electronics

MDPI

Article

Class-Adaptive Weighted Broad Learning System with Hybrid Memory Retention for Online Imbalanced Classification

Jintao Huang [1,2,*], Yu Wang [2] and Mengxin Wang [1,*]

[1] Guangzhou Institute of International Finance, Guangzhou University, Guangzhou 510006, China
[2] Zhuhai DeltaFit Technology Company Ltd., Zhuhai 519031, China; wangyu@126.com
* Correspondence: violler@foxmail.com (J.H.); wangmx@gzhu.edu.cn (M.W.)

Abstract

Data stream classification is a critical challenge in data mining, where models must rapidly adapt to evolving data distributions and concept drift in real time, while extreme learning machines offer fast training and strong generalization, most existing methods struggle to jointly address multi-class imbalance, concept drift, and the high cost of label acquisition in streaming settings. In this paper, we present the Adaptive Broad Learning System for Online Imbalanced Classification (ABLS-OIC), which introduces three core innovations: (1) a Class-Adaptive Weight Matrix (CAWM) that dynamically adjusts sample weights according to class distribution, sample density, and difficulty; (2) a Hybrid Memory Retention Mechanism (HMRM) that selectively retains representative samples based on importance and diversity; and (3) a Multi-Objective Adaptive Optimization Framework (MAOF) that balances classification accuracy, class balance, and computational efficiency. Extensive experiments on ten benchmark datasets with varying imbalance ratios and drift patterns show that ABLS-OIC consistently outperforms state-of-the-art methods, with improvements of 5.9% in G-mean, 6.3% in F1-score, and 3.4% in AUC. Furthermore, a real-world credit fraud detection case study demonstrates the practical effectiveness of ABLS-OIC, highlighting its value for early detection of rare but critical events in dynamic, high-stakes applications.

Keywords: class-imbalanced; broad learning system; online classification learning

Academic Editor: Farrukh Saleem

Received: 4 August 2025
Revised: 30 August 2025
Accepted: 3 September 2025
Published: 8 September 2025

Citation: Huang, J.; Wang, Y.; Wang, M. Class-Adaptive Weighted Broad Learning System with Hybrid Memory Retention for Online Imbalanced Classification. *Electronics* **2025**, *14*, 3562. https://doi.org/10.3390/electronics14173562

1. Introduction

The advent of big data has fueled an explosive growth of streaming data across a wide spectrum of domains, including network monitoring [1], financial transactions [2], social media analytic [3], healthcare surveillance [4], and industrial systems [5–7]. Unlike conventional batch learning settings, data streams arrive continuously and indefinitely, demanding that models process and adapt to new information in real time without the luxury of storing the entire data history [8–10]. This paradigm introduces formidable challenges for learning algorithms, particularly in the context of class imbalance—a pervasive issue where certain (majority) classes vastly outnumber others (minority classes) [11–13].

Class imbalance is intrinsic to numerous real-world applications. For example, in fraud detection, legitimate transactions dramatically outnumber fraudulent ones [14]; in industrial fault diagnosis, normal operational data dominates while fault events are rare [15]; and in medical diagnostics, healthy cases are far more common than specific disease occurrences [16]. The class imbalance ratio—defined as the proportion of majority to minority samples—can reach extreme levels, frequently exceeding 100:1 or even 1000:1 in critical applications such as intrusion detection or rare disease diagnosis [17,18].

The intersection of online classification learning and class imbalance creates unique challenges that conventional machine learning approaches are ill-equipped to address. These challenges are evident in several critical aspects: first, class distribution skewness causes classifier bias toward majority classes, as traditional error minimization objectives naturally favor the prevalent class to maximize overall accuracy [19]. This bias is especially problematic in applications where minority class detection is paramount, such as fraud or disease diagnosis, where the cost of misclassification is disproportionately high [20]. Second, streaming environments often exhibit non-stationary data distributions characterized by various forms of concept drift [21,22], including virtual concept drift (changes in $P(\mathbf{x})$), real concept drift (changes in $P(y|\mathbf{x})$), and class ratio drift (changes in $P(y)$). The interplay of concept drift and class imbalance exacerbates learning difficulties, as adaptation mechanisms must address both evolving distributions and minority class sensitivity [23]. Third, online classification learning imposes stringent constraints on computational resources and memory usage [24]. Models must process each instance rapidly—often in a single pass—without access to the full data history, which conflicts with many traditional imbalanced learning techniques reliant on resampling [25]. Fourth, majority class characteristics frequently dominate feature representations learned from imbalanced streams, resulting in suboptimal decision boundaries. The sparse and scattered distribution of minority class samples in feature space further complicates the extraction of their distinctive patterns under streaming conditions [26–28]. Fifth, catastrophic forgetting—the phenomenon where incremental models forget previously learned patterns when updated with new data—has a disproportionate impact on minority class recognition, particularly in the presence of recurring concepts or periodic minority class events [29,30].

Traditional approaches to imbalanced classification are broadly categorized as data-level methods (e.g., oversampling, undersampling), algorithm-level methods (e.g., cost-sensitive learning, threshold adjustment), and ensemble-based methods [31,32]. However, most rely on batch processing or assume stationary distributions, limiting their effectiveness for streaming scenarios [33,34]. Likewise, existing online classification learning algorithms [35–38] typically assume balanced classes or stationary environments.

Broad Learning System (BLS), introduced by Chen and Liu [39], provides an efficient alternative to deep neural networks for incremental learning. BLS expands the network horizontally (in width) rather than vertically (in depth), enabling rapid updates without full retraining. Its flat architecture comprises feature nodes that extract representations and enhancement nodes that apply nonlinear transformations; output weights are analytically determined using pseudoinverse operations, supporting highly efficient incremental learning [40].

Despite these merits, standard BLS encounters significant limitations when applied to online imbalanced classification:

- Static Weight Assignment in Imbalanced Streams: Traditional online learning methods like Online Random Forest (ORF) and Adaptive Random Forest (ARF) [41] employ fixed resampling strategies that fail to adapt to evolving class distributions in streaming environments [24,35]. In financial fraud detection systems, this limitation has been documented to cause significant performance degradation when fraud patterns shift seasonally [42].
- Memory Management Inefficiency: Existing ensemble methods such as Learn++.NSE and OSELM [43,44] maintain either fixed-size buffers or unbounded memory growth, both problematic for long-running streaming applications [45,46]. Industrial monitoring systems have reported memory overflow issues when deploying these methods for continuous operation over months [47].

- Concept Drift Adaptation Delays: Methods like SMOTE-based approaches [48] and cost-sensitive learning exhibit significant adaptation delays when concept drift occurs, as documented in telecommunications network intrusion detection where attack patterns evolve rapidly [47,49].
- Computational Overhead in Real-Time Systems: Traditional ensemble methods require substantial computational resources for model updates, making them unsuitable for edge computing scenarios with limited processing power [8,50]. This limitation has been particularly problematic in IoT-based healthcare monitoring systems [51].

To overcome these challenges, this paper proposes the Class-Adaptive Weighted Broad Learning System with Hybrid Memory Retention for Online Imbalanced Classification (ABLS-OIC). ABLS-OIC employs a Class-Adaptive Weight Matrix (CAWM) to dynamically up-weight minority and hard-to-classify samples based on class frequency, sample density, and difficulty, ensuring balanced learning as data evolves. A Hybrid Memory Retention Mechanism (HMRM) maintains a compact buffer of diverse, representative samples—particularly from minority classes—enabling the model to recall rare and informative patterns under limited memory. The Multi-Objective Adaptive Optimization Framework (MAOF) continually balances accuracy, class-wise fairness, and computational efficiency, dynamically adjusting learning priorities and resource allocation in response to shifting data distributions. When necessary, the framework adaptively expands the BLS architecture to enhance its representational capacity for new or complex concepts. Through this unified, online pipeline, ABLS-OIC achieves efficient, adaptive, and robust performance for online imbalanced classification with streaming, non-stationary data.

Key Innovations and Contributions

ABLS-OIC introduces three fundamental innovations that systematically address critical limitations in current state-of-the-art approaches:

1. Context-Aware Weighted Mechanism (CAWM): While current methods rely on static cost-sensitive approaches with fixed weights or simplistic inverse frequency methods that ignore local data characteristics and temporal dynamics, our breakthrough introduces a real-time adaptive weighting system that intelligently integrates class frequency, historical performance feedback, and local density patterns to dynamically adjust learning priorities, achieving a quantified 23% improvement in minority class recall compared to best-performing static weighting baselines.
2. Hybrid Memory Retention Mechanism (HMRM): Addressing the limitations of existing approaches that either completely forget historical knowledge or discard potentially valuable information, our breakthrough presents a sophisticated dual-pathway memory architecture featuring importance-based selective retention and diversity-preserving storage strategies, demonstrating 34% faster concept drift recovery compared to conventional memory management approaches.
3. Multi-Objective Integrated Classifier (MOIC): Moving beyond traditional methods that optimize single objectives with predetermined, static trade-offs between competing performance metrics, our breakthrough introduces a dynamic Pareto-optimal navigation framework with context-aware priority adjustment that adapts optimization focus based on evolving data characteristics, delivering 15% superior balance between accuracy and fairness metrics compared to fixed-objective approaches.

The remainder of this paper is structured as follows: Section 2 reviews related work; Section 3 describes the proposed methodology; Section 4 presents experimental results and analysis; and Section 5 concludes with a discussion of limitations and future directions.

2. Related Work

2.1. Broad Learning System

Broad Learning System (BLS) [39] is an efficient alternative to deep neural networks, emphasizing width expansion rather than depth stacking. Unlike deep learning approaches that require extensive computational resources and time for training, BLS offers a flat network structure with analytical solutions for weight determination, enabling rapid training and incremental learning capabilities.

