained by its use of weak supervision (session-level labels which fail to effectively represent emotional changes at a moment-to-moment level) and its inability to give strong comparisons with current continual learning methods[51].

Zhou et al. (2022) presented A method of learning new classes according to a deep learning model (LC-INC). This technique incorporates the comparison of a new sample and already familiar sample centers. New class was used to show whether the difference is large and a new center is formed and formed, with a process slowly spreading the model without complete training. This was tested on a variety of databases, including MNIST and Fashion-MNIST and tested on them using Accuracy, F-Measure and AUC. MNIST achieved the best results with ≈ 97% Accuracy and Fashion-MNIST achieved ≈ 90% Accuracy. Nevertheless, performance with high noise present more effectivity and more new classes.[52]

The ALF-GAN, suggested by Eisa et al. (2022), is an intelligent system that will be used in the Spark environment, where the big data will be operated. The system is able to refresh its information immediately and does not involve training at the start, through incremental learning. Important features were also selected using the ALO algorithm and the Fuzzy Bound technology was applied to manage the uncertainty. Supporting data was generated with the help of GAN. It was conducted using the Reuters database and the accuracy was 86.10. This approach is complex in its calculations, which is why it cannot deal with conceptual drift[53].

Yu et al. (2024) presented the CSFA incremental learning method. Base model is frozen in order to avoid catastrophic forgetting, and a prototype (average features of the class) and an optimization by calibration (correcting the representation with older classes) is used to represent each new class. It is able to adapt to concept drift with the aid of the RSGS algorithm: source-free adaptation. This decreases the intensity of losses (sensitivity of the model to variations) and replaces samples with low uncertainty (low probability of certainty) to stabilize the performance. Incremental accuracy was 42.44% on CUB200-C, which is good at working on large data sets. appropriate -tuning of the hyperparameter, and based on simulated drift as opposed to real-world data[54].

Zhao et al. (2025) suggested a method to introduce representative face summaries with the help of GCN (Graph Convolutional Networks). the connection between clusters when new batches have been obtained through summaries are used to identify when they have to enter new clusters or when they are able to be included in existing ones. This method also minimized error accumulation, with efficient incremental clustering with lower computational cost and lower error accumulation being done with, dataset and metrics results: MS-Celeb-1M (small): AF = 0.762, ANMI = 0.955; MS-Celeb-1M (big): AF = 0.672, ANMI = 0.951; IJB-B: AF = 0.690, ANMI = 0.936. Nevertheless, mere mistake in the summaries may compound up and diminish performance[55].

ELM-KL-LSTM incremental learning method is suggested by Zhou et al) 2023), involves using an ELM machine learning model that does light processing (elimination of noise, elimination of outliers, and smoothing) to reduce variation in distribution and an LSTM deep learning model that classifies data. LSTM reads the time series sequentially and classifies the correct one by the pattern in which it has reached i.e. determines the type of signal. It is used to counter signal drift. It was also tested on a CASIA database, which scored 95 per cent, The method however requires the success of the choice of the right update and the size of the data window[56].

The two enhanced versions, A-GEM and OGD, were developed to minimize forgetting (Lamers et al., 2023). Gradients or samples are clustered into groups based on an underlying memory where only the most useful features are stored which are a summary of the overall structure of the data and serve as the cluster centers. This does away with the necessity of knowing when a task began and when it was over. In order to determine the learning ability of the method, it was applied to a number of databases, such as MNIST, Fashion-MNIST, and NOT-MNIST. The overall accuracy over them was 0.878 in the case of MNIST:TA-A-GEM, 0.937 in the case of NOT-MNIST:TA-OGD, and 0.604 in the case of Fashion-MNIST:TA-A-GEM. The obtained results were based on a small dataset, in which simple MLP networks were tested and a simple clustering mechanism (L2 + FIFO). [57]

Hu et al. (2023) presented the DNE technique to process the expansion of model size during learning new tasks sequentially, rather than constructing large new networks, by reuse knowledge that is stored in new tasks that is extracted during learning new tasks extracted from old tasks, as well as the other way around. Through a novel attention mechanism, Ask Attention Block (TAB) that substitutes the conventional MLP layers to be able to combine old and new features effectively. The method proved to be more accurate and keep model size than current methods, as Last Accuracy (LA) of 70.04 on CIFAR-100 and Last Accuracy (LA) of 73.64 on ImageNet-100. the computational cost is proportional to the number of tasks, thus restricting the usage effect to image[58].