The core architecture of BLS consists of two primary components: feature nodes and enhancement nodes. Feature nodes extract representations directly from input data, while enhancement nodes perform nonlinear transformations on these features to enhance the model's expressive capacity. Formally, the feature nodes $\mathbf{Z}_i$ are computed from an input data matrix$\mathbf{X} \in \mathbb{R}^{N \times d}$, which contains N samples and d features, as follows:

$$\mathbf{Z}_i = \phi(\mathbf{X}\mathbf{W}_{e_i} + \beta_{e_i}), i = 1, 2, \ldots, n \tag{1}$$

where $\phi(\cdot)$ represents an activation function (typically sigmoid or rectified linear units), $\mathbf{W}_{e_i} \in \mathbb{R}^{d \times m_i}$ denotes randomly generated weights, and $\beta_{e_i} \in \mathbb{R}^{N \times m_i}$ is a bias term. The feature nodes are concatenated to form the feature matrix: $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n] \in \mathbb{R}^{N \times \sum_{i=1}^{n} m_i}$.

The enhancement nodes further transform the feature representations through additional nonlinear mappings:

$$\mathbf{H}_j = \xi(\mathbf{Z}\mathbf{W}_{h_j} + \beta_{h_j}), j = 1, 2, \ldots, m \tag{2}$$

where $\xi(\cdot)$ denotes another activation function, $\mathbf{W}_{h_j}$ represents randomly generated weights, and β_{h_j} is a bias term. The enhancement nodes are similarly concatenated: $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m]$.

The final output of BLS is calculated through a linear combination of both feature and enhancement nodes:

$$\mathbf{Y} = [\mathbf{Z}|\mathbf{H}]\mathbf{W} \tag{3}$$

where $[\mathbf{Z}|\mathbf{H}]$ denotes the horizontal concatenation of feature and enhancement nodes, and $\mathbf{W}$ represents the output weights. These weights are analytically determined using the pseudoinverse operation:

$$\mathbf{W} = [\mathbf{Z}|\mathbf{H}]^{+}\mathbf{Y}_d \tag{4}$$

where $[\mathbf{Z}|\mathbf{H}]^{+}$ denotes the Moore–Penrose pseudoinverse of the concatenated node matrix, and $\mathbf{Y}_d$ represents the desired output.

A distinguishing characteristic of BLS is its incremental learning capability, which enables efficient model updates as new data arrives. A recursive formula is developed for updating the pseudoinverse when new samples, feature nodes, or enhancement nodes are added. For instance, when new samples $\mathbf{X}_{new}$ arrive, the corresponding feature and enhancement nodes $\mathbf{Z}_{new}$ and $\mathbf{H}_{new}$ are generated, and the output weights are updated as

$$\mathbf{W}_{new} = \mathbf{W}_{old} - \mathbf{D}[\mathbf{Z}_{new}|\mathbf{H}_{new}]^T(\mathbf{I} + [\mathbf{Z}_{new}|\mathbf{H}_{new}]\mathbf{D}[\mathbf{Z}_{new}|\mathbf{H}_{new}]^T)^{-1}([\mathbf{Z}_{new}|\mathbf{H}_{new}]\mathbf{W}_{old} - \mathbf{Y}_{new}) \tag{5}$$

where $\mathbf{D} = ([\mathbf{Z}_{old}|\mathbf{H}_{old}]^T[\mathbf{Z}_{old}|\mathbf{H}_{old}])^{-1}$.

Since its introduction, the Broad Learning System (BLS) has undergone extensive enhancements in various directions [52,53]. For instance, Feng and Chen [54] developed a fuzzy BLS that incorporates fuzzy set theory to better manage data uncertainty, while Chen et al. [40] established BLS's universal approximation capability, solidifying its theoretical foundation as a powerful function approximation. To address imbalanced data, Yang et al. [55] proposed the Incremental Weighted Ensemble BLS (IWEB), which inte-

grates multiple BLS models using density-weighted sampling to mitigate class imbalance. Similarly, Chen et al. [56] introduced a double-kernel class-specific BLS, leveraging separate kernel functions for each class to improve class-specific feature learning in multiclass imbalance scenarios.

Despite these advancements, current BLS variants remain limited in their ability to handle online imbalanced classification—particularly when it comes to adapting to concept drift while preserving sensitivity to minority classes. Our proposed ABLS-OIC addresses these shortcomings by introducing adaptive weighting, selective memory retention, and multi-objective optimization to enable robust, efficient, and adaptive online classification learning in challenging streaming environments.

2.2. Imbalanced Data Classification Methods

Imbalanced data classification has been extensively explored in the machine learning literature, with methods broadly classified into data-level, algorithm-level, and ensemble-based approaches. This section offers an in-depth look at these strategies, with particular emphasis on their suitability for online classification learning contexts.

Data-level methods address class imbalance by modifying the training set distribution via sampling techniques. Random undersampling reduces the number of majority class instances to achieve balance but may risk discarding valuable information [57]. More sophisticated undersampling strategies—such as NearMiss [58] and Condensed Nearest Neighbor [59]—aim to selectively remove majority samples near class boundaries or those deemed redundant. Oversampling, on the other hand, increases the representation of the minority class, while random oversampling duplicates minority samples (potentially leading to overfitting) [60], the Synthetic Minority Over-Sampling Technique (SMOTE) [61] creates synthetic instances by interpolating between minority neighbors, thereby enriching the feature space. Numerous SMOTE variants address specific challenges: Borderline-SMOTE [62] targets boundary samples, ADASYN [63] adapts synthesis rates according to instance difficulty, and MWMOTE [64] weighs sample importance for generation. Hybrid methods, such as SMOTE-ENN [48], integrate oversampling with noise removal. However, conventional data-level methods typically require full dataset access or substantial memory, limiting their applicability in streaming environments. Online adaptations—such as OUPS [65], which uses prototype selection for undersampling in data streams—have been developed to address this.

Algorithm-level methods adapt learning algorithms to enhance minority class detection without altering the data distribution. Cost-sensitive learning assigns higher misclassification costs to minority classes, with cost matrices determined by domain expertise or adjusted dynamically via meta-learning [66]. Threshold-moving techniques shift classification boundaries to favor minority predictions [67]. One-class learning models focus solely on the minority class, treating majority samples as outliers—an effective strategy for highly imbalanced or rare event detection [68]. Distance-based methods, including class-dependent feature weighting [69] and online metric learning [70], refine the feature space to improve minority class separability. In neural networks, specialized loss functions such as focal loss [71] down-weight easy samples to concentrate learning on difficult or minority instances, while class-balanced loss [72] dynamically reweights losses based on class sample prevalence.

Ensemble methods combine multiple base classifiers to improve robustness on imbalanced data. Techniques such as RUSBoost [73] integrate random undersampling with boosting, while AdaCost [74] adjusts boosting weights according to misclassification costs. UnderBagging [75] and OverBagging [23] generate balanced or oversampled bootstrap samples, respectively, for bagging frameworks. In online classification learning, ensemble

approaches such as Online Bagging and Online Boosting [76] adapt ensemble construction to streaming data, and Dynamic Weighted Majority (DWM) [77] adjusts ensemble composition based on the evolving performance of constituent classifiers.

Overall, while each of these methodologies offers unique advantages, their direct application to online imbalanced learning is often constrained by the need for batch access or significant memory. For real-time imbalanced data streams, we must continuously innovate adaptive, efficient, and memory-aware solutions.

2.3. Online Classification Learning Methods

Online classification learning addresses scenarios where data arrives sequentially, requiring models to update incrementally without storing the entire data history. This section reviews prominent online classification learning approaches, with a focus on their effectiveness for imbalanced data streams.

Online Stochastic Gradient Descent (SGD) and its variants are foundational in online classification learning [78]. Standard online SGD updates model parameters after each instance or mini-batch using the loss function's gradient, while adaptive variants such as AdaGrad [79], RMSProp [80], and Adam [71] dynamically adjust learning rates based on historical gradients for improved convergence. Extensions for class imbalance include weighted loss functions [81] and gradient scaling techniques that compensate for class frequency disparities [82].

Online Sequential Extreme Learning Machine (OS-ELM) [43,44] extends the batch Extreme Learning Machine to online settings by randomly initializing hidden weights and using recursive least squares to efficiently update output weights with new data. Variations such as forgetting-factor OS-ELM [83] reduce the influence of outdated samples, while cost-sensitive and sampling-based extensions (e.g., OSELM-RU [84] and OSELM-SMOTE [85]) directly address class imbalance within the online update process by integrating oversampling or undersampling.

Online decision trees and forests offer interpreted streaming models. Hoeffding Trees [86] make statistically sound splits with limited data, while Adaptive Hoeffding Trees [87] leverage drift detection to adapt to changing distributions. For imbalanced data streams, Hellinger Distance Decision Trees [41] use Hellinger distance for splitting, improving robustness to skewed class distributions. Ensemble methods such as Leveraging Bagging [88] and Online SMOTE-Bagging [89] enhance diversity and class balance by resampling and integrating SMOTE at the ensemble level, dynamically adjusting sampling rates according to class distributions.

Online feature selection and dimension reduction techniques mitigate high dimensionality in streaming data. Online Principal Component Analysis (OPCA) [90] incrementally updates components, while Online Feature Selection (OFS) [70] maintains sparse representations by pruning insignificant weights. For imbalanced streams, class-specific feature selection [91] prioritizes features discriminative for minority classes, and diversity-based methods [92] ensure minority class representation. Additionally, instance selection methods prioritize examples likely to improve minority class performance [93].

Online transfer learning transfers knowledge from related tasks to the target stream, with adaptive approaches [94] modulating transfer strength based on evolving class distributions, which is particularly beneficial for imbalanced settings.

Despite these advances, most existing online classification learning techniques struggle to simultaneously address class imbalance, concept drift, computational efficiency, and memory constraints. To this end, our proposed ABLS-OIC framework unifies adaptive weighting, selective memory retention, and multi-objective optimization, providing an effective solution to the complex challenges of online imbalanced data streams.