. Ashfahani and Pratama (2021) introduced the ADCN model to process streaming data and adapt to changes based on deep clustering (extracting and self-classifying complex patterns without labels), as well as a preservation mechanism that does not make the old knowledge forgotten during the acquisition of the new knowledge. This makes learning sustainable and not having to start afresh. This is referred to as latent organization.. The model's prequential accuracy reached 86.49% when applied to the MNIST database. However, the model suffers from instability and high computational costs when dealing with highly variable and fluctuating data[59].

Addressing the issue of the data patterns gradually varying over time and the statistical distribution of the changes, a AdaMoE model has been presented as a model by Liu et al. (2018). The approach initiates by sending data to a Deep Interest Network (DIN) that is a network that extracts features by representing user interest to produce a readily available representation. These characteristics are further forwarded to the entire expert models to come up with various predictions of Click-Through Rate (CTR). A knowledgeable evaluation module defines the most precise expert and provides him a greater weight. Lastly, the expert predictions are summed together using a Weighted Sum (weighted sum of the expert predictions which put the best expert most influential) to be able to adapt fast without forgetting previous knowledge. It was tested on JD.com with an accuracy of 0.76 AUC. The specialist training approach is expensive in terms of computations[60]

Table 2: Summary of the Deep Learning and Hybrid deep learning.

Ref	Method	Type(deep)	Dataset	Metrics & Score	Advantage	Limitation
[43]	SPR (Self-Purified Replay)	Deep /Self-supervised / Incremental	MNIST	Acc = 86.7%	Removes label noise (96%), reduces forgetting	Limited to still images, no continuous stream adaptation
[44]	Incremental MLP	Deep /Supervised (Forecasting)	Italian energy network	Gain = 11.9% (PV), 8.6% (Load)	Effective PV & Load forecasting	Relies on MLP model only
[45]	DOST + VIA Adapter	Deep /Continuous Learning	Chicago-T	MAE = 0.72, RMSE = 2.06	Adapts to gradual drift, preserves local data	Primarily evaluated under gradual drift scenarios
[46]	AANets (Fixed/Flexible Paths)	Deep /Incremental Learning	CIFAR-100	Acc = 67.59%	Balances stability (old) and plasticity (new)	Efficiency drops with fast/wide-area data
[47]	RBM-IM	Deep /Unsupervised (Drift Detection)	Poker Dataset	pmAUC = 91.03%	Detects drift via reconstruction error	Limited responsiveness under abrupt drift
[48]	CNN LSTM (Edge-Cloud)	Deep /Incremental Learning	KEPCO AM	RMSE = 0.0630	Adapts batch size, no training from scratch	Sensitive to incorrect drift estimates
[49]	ICLA (GMM + Generative Replay)	Cognitively Inspired / Continual	MNIST	Acc = 97.2%	Lightweight memory, stores abstract clusters	Dependent on autoencoder generation quality
[50]	Bi-LSTM + Selective Memory	Deep /Supervised	NHANES 2017–2018	High Acc, F1, AUC	Captures context, handles concept drift	High computational cost and reliance on simulated drift scenarios

[51]	EGNN-C+	Deep /Online / Evolving	GAMEEMO (EEG)	Acc = 81.7%	Adapts to instantaneous brain-state changes	Relies on weak supervision (session-level labels)
[52]	LC-INC	Deep Learning, Stream Learning	MNIST	Acc: =97%	Learn new classes incrementally without full retraining	Designed for class-incremental learning, not drift-aware streaming
[53]	ALF-GAN	Hybrid deep learning	Reuters	Acc: 86.10%	Instant update; handles uncertainty	Complex calc; fails with concept drift
[54]	CSFA	Deep learning	CUB200-C	Acc: 42.44%	Adapts to drift without source data	Adapts to simulated drift under source-free settings
[55]	GCN Clustering	Deep learning	MS-Celeb-1M	AF: 0.76	Efficient; reduced error accumulation	Summary errors can accumulate
[56]	A-GEM / OGD	Deep learning	MNIST, Not-MNIST	Acc: =87-93%	Task agnostic; fixed memory	Tested only on simple networks
[58]	DNE (TAB)	Deep learning	CIFAR, ImageNet	Acc: =70-73%	Compact; efficient knowledge reuse	High cost with many tasks
[59]	ADCN	Deep learning	MNIST	Acc: 86.49%	Prevents forgetting; self-classifying	Unstable/costly with high variance
[60]	AdaMoE	Deep learning	JD.com	AUC: 0.76	Rapid adaptation (Weighted Experts)	High expert training cost

4.6 Analysis of Incremental Methods of Learning in terms of Computational Cost.

Many incremental learning methods are aimed at preventing re-training but the computational cost of these methods can differ greatly, based on the mechanism of adaptation used. Lightweight methods like sliding windows, prototype-based learning and selective parameter updates are usually cheap in terms of computation and can be applied in real-time and resource-constrained systems. Conversely, the computational resources of the deep learning-based and dynamic architecture expansion methods can be significantly increased because of repetitive gradient updates, memory replay or network expansion, making them unfeasible in strict real-time settings. Distributed and hybrid solutions enhance scalability but add extra communication and synchronization costs and a trade-off between performance and efficiency is inevitable.

Table 3 Comparative analysis is based on consistent observations reported in the reviewed literature

Source Studies	Method	Computational Cost	Memory Usage	Real-Time Feasibility	Limitation
[21], [27], [28], [35]	Sliding Window	Low	Low	High	Forgets long-term knowledge
[24], [30]	Prototype-Based Learning	Low	Low	High	Sensitive to prototype quality

[22], [45]	Selective Parameter Update	Low–Moderate	Low	High	Limited adaptation scope
[16], [28], [27]	Drift Detection (e.g., ADWIN)	Moderate	Moderate	Moderate–High	Detection delay
[17], [18]	Dynamic Architecture Expansion	High	High	Low	High training and maintenance cost
[16], [28]	Drift Detection (e.g., ADWIN)	Moderate	Moderate	Moderate–High	Detection delay
[43], [49], [57]	Replay / Memory-Based Learning	High	High	Low	Memory and computation overhead
[32], [38]	Distributed Incremental Learning	Moderate–High	Moderate	Moderate	Communication overhead

5. Databases

Incremental learning (its models) relies on the diversity and quality of data sources that continuously generate data to measure its adaptability and address drift. Databases can be divided into four main types. summarized in the table3 below.

Table 3 important type of datasets used in studies.

Data Type	Description	Example Datasets / Sources	Typical Use in Research
Sensor Data	**Continuous readings generated by physical or IoT devices.**	**• Gas sensors (pH, NH4N, Turbidity) • Temperature & humidity sensors • ECG / EEG sensors**	**Concept drift, streaming analysis, adaptive clustering, anomaly detection**
Time-Series Data	**Sequential values ordered by time; may include trends or drift.**	**Smart meter electricity readings • Water-quality time records • Energy consumption signals**	**Incremental learning, drift detection, forecasting, evolving pattern tracking**
Image Data	**Pixel-based visual information arranged in 2D (or RGB 3D).**	**MNIST • CIFAR-10 / CIFAR-100 • Satellite images (CROP)**	**Online classification, image clustering, representation drift, embedding evolution**
Tabular Data	**Structured rows and columns with numerical or categorical attributes.**	**• KDD99 • LendingClub • Covtype dataset**	**Streaming classification, unsupervised clustering, online model updating**

6. Performance metrics

Many metrics were used, such as Accuracy, F1-Score, and AUC, to measure classification accuracy, and metrics such as RI, ARI, NMI, and Silhouette, to measure assembly quality using specific criteria to obtain an accurate assessment that balances flexibility and stability. The following is a summary of the most important metrics used in the 37 studies.

accuracy (ACC) It represents the number of instances that are accurately predicted (positively and negatively) divided by the total number of instances. Formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Where: TP = True Positives,TN = True Negatives,FP = False Positives,FN = False Negatives

F1-Score is a performance measure that weighted the Precision and Recall into one value by computing the harmonic mean. It offers a balanced assessment of the accuracy of a model, particularly in the case of unbalanced data. Formula:

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

Where: Precision measures how many predicted positives are correct

Recall measures how many actual positives are correctly identified.