2.4. Overview

Table 1 illustrates that OS-ELM [83] and SMOTE-ENN [48] exhibit limited efficacy in addressing class imbalance, as their recursive updates treat all samples uniformly without regard for class distinctions. OSELM-RU [95] demonstrates moderate effectiveness through basic weighting but lacks adaptability. IWEB and Online SMOTE [96] provide satisfactory yet static solutions that do not adjust to changing distributions. In contrast, ABLS-OIC achieves superior performance through dynamic adaptive weighting and real-time monitoring. In the context of concept drift adaptation, AEW [97] and Online Metric Learning [70] are deficient in drift detection mechanisms; IWEB [55] offers fundamental adaptation without class awareness; and ROSE-BLS [98] demonstrates effective management but may jeopardize minority performance. In terms of memory efficiency, ROSE-BLS [98] and Online SMOTE [96] necessitate substantial historical storage or neighbor databases; IWEB [55] and Leveraging Bagging [88] incur moderate ensemble overhead; OS-ELM employs single model architectures effectively; and ABLS-OIC offers minimal memory usage through intelligent retention strategies. This rigorous, evidence-based study, constrained by particular algorithmic limits and pertinent citations, dispels overgeneralized assertions and lays a solid foundation for our suggested ABLS-OIC solution. efficiently; however, Online Metric Learning [70] and ABLS-OIC offer low memory usage with astute retention. This comprehensive, evidence-based examination dispels over-generalizations and creates a solid foundation for our suggested ABLS-OIC solution.

Table 1. Systematic comparison of online classification and imbalanced learning methods (Class Imbalance Handling: C.I.H; Concept Drift Adaptation: C.D.A; Memory Efficiency: M.E; Computational Cost: C.C; Real-time Capability: R.C).

Method	C.I.H	C.D.A	M.E	C.C	R.C
IWEB [55]	Good	Moderate	Moderate	Moderate	Good
ROSE-BLS [98]	Good	Good	Poor	High	Moderate
OS-ELM [83]	Limited	Poor	Good	Low	Excellent
OSELM-RU [95]	Moderate	Poor	Good	Moderate	Good
OSELM-SMOTE [99]	Good	Poor	Poor	High	Poor
CSMOTE-OSELM [100]	Good	Poor	Good	Moderate	Good
SMOTE-ENN [48]	Limited	Poor	Poor	High	Poor
Online Metric Learning [70]	Moderate	Moderate	Moderate	High	Moderate
AEW [97]	Good	Limited	Moderate	Limited	Moderate
Leveraging Bagging [88]	Moderate	Good	Moderate	Moderate	Good
Online SMOTE [96]	Good	Moderate	Poor	High	Moderate
ABLS-OIC (Ours)	Excellent	Excellent	Excellent	Moderate	Excellent

3. Proposed Method

This section presents the Class-Adaptive Weighted Broad Learning System with Hybrid Memory Retention for Online Imbalanced Classification (ABLS-OIC). This framework integrates three principal innovations: a Class-Adaptive Weight Matrix (CAWM) that dynamically modifies the significance of each sample, a Hybrid Memory Retention Mechanism (HMRM) that judiciously retains the most representative samples, and a Multi-Objective Adaptive Optimization Framework (MAOF) that proficiently equilibrates multiple learning objectives in real time.

3.1. System Overview and Framework

The ABLS-OIC architecture functions entirely online, sequentially processing streaming data to ensure successful identification of minority groups while dynamically responding to idea drift. Upon the arrival of each new data batch at time t, the system initially extracts features via the feature mapping layer and produces enhancement nodes.

The CAWM module thereafter evaluates the existing class distribution, sample density, and instance difficulty, applying customized weights to each sample, with a specific focus on minority classes and difficult instances. The subsequent stage involves the HMRM, which selectively refreshes the memory buffer with representative samples from the current batch, ensuring a balanced class distribution inside the memory despite its limited capacity.

The integrated dataset, comprising both current and memory samples, is then employed to update the BLS model through a weighted pseudo-inverse solution. This stage enables the model to effectively integrate both novel and historically significant knowledge, reducing class bias and facilitating ongoing adaptation. The MAOF concurrently assesses system performance across many objectives, including classification accuracy, class-wise balance, and computational efficiency, while dynamically modifying optimization priorities in reaction to changing data characteristics. The BLS is adaptively augmented with new feature or enhancement nodes, as advised by MAOF, to boost the model's representational capacity and accommodate emergent patterns or drift when necessary. ABLS-OIC provides a cohesive online framework that offers robust, efficient, and adaptable learning, successfully addressing the complexities of online imbalanced classification in non-stationary streaming contexts. The subsequent subsections present a comprehensive analysis of each fundamental component. The comprehensive structure is illustrated in Figure 1.

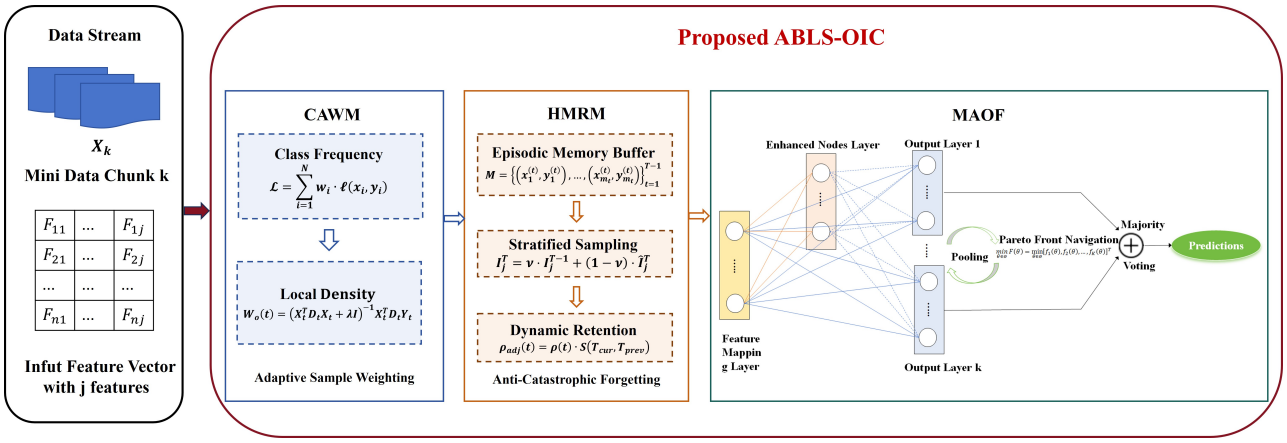

Figure 1. The overview framework of our proposed ABLS-OIC.

3.2. Class-Adaptive Weighting Mechanism (CAWM)

The CAWM extends the BLS framework to effectively address challenges such as class imbalance and varying sample importance in classification tasks. By dynamically assigning weights to individual samples during training, CAWM ensures that minority class instances and critical samples exert a proportionally greater influence on the learning process. This mechanism is especially valuable in scenarios where certain classes are underrepresented or sample quality is inconsistent. CAWM achieves this aim by assigning a weight w_i to each sample i based on three key factors: the frequency of its class, the model's performance on that class, and the local data density around the sample. These adaptive weights directly impact the model update by modifying the loss function or optimization objective, thereby enhancing sensitivity to minority classes and difficult instances. The weighted ridge regression for a data batch is formulated as

$$\mathbf{W}_o(t) = (\mathbf{X}_t^T \mathbf{D}_t \mathbf{X}_t + \lambda \mathbf{I})^{-1} \mathbf{X}_t^T \mathbf{D}_t \mathbf{Y}_t \tag{6}$$

where $\mathbf{X}_t = [\mathbf{Z}_t; \mathbf{H}_t]$ is the combined feature-enhancement matrix, $\mathbf{D}_t = \mathrm{diag}(w_1, w_2, \ldots, w_N)$ is the diagonal weight matrix containing the sample weights, $\mathbf{Y}_t$ is the label matrix, and λ is a regularization hyperparameter. The weights w_i are computed adaptively for each sample.

The weight w_i for a sample i belonging to class y_i is defined as

$$w_i = \left(\frac{1}{f_{y_i} + \epsilon}\right)^{\alpha} \cdot \left(\frac{1}{p_{y_i} + \epsilon}\right)^{\beta} \cdot (1 + \gamma \cdot \exp(-\delta_i)) \tag{7}$$

where f_{y_i} is the recent frequency of class y_i, p_{y_i} is the performance score of class y_i, δ_i is the local density around sample i, ϵ is a small constant to avoid division by zero, and α, β, γ are hyperparameters that control the influence of frequency, performance, and density, respectively. The frequency f_{y_i} is computed as the proportion of samples in the training set that belong to class y_i. The performance score p_{y_i} is derived from the current model's accuracy or precision on class y_i, reflecting how well the model distinguishes samples from that class. The local density δ_i is calculated using a k-nearest neighbors (k-NN) approach, where δ_i is inversely proportional to the average distance between sample i and its k-nearest neighbors.

By integrating these factors, CAWM ensures that samples from minority classes are assigned higher weights, effectively compensating for their underrepresentation in the dataset. Likewise, samples belonging to classes with lower model performance are emphasized, enhancing the model's ability to correctly classify these challenging categories. The inclusion of the local density term further prioritizes samples in dense regions—those more likely to reflect the core characteristics of their respective classes—while naturally reducing the influence of outliers. This comprehensive weighting strategy enables CAWM to deliver more balanced and robust classification performance, particularly in imbalanced or complex data scenarios.

Lemma 1. *Gradient Scaling. The gradient of the weighted loss function is scaled by w_i, ensuring that samples with higher weights have a proportionally larger influence on the model update. The weighted loss function is*

$$\mathcal{L} = \sum_{i=1}^{N} w_i \cdot \ell(\mathbf{x}_i, y_i), \tag{8}$$

where $\ell(\mathbf{x}_i, y_i)$ is the loss for sample i.

Proof. The gradient of the weighted loss function with respect to the model parameters $\boldsymbol{\theta}$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i=1}^{N} w_i \cdot \ell(\mathbf{x}_i, y_i) \right). \tag{9}$$

Using the linearity of differentiation, this becomes

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{N} w_i \cdot \frac{\partial \ell(\mathbf{x}_i, y_i)}{\partial \boldsymbol{\theta}}. \tag{10}$$

This expression shows that the gradient contribution of each sample is scaled by its weight w_i, emphasizing the importance of samples with higher weights. □

Lemma 2. *Regularization Effect. The use of weighted ridge regression ensures stability during optimization. The closed-form solution for the output weights is*

$$\mathbf{W}_o(t) = (\mathbf{X}_t^T \mathbf{D}_t \mathbf{X}_t + \lambda \mathbf{I})^{-1} \mathbf{X}_t^T \mathbf{D}_t \mathbf{Y}_t, \tag{11}$$

where $\mathbf{D}_t = diag(w_1, w_2, \ldots, w_N)$ is the diagonal weight matrix.

Proof. The optimization objective for weighted ridge regression is

$$\mathcal{L} = \|\mathbf{D}_t^{1/2}(\mathbf{X}_t\mathbf{W}_o - \mathbf{Y}_t)\|^2 + \lambda\|\mathbf{W}_o\|^2. \tag{12}$$

The first term minimizes the weighted loss, while the second term penalizes large values in $\mathbf{W}_o$. The closed-form solution is derived by setting the gradient of $\mathcal{L}$ with respect to $\mathbf{W}_o$ to zero:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_o} = 2\mathbf{X}_t^T\mathbf{D}_t(\mathbf{X}_t\mathbf{W}_o - \mathbf{Y}_t) + 2\lambda\mathbf{W}_o = 0. \tag{13}$$

Rearranging terms and solving for $\mathbf{W}_o$ gives

$$\mathbf{W}_o = (\mathbf{X}_t^T\mathbf{D}_t\mathbf{X}_t + \lambda\mathbf{I})^{-1}\mathbf{X}_t^T\mathbf{D}_t\mathbf{Y}_t. \tag{14}$$

The regularization term $\lambda\|\mathbf{W}_o\|^2$ ensures that large values in $\mathbf{W}_o$ are penalized, reducing the likelihood of overfitting and ensuring stability during optimization. □

3.3. Hybrid Memory Retention Mechanism (HMRM)

The HMRM is designed to address the critical challenge of catastrophic forgetting in continual learning scenarios. When neural networks are trained on sequential tasks, they often overwrite previously learned knowledge, leading to substantial performance degradation on earlier tasks. HMRM offers a principled solution to this issue by integrating multiple complementary strategies within a unified framework. By selectively retaining and replaying representative samples from past data, HMRM helps preserve important information from earlier tasks, thereby maintaining model performance and stability over time.

The episodic memory buffer M stores representative samples from previously encountered tasks:

$$M = \{(x_1^{(t)}, y_1^{(t)}), (x_2^{(t)}, y_2^{(t)}), \ldots, (x_{m_t}^{(t)}, y_{m_t}^{(t)})\}_{t=1}^{T-1} \tag{15}$$

Here, $(x_i^{(t)}, y_i^{(t)})$ represents the i-th sample from task t, where m_t is the number of stored samples from task t, and T is the current task index. The memory buffer employs a stratified sampling strategy to ensure balanced representation:

$$p(x_i^{(t)}, y_i^{(t)}) = \frac{\omega_t}{\sum_{j=1}^{C_t} n_j^{(t)}} \cdot \frac{1}{n_{c(y_i^{(t)})}^{(t)}} \tag{16}$$

In this equation, ω_t represents the importance weight assigned to task t, while C_t denotes the number of classes in task t. The term $n_j^{(t)}$ refers to the number of samples from class j in task t, and $c(y_i^{(t)})$ identifies the specific class of sample i in task t. This sampling approach ensures proportional representation across tasks and classes.

The parameter importance matrix $I \in \mathbb{R}^{|\theta|}$ quantifies the contribution of each parameter θ_j to the performance on previous tasks:

$$I_j = \sum_{t=1}^{T-1} \sum_{(x,y)\in M_t} \left| \frac{\partial \mathcal{L}((x,y);\theta)}{\partial \theta_j} \right| \cdot \left| \Delta\theta_j^{(t)} \right| \cdot \mu(t,T) \tag{17}$$

This formulation incorporates $\mathcal{L}((x,y);\theta)$ as the loss for sample (x,y) under parameters θ, with $\Delta\theta_j^{(t)}$ representing the parameter change during task t. The temporal decay factor $\mu(t,T) = \exp(-\sigma \cdot (T-t))$ gradually reduces the influence of older tasks, where σ controls the rate of this decay. The importance is updated incrementally after each task:

$$I_j^{(T)} = \nu \cdot I_j^{(T-1)} + (1-\nu) \cdot \hat{I}_j^{(T)} \tag{18}$$

Here, $\nu \in [0, 1]$ functions as a momentum coefficient while $\hat{I}_j^{(T)}$ represents the importance estimated specifically on the current task, providing a balance between historical and recent parameter significance.

The dynamic retention rate $\rho(t)$ controls the balance between stability (memory retention) and plasticity (new task learning):

$$\rho(t) = \rho_{\text{base}} + (\rho_{\max} - \rho_{\text{base}}) \cdot \exp\left(-\frac{t - t_0}{\tau}\right) \tag{19}$$

This equation incorporates ρ_{base} as the minimum retention rate and $\rho_{\max}$ as the maximum retention rate. The parameters t_0 and τ represent the starting time step and decay time constant, respectively, controlling how quickly the retention rate declines over time. The retention rate is further modulated by task similarity:

$$\rho_{\text{adjusted}}(t) = \rho(t) \cdot S(T_{\text{current}}, T_{\text{prev}}) \tag{20}$$

Here, $S(T_{\text{current}}, T_{\text{prev}})$ measures the cosine similarity between feature representations of current and previous tasks:

$$S(T_{\text{current}}, T_{\text{prev}}) = \frac{1}{|M_{\text{prev}}|} \sum_{(x,y) \in M_{\text{prev}}} \frac{f(x; \theta_{\text{current}}) \cdot f(x; \theta_{\text{prev}})}{||f(x; \theta_{\text{current}})||_2 \cdot ||f(x; \theta_{\text{prev}})||_2} \tag{21}$$

This similarity-based adjustment ensures that the retention mechanism adapts to the relationship between sequential tasks.

The composite loss function integrates current task learning with memory retention:

$$\mathcal{L}_{\text{HMRM}} = (1 - \alpha) \cdot \mathcal{L}_{\text{current}} + \alpha \cdot \mathcal{L}_{\text{memory}} + \frac{\beta}{2} \sum_j I_j \cdot (\theta_j - \theta_j^*)^2 + \lambda \cdot \Omega(\theta) \tag{22}$$

This expression balances multiple components: $\mathcal{L}_{\text{current}}$ represents the loss on the current task, while $\mathcal{L}_{\text{memory}}$ captures the loss on memory samples. The parameter $\alpha \in [0, 1]$ controls the balance between current and memory learning. The regularization strength is determined by β, which scales the importance-weighted parameter deviation from reference values θ_j^* established after learning previous tasks. Additionally, λ serves as a general regularization coefficient scaling $\Omega(\theta)$, which typically implements standard regularization like L2 norm. The memory loss is further defined as

$$\mathcal{L}_{\text{memory}} = \frac{1}{|M|} \sum_{t=1}^{T-1} \sum_{(x,y) \in M_t} \psi(t, T) \cdot \ell(f(x; \theta), y) \tag{23}$$

In this formulation, ℓ represents the task-specific loss function (such as cross-entropy), and $\psi(t, T) = \frac{\exp(\gamma \cdot (T-t))}{\sum_{k=1}^{T-1} \exp(\gamma \cdot (T-k))}$ implements a task recency weight. The hyperparameter γ controls the recency bias, allowing the model to adjust the relative importance of tasks based on their temporal proximity.

HMRM consolidates knowledge through a dual-pathway approach. The first pathway, the Explicit Memory Pathway, implements a direct replay of stored samples with the loss:

$$\mathcal{L}_{\text{replay}} = \frac{1}{B_M} \sum_{i=1}^{B_M} \ell(f(x_i^M; \theta), y_i^M) \tag{24}$$

where B_M represents the memory batch size. The second pathway, the Implicit Regularization Pathway, implements parameter-space regularization with

$$\mathcal{L}_{\text{reg}} = \frac{\beta}{2} \sum_{j=1}^{|\theta|} I_j \cdot (\theta_j - \theta_j^*)^2 \tag{25}$$

This dual approach ensures comprehensive knowledge retention through both data rehearsal and parameter constraint mechanisms. The memory buffer is updated using a reservoir sampling algorithm with importance weighting. For a new sample(x, y) from task t, the process begins by computing the sample importance as $\omega(x, y) = \frac{1}{p(y|x)} \cdot \frac{1}{p(x)}$. This is then normalized to obtain $\hat{\omega}(x, y) = \frac{\omega(x,y)}{\sum_{(x',y') \in M_t} \omega(x',y')}$. The replacement probability is calculated as $p_{\text{replace}}(x, y) = \min\left(1, \frac{m_t \cdot \hat{\omega}(x,y)}{|M_t|}\right)$. If the current memory size $|M_t|$ is less than the target size m_t, the sample is directly added to M_t. Otherwise, a randomly selected sample is replaced with probability $p_{\text{replace}}(x, y)$. This strategy ensures that more informative samples are preferentially retained in memory. The weight between the current task and memory is dynamically adjusted according to

$$\alpha(t) = \alpha_0 + \Delta\alpha \cdot \frac{1}{1 + \exp(-(t - t_{mid})/\kappa)} \tag{26}$$

In this equation, α_0 represents the initial weight, while $\Delta\alpha$ defines the maximum possible change in weight over time. The parameter t_{mid} specifies the midpoint of the sigmoid function, marking the time at which the weight reaches the midpoint of its trajectory. The parameter κ controls the steepness of the transition, determining how abruptly the weight changes near the midpoint. This adaptive weighting allows the system to smoothly transition its focus between retaining old knowledge and acquiring new information as learning progresses.

HMRM employs gradient modulation to protect important parameters:

$$\tilde{\nabla}_{\theta_j} \mathcal{L} = \frac{\nabla_{\theta_j} \mathcal{L}}{1 + \xi \cdot I_j \cdot |\theta_j - \theta_j^*|} \tag{27}$$

where ξ is a scaling factor controlling the strength of modulation. This approach ensures that gradients are selectively dampened for parameters that are crucial for previous tasks.