AUC (Area Under the ROC Curve) measures how well a model can recognize between the positive and negative classes via all decision thresholds.to ranking quality it's rang [0.5, 1]

RI (Rand Index): measures the agreement between two clustering's based on pairwise consistency.it range 0.5 ,1

$$RI = \frac{\text{Agreement}}{\text{Total Pairs}} \tag{3}$$

Where: Agreement = pairs correctly assigned to the same or different clusters)/Total Pairs = $TP + TN + FP + FN$

ARI (Adjusted Rand Index) is a clustering evaluation metric that measures the similarity between two clusterings while correcting for agreement that occurs by chance.

$$\text{Formula: } ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \tag{4}$$

Where: RI is the Rand Index,$\mathbb{E}[RI]$is the expected Rand Index under random labeling

ARI ranges from −1 to 1, where 1 indicates perfect agreement, 0 corresponds to random clustering, and negative values indicate worse-than-random agreement.

NMI (Normalized Mutual Information) is a clustering evaluation metric that measures the shared information between two clusterings, normalized to allow fair comparison.

$$\text{Formula:} NMI = \frac{2\, I(X;Y)}{H(X) + H(Y)} \tag{5}$$

Where:$I(X;Y)$= Mutual Information between clusterings $X$and Y,$H(X)$, $H(Y)$= Entropy of $X$and Y

Silhouette Score measures how well a data point fits within its own cluster compared to other clusters.

$$\text{Formula:} s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6}$$

Where:$a(i)$= average distance between point $i$and all other points in the same cluster

$b(i)$= minimum average distance between point $i$and points in the nearest neighboring cluster. The score ranges from −1 to 1, where higher values indicate better clustering. All metrics depend on [53]..

7. Discussion and research gap

Speed and instability define data flow, therefore producing patterns that are always shifting. Thus, the move from batch to incremental learning is a unavoidable and essential reaction as well as an original answer for data processing and adapting to drift in dynamic settings. An examination of 38 current research reveals:

- Effectiveness of Drift Adaptation Strategies: The studies show that several techniques have been applied to reduce drift. Deep learning models concentrate on enlarging the dynamic environment to fit fresh patterns without sacrificing previous ones, such as EGNN-C+ and PNN models; traditional approaches depend on sliding windows, such online SVM and ADWIN.
- The dilemma of stability and flexibility: Among the most obvious issues models have tried to solve in order to gain fresh information without losing previously learned material is catastrophic forgetting. Selective parameter updating (LORA), in which only particular weights are set to lower computational expenses, has proven to be successful in saving time among other methods
- Memory efficiency and prototype Prototypes and recall efficiency: Their significance is in summarizing information, such using centers and several points mirroring the overall form or cluster pattern, which dramatically lowers memory use. ASIC and CluStream among other examples.

Table 4- Comparison of Incremental Learning Approaches under Concept Drift.

Approach	Strength	Limitation	Best Use Case
Sliding Window	Fast adaptation, low cost	Loses long-term knowledge	High-speed data streams
Prototype-Based	Stable and memory-efficient	Depends on prototype quality	Gradual or recurring drift
Deep Learning	Captures complex patterns	High cost, low interpretability	High-dimensional data

7.1. research gap

by reviewed studies found, several limitations remain unresolved. Most existing incremental learning method focus on adaptation speed, drift detection, memory efficiency in isolation. However, they often suffer from high computational costs, limited scalability, lack of interpretability, or weak performance under complex and recurring concept drift. Moreover, few studies address incremental learning within distributed real-world streaming environments while maintaining explainability and long-term stability. This leads us to a clear research gap: the lack of a unified incremental learning framework that is scalable, interpretable, adaptable to different environments, and effective in processing streaming data.