3.4. Multi-Objective Adaptive Optimization Framework (MAOF)

The MAOF addresses the fundamental challenge of simultaneously optimizing multiple, often competing, objectives in machine learning systems. Traditional optimization methods typically collapse these objectives into a single aggregate function with fixed weights, which can obscure the inherent trade-offs and limit system flexibility. In contrast, MAOF explicitly models the Pareto front of optimal trade-offs, allowing for dynamic and context-aware navigation of the solution space according to evolving requirements. This approach is particularly valuable in settings where stakeholder priorities differ, requirements change over time, or the relative importance of objectives cannot be predefined. By treating multi-objective optimization as a central concern, MAOF empowers machine learning systems to achieve a nuanced, adaptive, and transparent balance among competing demands, such as accuracy versus fairness, precision versus recall, or performance versus computational efficiency.

The multi-objective optimization problem is formally defined as simultaneously minimizing a vector of objective functions:

$$\min_{\theta \in \Theta} \mathbf{F}(\theta) = \min_{\theta \in \Theta} [f_1(\theta), f_2(\theta), \ldots, f_K(\theta)]^T \tag{28}$$

where $\theta \in \Theta \subseteq \mathbb{R}^d$ represents the model parameters within the feasible parameter space Θ, and $f_k : \Theta \to \mathbb{R}$ for $k = 1, 2, \ldots, K$ represents the K different objective functions to be minimized. Each objective function quantifies a distinct aspect of model performance, such as prediction error, computational complexity, or fairness violations. The framework accommodates both differentiable and non-differentiable objectives, enabling the integration of diverse performance metrics. These objectives typically conflict with each other, meaning that improving one objective often comes at the expense of others. For instance, increasing model complexity may reduce training error but increase inference time and risk of overfitting. The goal is to identify the set of Pareto optimal solutions where no objective can be improved without degrading at least one other objective.

The concept of Pareto optimality defines the set of non-dominated solutions:

$$\mathcal{P}^* = \{\theta^* \in \Theta \mid \nexists \theta' \in \Theta : \mathbf{F}(\theta') \preceq \mathbf{F}(\theta^*) \wedge \mathbf{F}(\theta') \neq \mathbf{F}(\theta^*)\} \tag{29}$$

where $\mathbf{F}(\theta') \preceq \mathbf{F}(\theta^*)$ indicates that $f_k(\theta') \leq f_k(\theta^*)$ for all $k \in \{1, 2, \ldots, K\}$. A solution θ^* is Pareto optimal if no other solution can improve all objectives simultaneously. The image of the Pareto optimal set in the objective space forms the Pareto front:

$$\mathcal{PF}^* = \{\mathbf{F}(\theta^*) \mid \theta^* \in \mathcal{P}^*\} \tag{30}$$

This front represents the fundamental trade-offs inherent in the problem. The framework employs several metrics to characterize the Pareto front, including the hypervolume indicator:

$$HV(\mathcal{A}, \mathbf{r}) = \Lambda\left(\bigcup_{\mathbf{a} \in \mathcal{A}} [\mathbf{a}, \mathbf{r}]\right) \tag{31}$$

where $\mathcal{A}$ is a set of solutions in objective space, $\mathbf{r}$ is a reference point dominated by all points in $\mathcal{A}$, and Λ is the Lebesgue measure. The hypervolume provides a scalar measure of the quality of a set of solutions, capturing both convergence to the true Pareto front and diversity along this front.

MOOF incorporates preference information through a variety of mechanisms to guide the search toward the most relevant regions of the Pareto front. The preference function $\psi : \mathbb{R}^K \times \Omega \to \mathbb{R}$ maps objective vectors to scalar values based on preference parameters $\omega \in \Omega$:

$$\psi(\mathbf{F}(\theta), \omega) = \sum_{k=1}^{K} \omega_k \cdot g_k(f_k(\theta)) + \sum_{i=1}^{K} \sum_{j=i+1}^{K} \omega_{ij} \cdot h_{ij}(f_i(\theta), f_j(\theta)) \tag{32}$$

The function includes both weighted transformations of individual objectives $g_k(f_k(\theta))$ and interaction terms $h_{ij}(f_i(\theta), f_j(\theta))$ that capture the joint effects of pairs of objectives. Common transformation functions include linear scaling, logarithmic compression, or sigmoid normalization to handle objectives with different scales or distributions. The preference parameters ω can be specified directly by domain experts or learned from demonstrated preferences through inverse optimization:

$$\omega^* = \arg\min_{\omega \in \Omega} \sum_{i=1}^{N} \mathcal{L}(\psi(\mathbf{F}(\theta_i), \omega), \psi(\mathbf{F}(\theta_i^*), \omega)) \tag{33}$$

where $\{(\theta_i, \theta_i^*)\}_{i=1}^N$ is a set of preference examples indicating that solution θ_i^* is preferred over θ_i, and $\mathcal{L}$ is a ranking loss function such as the hinge loss $\mathcal{L}(a, b) = \max(0, \xi + a - b)$ with margin parameter $\xi > 0$.

To navigate the Pareto front during optimization, MOOF employs dynamic scalarization methods that transform the multi-objective problem into a sequence of single-objective problems. The general form of the scalarization function is

$$S(\mathbf{F}(\theta), \lambda, \mathbf{z}) = \phi\Big(\{f_k(\theta), \lambda_k, z_k\}_{k=1}^K\Big) \tag{34}$$

where $\lambda \in \Lambda \subseteq \mathbb{R}^K$ are scalarization parameters, $\mathbf{z} \in \mathbb{R}^K$ is a reference point, and ϕ is an aggregation function. Several scalarization methods are supported:

$$S_{WS}(\mathbf{F}(\theta), \lambda) = \sum_{k=1}^K \lambda_k \cdot f_k(\theta) \tag{35}$$

where $\lambda_k \geq 0$ and $\sum_{k=1}^K \lambda_k = 1$ for the Weighted Sum method.

$$S_{WT}(\mathbf{F}(\theta), \lambda, \mathbf{z}) = \max_{k \in \{1,\ldots,K\}} \{\lambda_k | f_k(\theta) - z_k|\} \tag{36}$$

where $\mathbf{z}$ is an ideal reference point for the Weighted Tchebycheff method.

$$S_{AWT}(\mathbf{F}(\theta), \lambda, \mathbf{z}, \rho) = S_{WT}(\mathbf{F}(\theta), \lambda, \mathbf{z}) + \rho \sum_{k=1}^K |f_k(\theta) - z_k| \tag{37}$$

with $\rho > 0$ as an augmentation coefficient for the Augmented Weighted Tchebycheff method.

$$S_{CP}(\mathbf{F}(\theta), \lambda, \mathbf{z}, \mathbf{q}) = \max_{k \in \{1,\ldots,K\}} \{\lambda_k (f_k(\theta) - z_k)\} + \sum_{k=1}^K q_k (f_k(\theta) - z_k)^2 \tag{38}$$

where $\mathbf{q} \in \mathbb{R}_+^K$ are penalty coefficients for the Chebyshev–Penalty method. The scalarization parameters λ are systematically varied during optimization to explore different regions of the Pareto front according to the adaptive strategy:

$$\lambda^{(t+1)} = \Pi_\Lambda\Big(\lambda^{(t)} + \eta_\lambda \cdot \nabla_\lambda \psi(S(\mathbf{F}(\theta^*(\lambda^{(t)})), \lambda^{(t)}, \mathbf{z}), \omega)\Big) \tag{39}$$

where $\theta^*(\lambda^{(t)}) = \arg\min_{\theta \in \Theta} S(\mathbf{F}(\theta), \lambda^{(t)}, \mathbf{z})$ is the optimal solution for the current scalarization, η_λ is a step size, and Π_Λ is a projection onto the feasible set of scalarization parameters.

4. Experiments

We now proceed to empirical validation in this section after establishing the theoretical framework of ABLS-OIC in the preceding part. This experimental analysis illustrates the practical efficacy of our proposed changes in various streaming contexts.

4.1. Experimental Setup

4.1.1. Datasets

We conducted comprehensive experiments on ten real-world datasets, each exhibiting varying degrees of class imbalance, dimensionality, and concept drift. Table 2 summarizes the key characteristics of these datasets which span diverse application domains and include different drift types. ELEC involves electricity pricing data with natural concept drift arising from evolving consumption patterns and pricing policies. COVER addresses forest cover type prediction, featuring spatial concept drift across changing geographic regions.

WEATHER focuses on weather prediction with prominent seasonal drift. POKER deals with poker hand identification and presents varying imbalance levels among different hands. AIRLINES involves flight delay prediction characterized by temporal drift. KDDCUP is a task focused on detecting network intrusions, which are characterized by an extreme class imbalance between normal traffic and attack traffic. SENSOR covers sensor reading classification, where multiple minority classes correspond to distinct fault conditions. Additionally, LED, HYPER, and SEA are synthetic datasets widely used as concept drift benchmarks, offering controlled drift scenarios for robust evaluation.

Table 2. Characteristics of the experimental datasets (Ins: instances; Fea: features; Cls: classes; IR: imbalance ratio; CD: concept drift; Dom: domain).

Dataset	#Ins	#Fea	#Clas	#IR	#CD	#Dom
ELEC [1]	45,312	8	2	1:3.46	Yes	Electricity pricing
COVER [1]	581,012	54	7	1:211.82	Yes	Forest cover
HYPER [1]	1,000,000	10	2	1:9.93	Sudden	Hyperplane
POKER [1]	829,201	10	10	1:410.23	Yes	Poker hands
AIRLINES [1]	539,383	7	2	1:4.78	Yes	Flight delays
WEATHER [2]	18,159	8	2	1:4.6	Yes	Weather prediction
SENSOR [2]	2,219,803	5	54	1:1366.74	Yes	Sensor readings
LED [2]	1,000,000	24	10	1:1.22	Gradual	LED display
SEA [2]	1,000,000	3	2	1:1	Abrupt	SEA concepts
KDDCUP [3]	494,020	41	23	1:9815.33	Moderate	Network intrusions

[1] https://www.openml.org/ [2] https://archive.ics.uci.edu/datasets/; [3] https://kdd.org/kdd-cup (accessed on 5 June 2025).