8. Conclusion

The incremental learning paradigm has become a necessity and not an optional improvement due to the rapid increase in the volume of streaming data produced by contemporary applications including the IoT system, smart cities, healthcare monitoring, and real-time analytics. In contrast to the classic batch learning, incremental learning allows the model to continually adapt with a new data distribution, and retain the already learned information, overcoming the need to address concept drift issues in dynamical settings. This survey has provided the overall overview of incremental learning methods used on streaming information, including supervised, unsupervised, semi-supervised, deep-learning and hybrid. A systematic analysis and comparison of 38 recent studies in terms of adaptation strategies, performance measures, computational efficiency and resistance to various forms of concept drift, such as abrupt, gradual and recurrent drift were performed. Sliding windows, replay strategies, prototype-based representations, drift detection methods, dynamic architecture expansion, selective parameter updates, stability-plasticity balancing were reviewed and looked into in detail as some of the key incremental learning mechanisms. The discussion points out that there exists no universal best incremental learning strategy. Conventional machine learning models are computationally efficient and quick to adapt, but commonly fail to deal with complex or long-term drift. Deep learning models can learn high-dimensional and very complex patterns and are characterized by high computational cost, catastrophic forgetting, and low interpretability. Hybrid models attempt to strike these trade-offs by integrating lightweight preprocessing or clustering algorithms with deep architectures, with encouraging outcomes in the context of real-world streaming. Even though such achievements have been made, there are still a number of challenges. Most of the current methods take advantage of individual solutions, e.g. speed of adaptation, memory efficiency, or drift detection individually and as such cannot be used to achieve scalability, reduced interpretability or stability to more complex and recurrent drift. In addition, there is a lack of research that considers incremental learning in distributed and real-world streaming scenarios with explainability, minimal resource usage, and sustained stability. The future study needs to entail coming up with simple incremental learning frameworks that combine effective drift detection, selective updates and memory conscious representations and be able to scale as well as be understandable. The future directions seem promising through few-shot and semi-supervised learning to eliminate the reliance on labeled data, generative representations to maintain previous knowledge with minimum stores, and adaptive hybrid architecture to dynamically trade-off between stability and plasticity. These challenges will be important to implement robust and smart incremental learning systems in actual and constantly evolving data environments.

References

[1] B. Halstead, Y. S. Koh, P. Riddle, M. Pechenizkiy, and A. Bifet, "A Probabilistic Framework for Adapting to Changing and Recurring Concepts in Data Streams," in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2022, pp. 1–10. doi: 10.1109/DSAA54385.2022.10032368.

[2] G. M. van de Ven, N. Soures, and D. Kudithipudi, "Continual Learning and Catastrophic Forgetting," 2025, pp. 153–168. doi: 10.1016/B978-0-443-15754-7.00073-0.

[3] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, and X.-M. Wu, "Overcoming Catastrophic Forgetting in Incremental Few-Shot Learning by Finding Flat Minima".

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, p. 44:1-44:37, Mar. 2014, doi: 10.1145/2523813.

[5] L. Wang, X. Zhang, H. Su, and J. Zhu, "A Comprehensive Survey of Continual Learning: Theory, Method and Application," Feb. 06, 2024, *arXiv*: arXiv:2302.00487. doi: 10.48550/arXiv.2302.00487.

[6] N. Zucchet, R. Meier, S. Schug, A. Mujika, and J. Sacramento, "Online learning of long-range dependencies," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10477–10493, Dec. 2023.

[7] I. C. Martin, S. Mukherjee, A. Baimagambetov, J. Vanschoren, and N. Polatidis, "Evolving Machine Learning: A Survey," Oct. 10, 2025, *arXiv*: arXiv:2505.17902. doi: 10.48550/arXiv.2505.17902.

[8] Z. Li and D. Hoiem, "Learning without Forgetting," Feb. 14, 2017, *arXiv*: arXiv:1606.09282. doi: 10.48550/arXiv.1606.09282.

[9] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari, "Continual Learning for Large Language Models: A Survey," Feb. 07, 2024, *arXiv*: arXiv:2402.01364. doi: 10.48550/arXiv.2402.01364.

[10] L. Boukela, G. Zhang, M. Yacoub, and S. Bouzefrane, "A near-autonomous and incremental intrusion detection system through active learning of known and unknown attacks," in *2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Jun. 2021, pp. 374–379. doi: 10.1109/SPAC53836.2021.9539947.

[11] Y. Ouyang, J. Shi, H. Wei, and H. Gao, "Incremental Learning for Personalized Recommender Systems," Aug. 13, 2021, *arXiv*: arXiv:2108.13299. doi: 10.48550/arXiv.2108.13299.

[12] C. Li, Y. Tang, and G. Xu, "SVMWPE-HLLE Based Fault Diagnosis Approach for Rolling Bearing," *IEEE Access*, vol. 11, pp. 4645–4659, 2023, doi: 10.1109/ACCESS.2023.3234528.

[13] D. Mulimani, P. Patil, S. Totad, and R. Benni, "Online Detection and Adaptation of Concept Drift in Streaming Data Classification," *Procedia Computer Science*, vol. 235, pp. 2803–2811, 2024, doi: 10.1016/j.procs.2024.04.265.

[14] T. Wang, X. Li, and Y. Wang, "SAR Image Simulations of Ocean Scenes Based on the Improved Facet TSM," *Sensors*, vol. 23, no. 5, p. 2564, Feb. 2023, doi: 10.3390/s23052564.

[15] Z. Tao, S. Huang, and G. Wang, "Prototypes Sampling Mechanism for Class Incremental Learning," *IEEE Access*, vol. 11, pp. 81942–81952, 2023, doi: 10.1109/ACCESS.2023.3301123.

[16] H. Gâlmeanu and R. Andonie, "Concept Drift Adaptation with Incremental–Decremental SVM," *Applied Sciences*, vol. 11, no. 20, p. 9644, Oct. 2021, doi: 10.3390/app11209644.

[17] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong Learning with Dynamically Expandable Networks," Jun. 11, 2018, *arXiv*: arXiv:1708.01547. doi: 10.48550/arXiv.1708.01547.

[18] F. Giannini, G. Ziffer, and E. Della Valle, "cPNN: Continuous Progressive Neural Networks for Evolving Streaming Time Series," in *Advances in Knowledge Discovery and Data Mining*, vol. 13938, H. Kashima, T. Ide, and W.-C. Peng, Eds., in Lecture Notes in Computer Science, vol. 13938. , Cham: Springer Nature Switzerland, 2023, pp. 328–340. doi: 10.1007/978-3-031-33383-5_26.

[19] G. Lin, H. Chu, and H. Lai, "Towards Better Plasticity-Stability Trade-off in Incremental Learning: A Simple Linear Connector," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 89–98. doi: 10.1109/CVPR52688.2022.00019.

[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Apr. 2014, doi: 10.1145/2523813.

[21] H. Gao, Z. Ding, and M. Pan, "Incremental Learning Method for Data with Delayed Labels," *cai*, vol. 41, no. 5, pp. 1260–1283, 2022, doi: 10.31577/cai_2022_5_1260.

[22] R. Ramanath *et al.*, "Lambda Learner: Fast Incremental Learning on Data Streams," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore: ACM, Aug. 2021, pp. 3492–3502. doi: 10.1145/3447548.3467172.

[23] Ž. Stržinar, I. Škrjanc, M. Pratama, and B. Pregelj, "Evolving Clustering of Time Series for Unsupervised Analysis of Industrial Data Streams," 2024, *SSRN*. doi: 10.2139/ssrn.5026151.

[24] Ž. Stržinar, I. Škrjanc, and B. Pregelj, "Evolving interval-based time series clustering for streaming industrial data," *Evolving Systems*, vol. 16, no. 3, p. 82, Sep. 2025, doi: 10.1007/s12530-025-09713-w.

[25] B. Krawczyk, "Tensor decision trees for continual learning from drifting data streams," *Mach Learn*, vol. 110, no. 11–12, pp. 3015–3035, Dec. 2021, doi: 10.1007/s10994-021-06054-y.

[26] E. Lopez-Lopez, X. M. Pardo, and C. V. Regueiro, "Incremental Learning from Low-labelled Stream Data in Open-Set Video Face Recognition," *Pattern Recognition*, vol. 131, p. 108885, Nov. 2022, doi: 10.1016/j.patcog.2022.108885.

[27] H. Gâlmeanu and R. Andonie, "Concept Drift Adaptation with Incremental–Decremental SVM," *Applied Sciences*, vol. 11, no. 20, p. 9644, Oct. 2021, doi: 10.3390/app11209644.

[28] A. Cano and B. Krawczyk, "ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams," *Mach Learn*, vol. 111, no. 7, pp. 2561–2599, Jul. 2022, doi: 10.1007/s10994-022-06168-x.

[29] D. Jayaratne, D. De Silva, D. Alahakoon, and X. Yu, "Continuous detection of concept drift in industrial cyber-physical systems using closed loop incremental machine learning," *Discov Artif Intell*, vol. 1, no. 1, p. 7, Dec. 2021, doi: 10.1007/s44163-021-00007-z.