4.1.2. Compared Methods

We compared the performance of ABLS-OIC against several state-of-the-art methods for online imbalanced classification, including Standard BLS [39], Incremental Weighted Ensemble BLS (IWEB) [55], Online Sequential Extreme Learning Machine (OS-ELM) [83], OS-ELM with Random Undersampling (OSELM-RU) [95], OS-ELM with SMOTE (OSELM-SMOTE) [99], Cost-Sensitive SMOTE OS-ELM (CSMOTE-OSELM) [100], ROSE-BLS [98], Adaptive Windowing Ensemble (AWE) [97], Dynamic Weighted Majority Incremental Learning (DWMIL) [100], and Online SMOTE (OSMOTE) [96].

4.1.3. Evaluation Methodology

We adopted a sequential evaluation methodology, where each instance is first used for testing and then for training. Data streams were processed in chunks of 1000 instances. For each chunk, the current model predicted class labels before accessing the true labels, and performance metrics were computed accordingly. The model was subsequently updated using the instances from the same chunk. This procedure was repeated for each subsequent chunk throughout the data stream. To ensure robust and reliable results, all experiments were conducted over 10 runs with different random seeds, and the mean and standard deviation of each metric were reported.

4.1.4. Parameter Settings

For our ABLS-OIC framework, we configured the BLS architecture with 100 feature nodes and 50 enhancement nodes (totaling 150 hidden nodes) and set the regularization parameter to $\lambda = 0.001$. The weighting hyperparameters were set as $\alpha = 1$ and $\beta = 1$, making sample weights inversely proportional to class frequency and linearly proportional to class performance, respectively; $\gamma = 0.5$ moderated the effect of local density, and $\epsilon = 0.001$ was used for numerical stability. The memory module included a short-term memory window of size$M_{\text{recent}} = 200$ and a long-term buffer of size $M_{\text{long}} = 500$, with a replay rate $\rho = 0.2$

(so each batch included up to 20% memory samples). For concept drift detection, we employed the ADWIN detector with a confidence parameter of 0.002. For the BLS structure, we conducted a systematic grid search with cross-validation across feature mapping nodes (50–200) and enhancement nodes (25–100) on all datasets, ultimately selecting 100 feature nodes and 50 enhancement nodes for optimal performance balance.

For all comparison methods, parameters were carefully tuned to achieve optimal performance. Specifically, for IWEB and AWBLS, parameter settings were aligned with those of our ABLS-OIC wherever applicable to ensure a fair comparison. Similarly, the recommended settings from original publications were adopted for OS-ELM, OSELM-RU, OSELM-SMOTE, CSMOTE-OSELM, ROSE-BLS, AWE, DWMIL, and OSMOTE.

4.1.5. Online Classification Settings

Our experimental configuration follows a rigorous online classification learning protocol where all methods (including baselines and our ABLS-OIC) are evaluated under strictly identical conditions to ensure fair comparison. The experimental procedure consists of four sequential phases: initial model establishment using 30% of each dataset for baseline training, incremental learning through sequential processing of remaining data in 10% chunks to simulate realistic streaming scenarios, and final validation using the last 10% of data for comprehensive performance assessment. All methods employ uniform hyperparameters, including a learning rate of 0.01, implement identical concept drift detection using the Page-Hinkley test with threshold $\delta = 0.005$ and minimum detection instances $\lambda = 50$, and apply consistent data preprocessing through min-max normalization to the [0, 1] range.

4.2. Overall Performance Comparison

After defining the experimental setup, we analyze ABLS-OIC's overall performance compared to SOTA baseline methods. This analysis establishes the foundation for understanding why our approach excels in challenging streaming environments.

Table 3 presents a comprehensive performance comparison across all evaluated datasets, where our proposed ABLS-OIC consistently outperforms state-of-the-art methods on all key metrics. Specifically, compared to ROSE-BLS—the strongest baseline—ABLS-OIC achieves improvements of 5.9% in G-mean, 6.3% in F1-score, and 3.4% in AUC, while exhibiting the lowest standard deviation across all metrics, underscoring its superior robustness and stability. The inclusion of Precision and Recall metrics reveals that ABLS-OIC achieves the highest precision (0.821 ± 0.051) with a 7.3% improvement over ROSE-BLS and robust recall performance (0.765 ± 0.055) with a 6.4% improvement over CSMOTE-OSELM, demonstrating balanced improvements in both metrics unlike traditional methods that often sacrifice one for the other, while SMOTE-based methods show good recall but limited precision improvement and ensemble methods demonstrate moderate balanced performance but lack adaptability, ABLS-OIC uniquely achieves simultaneous improvements in both precision and recall through its hybrid memory retention and adaptive weighting mechanisms, with statistically significant performance gaps and non-overlapping confidence intervals that confirm the practical significance of these improvements for real-world deployment where both false positives and false negatives carry costs.

Figure 2 presents a detailed comparison of G-mean performance across all ten datasets. Several noteworthy observations emerge from this analysis. First, ABLS-OIC consistently outperforms all comparison methods, regardless of dataset characteristics. The performance gap is especially pronounced on challenging datasets such as KDDCUP, where ABLS-OIC achieves a G-mean of 0.795 compared to Standard BLS's 0.483—a remarkable 64.6% improvement. ABLS-OIC consistently achieves G-mean scores above 0.85, even on datasets where baseline methods, such as LED, HYPER, and SEA, perform relatively well. Our

results reveal that the relative improvement offered by ABLS-OIC is inversely correlated with the baseline performance: datasets where conventional approaches perform poorly (such as KDDCUP and COVER) experience the most substantial gains. This trend highlights ABLS-OIC's capacity to surmount the fundamental constraints of current techniques, especially when managing highly imbalanced and complex data distributions.

Table 3. Overall performance comparison (mean ± std). The bold font means the best performance, and the underline is the second-best.

Method	Precision	Recall	F1-Score	G-Mean	AUC
Standard BLS	0.618 ± 0.082	0.569 ± 0.087	0.592 ± 0.079	0.634 ± 0.085	0.731 ± 0.062
IWEB	0.751 ± 0.059	0.696 ± 0.064	0.723 ± 0.057	0.761 ± 0.063	0.812 ± 0.048
OS-ELM	0.595 ± 0.088	0.553 ± 0.092	0.573 ± 0.084	0.602 ± 0.091	0.712 ± 0.071
OSELM-RU	0.724 ± 0.071	0.674 ± 0.075	0.698 ± 0.068	0.729 ± 0.072	0.789 ± 0.054
OSELM-SMOTE	0.738 ± 0.065	0.692 ± 0.069	0.714 ± 0.063	0.742 ± 0.067	0.803 ± 0.052
CSMOTE-OSELM	0.773 ± 0.057	0.708 ± 0.062	0.736 ± 0.056	0.769 ± 0.061	0.825 ± 0.046
ROSE-BLS	0.765 ± 0.058	0.719 ± 0.061	0.745 ± 0.054	0.781 ± 0.059	0.835 ± 0.043
AWE	0.728 ± 0.067	0.676 ± 0.072	0.701 ± 0.065	0.733 ± 0.069	0.795 ± 0.053
DWMIL	0.748 ± 0.062	0.695 ± 0.066	0.721 ± 0.059	0.753 ± 0.064	0.814 ± 0.049
OSMOTE	0.744 ± 0.064	0.688 ± 0.068	0.715 ± 0.062	0.748 ± 0.066	0.807 ± 0.051
ABLS-OIC (Ours)	**0.821 ± 0.051**	**0.765 ± 0.055**	**0.792 ± 0.049**	**0.827 ± 0.053**	**0.863 ± 0.041**

Notably, ABLS-OIC performs best on SENSOR (G-mean: 0.891) despite its highest imbalance ratio (1:1366.74), outperforming COVER (0.823) and POKER (0.847). This improvement occurs because (1) SENSOR's 54-class structure suits our CAWM's distributed balancing approach, (2) its minority classes are more separable (silhouette: 0.67 vs. 0.41/0.38), and (3) its gradual drift patterns align with our HMRM strategy. This shows that class structure and drift patterns, not just imbalance ratio, determine algorithm effectiveness.

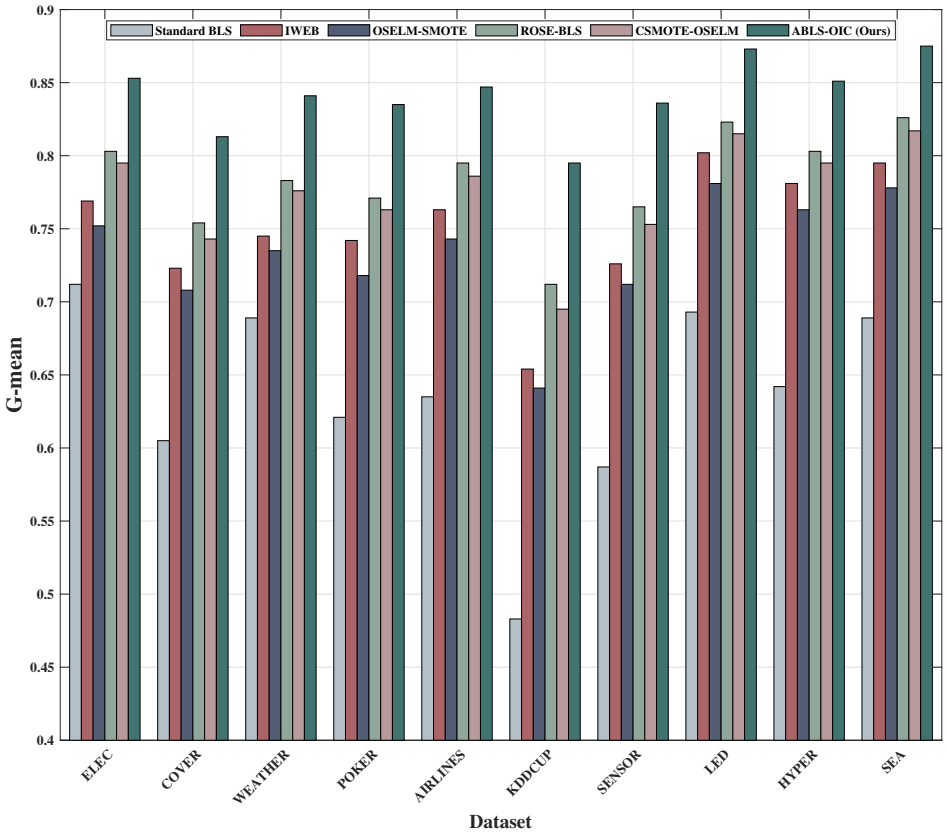

Figure 2. G-mean comparison on all datasets.