[30] F. Periti, S. Montanelli, A. Ferrara, and S. Castano, "Incremental Affinity Propagation Based on Cluster Consolidation and Stratification," *Neural Process Lett*, vol. 57, no. 3, p. 44, Apr. 2025, doi: 10.1007/s11063-025-11752-y.

[31] C. W. Chiu and L. L. Minku, "SMOClust: Synthetic Minority Oversampling based on Stream Clustering for Evolving Data Streams," Aug. 28, 2023, *arXiv*: arXiv:2308.14845. doi: 10.48550/arXiv.2308.14845.

[32] A. Chaudhari, P. Mulay, A. Agarwal, K. Iyer, and S. Sarbhai, "DIC2FBA: Distributed Incremental Clustering with Closeness Factor Based Algorithm for Analysis of Smart Meter Data," *IJCDS*, vol. 15, no. 1, pp. 29–38, Jul. 2024, doi: 10.12785/ijcds/160103.

[33] A. Bhanderi and R. Bhatnagar, "Incremental Gaussian Mixture Clustering for Data Streams," in *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, Dec. 2024, pp. 197–204. doi: 10.1109/ICDMW65004.2024.00032.

[34] A. Zubaroğlu and V. Atalay, "Online embedding and clustering of evolving data streams," *Statistical Analysis*, vol. 16, no. 1, pp. 29–44, Feb. 2023, doi: 10.1002/sam.11590.

[35] J. Chen, J. Chen, and Z. Zhong, "A weight clustering algorithm based on sliding window model for stream data," *Sci Rep*, vol. 15, no. 1, p. 22521, Jul. 2025, doi: 10.1038/s41598-025-96696-y.

[36] A. Al-Shammari, "A Fully Online Clustering Approach for Enhanced Performance of Health Information System," *J. Al-Qadisiyah Comp. Sci. Math.*, vol. 15, no. 2, Sep. 2023, doi: 10.29304/jqcm.2023.15.2.1239.

[37] "An Incremental Clustering Algorithm for Arbitrary Shaped in IoT Data," *AJETS*, vol. 7, no. 3, 2024, doi: 10.25236/AJETS.2024.070325.

[38] C. Mu, Y. Hou, J. Zhao, S. Wei, and Y. Wu, "Stream-DBSCAN: A Streaming Distributed Clustering Model for Water Quality Monitoring," *Applied Sciences*, vol. 13, no. 9, p. 5408, Apr. 2023, doi: 10.3390/app13095408.

[39] Q. Zeng, H. Bao, Y. Hu, Z. Zhang, Y. Zheng, and L. Wen, "TNStream: Applying Tightest Neighbors to Micro-Clusters to Define Multi-Density Clusters in Streaming Data," May 01, 2025, *arXiv*: arXiv:2505.00359. doi: 10.48550/arXiv.2505.00359.

[40] X. Bi, C. Zhang, X. Zhao, D. Li, Y. Sun, and Y. Ma, "CODES: Efficient Incremental Semi-Supervised Classification Over Drifting and Evolving Social Streams," *IEEE Access*, vol. 8, pp. 14024–14035, 2020, doi: 10.1109/ACCESS.2020.2965766.

[41] L. Alzubaidi *et al.*, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *J Big Data*, vol. 10, no. 1, p. 46, Apr. 2023, doi: 10.1186/s40537-023-00727-2.

[42] M. A. Shyaa, Z. Zainol, R. Abdullah, M. Anbar, L. Alzubaidi, and J. Santamaría, "Enhanced Intrusion Detection with Data Stream Classification and Concept Drift Guided by the Incremental Learning Genetic Programming Combiner," *Sensors*, vol. 23, no. 7, p. 3736, Apr. 2023, doi: 10.3390/s23073736.

[43] C. D. Kim, J. Jeong, S. Moon, and G. Kim, "Continual Learning on Noisy Data Streams via Self-Purified Replay," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 517–527. doi: 10.1109/ICCV48922.2021.00058.

[44] E. Sarmas, S. Strompolas, V. Marinakis, F. Santori, M. A. Bucarelli, and H. Doukas, "An Incremental Learning Framework for Photovoltaic Production and Load Forecasting in Energy Microgrids," *Electronics*, vol. 11, no. 23, p. 3962, Nov. 2022, doi: 10.3390/electronics11233962.