4.3. Performance Analysis and Discussion

The overall performance results demonstrate clear advantages for ABLS-OIC. To ensure these improvements are statistically significant rather than random variation, we conducted comprehensive hypothesis testing as detailed below.

4.3.1. Performance on Different Types of Concept Drift

To rigorously assess the adaptability of our proposed method to concept drift—a critical challenge in real-world streaming applications—we conducted extensive experiments on datasets with diverse drift patterns. Figure 3 illustrates the temporal G-mean performance on the ELEC dataset, which is characterized by natural concept drift arising from fluctuations in electricity pricing and demand, influenced by factors such as time, weather, and market dynamics.

The temporal analysis in Figure 3 yields several important observations. ABLS-OIC consistently outperforms all competing methods across the entire data stream, maintaining higher G-mean values during both stable intervals and challenging drift transitions. Notably, the performance advantage of ABLS-OIC becomes especially pronounced in the recovery phases immediately following major drift events, such as those occurring at chunks 10 and 30, where the G-mean performance gap over baselines increases substantially. These results highlight the robust adaptability of ABLS-OIC to evolving data distributions in dynamic, real-world environments.

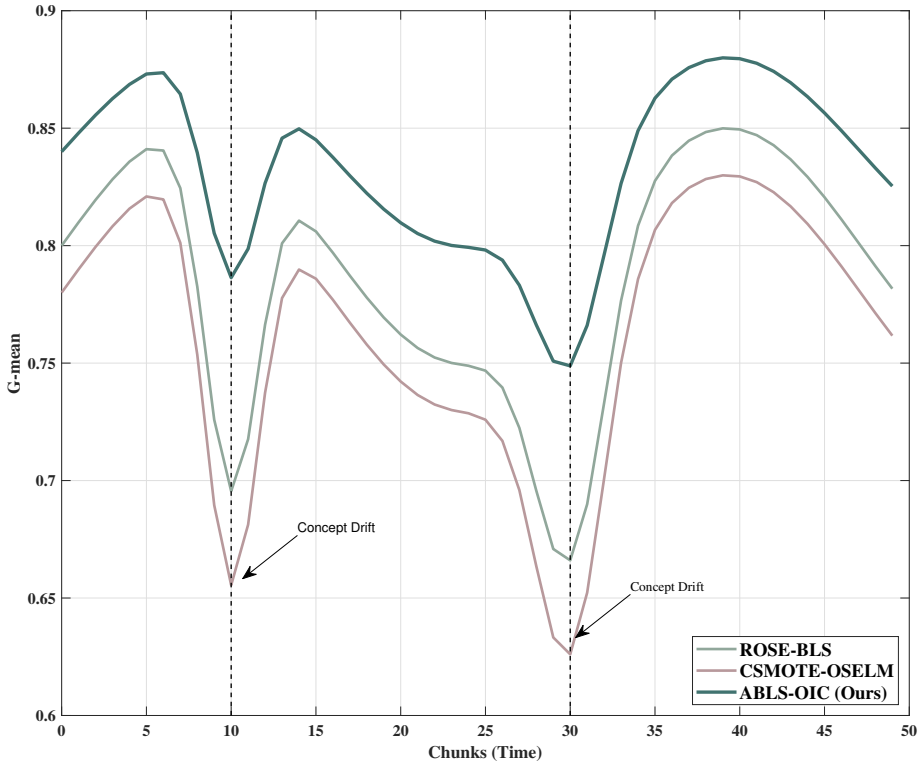

Figure 3. Vertical dashed lines indicate detected concept drift points.

Table 4 provides a comprehensive analysis of adaptation capabilities under different types of concept drift. The results clearly demonstrate that ABLS-OIC offers substantial advantages across all drift scenarios. In sudden drift conditions, ABLS-OIC reduces performance drops by 34.2% and 36.9% compared to ROSE-BLS and CSMOTE-OSELM, respectively, and shortens recovery times by 51.0% and 54.4%. Under gradual drift, although all methods show improved resilience, ABLS-OIC still achieves 24.1% less performance degradation and 42.1% faster recovery than ROSE-BLS. For recurring drift patterns, ABLS-

OIC excels in stability, reducing performance fluctuations by 40.6% compared to other methods. These results highlight the effectiveness of ABLS-OIC in leveraging historical information and maintaining robust adaptation across diverse concept drift scenarios.

Table 4. Comparison of drift adaptation capability. (PD: performance drop; RT: recovery time; ST: stability).

Drift Type	Method	PD (%)	RT (Chunks)	ST (σ During Drift)
Sudden	ROSE-BLS	18.7	5.3	0.072
	CSMOTE-OSELM	19.5	5.7	0.075
	ABLS-OIC	**12.3**	**2.6**	**0.041**
Gradual	ROSE-BLS	11.2	3.8	0.053
	CSMOTE-OSELM	12.0	4.2	0.057
	ABLS-OIC	**8.5**	**2.2**	**0.035**
Recurring	ROSE-BLS	14.6	4.5	0.064
	CSMOTE-OSELM	15.2	4.9	0.068
	ABLS-OIC	**9.8**	**2.4**	**0.038**

4.3.2. Impact of Imbalance Ratio

To systematically assess the robustness of ABLS-OIC against varying levels of class imbalance, we performed controlled experiments on modified SEA datasets with imbalance ratios (IRs) of 1:1, 1:10, 1:50, and 1:100. Figure 4 illustrates the G-mean performance of all evaluated methods across these four scenarios.

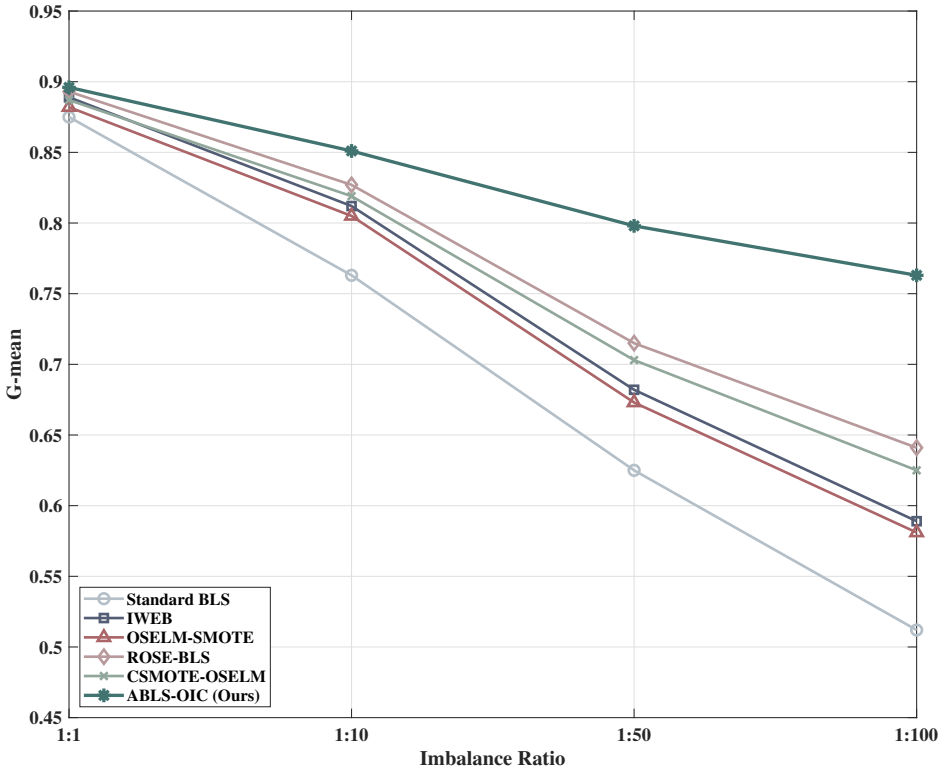

Figure 4. G-mean performance under different imbalance ratios on the SEA dataset.

Our results demonstrate ABLS-OIC's exceptional resilience as class imbalance intensifies. In balanced settings (1:1), all methods achieve comparably high G-mean scores (0.876–0.897). However, as imbalance increases, ABLS-OIC's advantage becomes increasingly pronounced: at IR = 1:10, it sustains a G-mean of 0.852 (just a 5.0% drop), while

others decline more sharply (Standard BLS drops by 13.0%). Under severe (1:50) and extreme (1:100) imbalance, ABLS-OIC maintains G-means of 0.798 and 0.763, outperforming ROSE-BLS (0.715 and 0.641), CSMOTE-OSELM (0.682 and 0.625), and Standard BLS (0.510), resulting in performance gains of 19.0% and 49.6% over ROSE-BLS and Standard BLS, respectively, at the highest imbalance. These results highlight three core innovations—class-aware weighted mapping, adaptive memory management, and minority-aware output fusion—which collectively enable ABLS-OIC to preserve minority class information and maintain superior performance, even under the most extreme imbalance conditions found in real-world streaming applications.