[45] C. Wang, G. Tan, S. B. Roy, and B. C. Ooi, "Distribution-aware Online Continual Learning for Urban Spatio-Temporal Forecasting," Nov. 24, 2024, *arXiv*: arXiv:2411.15893. doi: 10.48550/arXiv.2411.15893.

[46] Y. Liu, B. Schiele, and Q. Sun, "Adaptive Aggregation Networks for Class-Incremental Learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 2544–2553. doi: 10.1109/CVPR46437.2021.00257.

[47] Ł. Korycki and B. Krawczyk, "Concept Drift Detection from Multi-Class Imbalanced Data Streams," Apr. 20, 2021, *arXiv*: arXiv:2104.10228. doi: 10.48550/arXiv.2104.10228.

[48] S.-H. Kim, C. Lee, and C.-H. Youn, "An Accelerated Edge Cloud System for Energy Data Stream Processing Based on Adaptive Incremental Deep Learning Scheme," *IEEE Access*, vol. 8, pp. 195341–195358, 2020, doi: 10.1109/ACCESS.2020.3033771.

[49] M. Rostami and A. Galstyan, "Cognitively Inspired Learning of Incremental Drifting Concepts," Apr. 21, 2023, *arXiv*: arXiv:2110.04662. doi: 10.48550/arXiv.2110.04662.

[50] B. S. Prashanth *et al.*, "Adaptive Buffering Strategies for Incremental Learning Under Concept Drift in Lifestyle Disease Modeling," *IEEE Access*, vol. 13, pp. 174001–174033, 2025, doi: 10.1109/ACCESS.2025.3616776.

[51] D. Leite, A. Silva, G. Casalino, A. Sharma, D. Fortunato, and A.-C. Ngomo, "EGNN-C+: Interpretable Evolving Granular Neural Network and Application in Classification of Weakly-Supervised EEG Data Streams," Feb. 26, 2024, *arXiv*: arXiv:2402.17792. doi: 10.48550/arXiv.2402.17792.

[52] D.-W. Zhou, Y. Yang, and D.-C. Zhan, "Learning to Classify With Incremental New Class," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 33, no. 6, pp. 2429–2443, Jun. 2022, doi: 10.1109/TNNLS.2021.3104882.

[53] A. Eisa, N. EL-Rashidy, M. Dahman Alshehri, H. M. El-bakry, and S. Abdelrazek, "Incremental Learning Framework for Mining Big Data Stream," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 2901–2921, 2022, doi: 10.32604/cmc.2022.021342.

[54] E. Yu, J. Lu, and G. Zhang, "Generalized Incremental Learning under Concept Drift across Evolving Data Streams," Jun. 06, 2025, *arXiv*: arXiv:2506.05736. doi: 10.48550/arXiv.2506.05736.

[55] X. Zhao, Z. Wang, L. Gao, Y. Li, and S. Wang, "Incremental face clustering with optimal summary learning via graph convolutional network," *Tinshhua Sci. Technol.*, vol. 26, no. 4, pp. 536–547, Aug. 2021, doi: 10.26599/TST.2020.9010024.

[56] Q. Zhou, Z.-Y. Wang, and L. Huang, "ELM-KL-LSTM: a robust and general incremental learning method for efficient classification of time series data," *PeerJ Computer Science*, vol. 9, p. e1732, Dec. 2023, doi: 10.7717/peerj-cs.1732.

[57] C. Lamers, R. Vidal, N. Belbachir, N. Van Stein, T. Bäck, and P. Giampouras, "Clustering-based Domain-Incremental Learning," in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Paris, France: IEEE, Oct. 2023, pp. 3376–3384. doi: 10.1109/ICCVW60793.2023.00363.

[58] R. Hu, Luc, Chen, and Y. Wang, "An Efficient Outlier Detection Algorithm for Data Streaming," Jan. 02, 2025, *arXiv*: arXiv:2501.01061. doi: 10.48550/arXiv.2501.01061.

[59] A. Ashfahani and M. Pratama, "Unsupervised Continual Learning in Streaming Environments," Sep. 20, 2021, *arXiv*: arXiv:2109.09282. doi: 10.48550/arXiv.2109.09282.

[60] C. Liu *et al.*, "On the Adaptation to Concept Drift for CTR Prediction," Feb. 23, 2023, *arXiv*: arXiv:2204.05101. doi: 10.48550/arXiv.2204.05101.