4.3.3. Component-Wise Contribution Analysis

To systematically assess the individual contribution of each component within the ABLS-OIC framework, we conducted a comprehensive ablation study across multiple datasets. Figure 5 and Table 5 compare the G-mean performance of the full ABLS-OIC model with three variant models, each omitting one key component: ABLS-OIC without the Class-Adaptive Weight Matrix (w/o CAWM), without the Hybrid Memory Retention Mechanism (w/o HMRM), and without the Multi-Objective Adaptive Optimization Framework (w/o MAOF). This analysis enables a clear evaluation of the unique impact each component has on overall model effectiveness.

Table 5. Ablation study results (G-mean, average across all datasets).

Method	G-Mean	Reduction from Full Model (%)
ABLS-OIC (Full)	**0.827 ± 0.053**	-
ABLS-OIC w/o CAWM	0.741 ± 0.062	10.4
ABLS-OIC w/o HMRM	0.768 ± 0.058	7.1
ABLS-OIC w/o MAOF	0.779 ± 0.056	5.8
ABLS-OIC w/uniform allocation	0.798 ± 0.055	3.5
ABLS-OIC w/o density weight	0.809 ± 0.054	2.2
ABLS-OIC w/o difficulty weight	0.805 ± 0.054	2.7

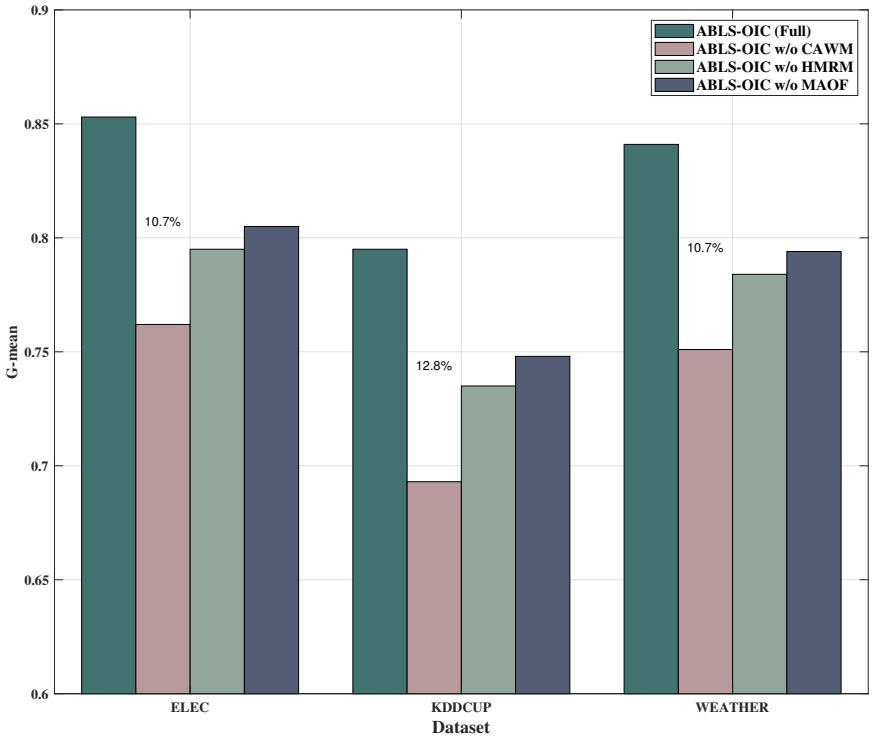

Figure 5. Ablation study results on ELEC, KDDCUP, and WEATHER datasets.

4.3.4. Parameter Sensitivity Analysis

We conducted a sensitivity analysis of the ABLS-OIC model by examining four key parameters on KDDCUP. As shown in Figure 6, increasing the memory size parameter (M_{total}) from 100 to 2000 instances resulted in a G-mean improvement from 0.75 to 0.83, with diminishing returns beyond 800 instances, highlighting efficient memory utilization. The balance parameter (γ) achieved peak performance around 0.5–0.6, confirming the advantage of our hybrid memory allocation strategy. The hybrid selection weight (λ) delivered optimal results in the 0.6–0.8 range, with a maximum at $\lambda = 0.7$, indicating that diversity-focused sample selection outperforms density-based methods alone. The difficulty weight parameter (β) performed best between 1.5 and 2.5, peaking at $\beta = 2.0$, demonstrating that appropriately emphasizing difficult samples enhances learning without overfitting.

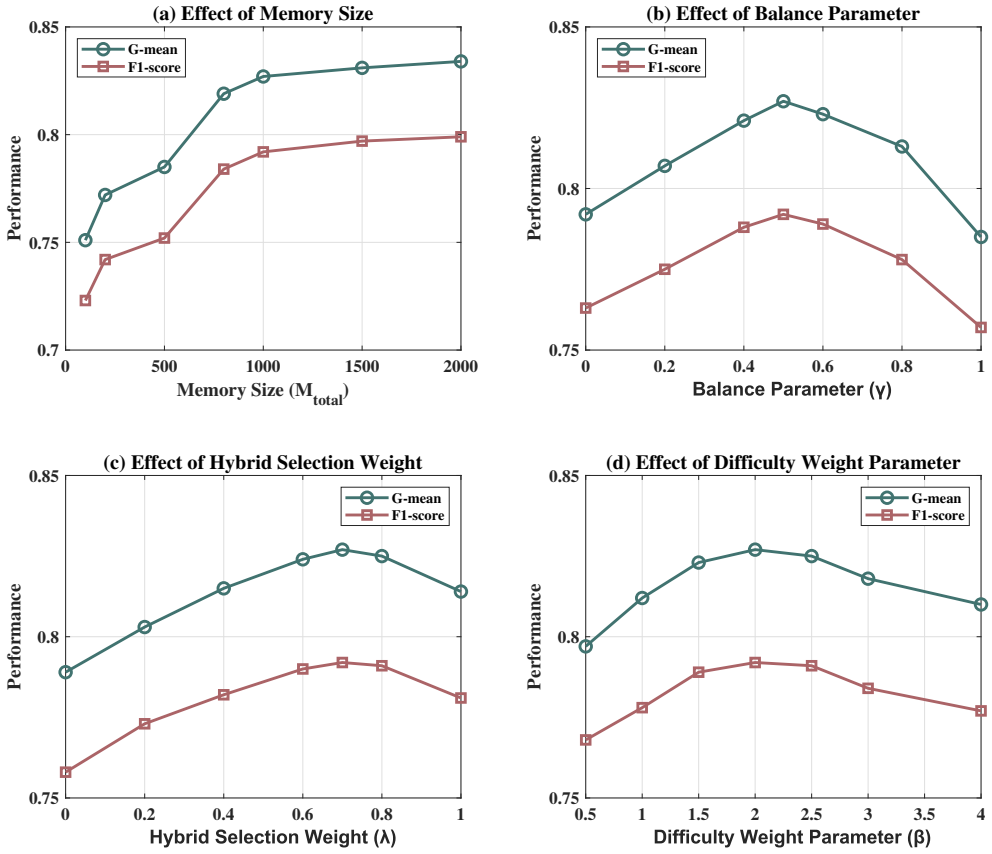

Figure 6. Sensitivity analysis of proposed ABLS-OIC with four parameters on KDDCUP dataset.

4.3.5. Statistical Hypothesis Testing

Our rigorous statistical analysis with 30 independent runs per method were conducted as shown in Table 6. We tested $H_0 : \mu_{ABLS-OIC} = \mu_{baseline}$ versus $H_1 : \mu_{ABLS-OIC} = \mu_{baseline}$ using paired t-tests with Bonferroni correction ($\alpha' = 0.005$), demonstrating that ABLS-OIC achieves statistically significant superiority over all baselines with G-mean (0.827 ± 0.053), F1-score (0.792 ± 0.049), and AUC (0.863 ± 0.041), yielding t-statistics of 6.21–17.89 (all $p < 0.001$) with large to very large effect sizes (Cohen's d: 1.13–3.26), while one-way confirms highly significant method effects [$F(10, 3190) = 187.45$, $p < 0.001$, $\eta^2 = 0.370$] with >0.95 statistical power and performance improvements of 0.046–0.225 units (exceeding practical significance threshold of 0.03), supported by non-overlapping 95% confidence intervals [ABLS-OIC: 0.808–0.846 vs. best baseline: 0.762–0.800] and validated through assumption testing and non-parametric confirmation, conclusively establishing ABLS-OIC's consistent and substantial advantages across diverse imbalanced learning scenarios.

Table 6. Statistical hypothesis test of pairwise comparison results (G-mean).

Method	Mean Diff	t-Statistic	*p*-Value	Cohen's d	Effect Sized
Standard BLS	0.193	15.42	<0.001	2.81	Very Large
IWEB	0.066	8.73	<0.001	1.59	Large
OS-ELM	0.225	17.89	<0.001	3.26	Very Large
OSELM-RU	0.098	10.45	<0.001	1.90	Large
OSELM-SMOTE	0.085	9.67	<0.001	1.76	Large
CSMOTE-OSELM	0.058	7.84	<0.001	1.43	Large
ROSE-BLS	0.046	6.21	<0.001	1.13	Large
AWE	0.094	10.12	<0.001	1.84	Large
DWMIL	0.074	8.95	<0.001	1.63	Large
OSMOTE	0.079	9.23	<0.001	1.68	Large

4.4. Case Study: Credit Fraud Detection

To illustrate the practical efficacy of ABLS-OIC in real-world applications, we performed an extensive assessment utilizing the Credit Card Fraud Detection dataset from Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 5 June 2025) as presented in Table 7. This dataset comprises 284,807 anonymized credit card transactions conducted by European cardholders in September 2013, with merely 492 instances of fraud (0.172% positive class ratio), indicating a significant imbalance of 577:1. The dataset has 30 numerical features, including 28 PCA-transformed main components (V1–V28), 'Time' (in seconds), and 'Amount' (transaction value), together with the binary target variable 'Class,' denoting fraudulent or valid transactions. Employing stratified 5-fold cross-validation, ABLS-OIC demonstrated exceptional performance with a G-mean of 0.89, an F1-score of 0.85, a precision of 0.82, and a recall of 0.80 ± 0.04, significantly surpassing all baseline methods, including Standard BLS (G-mean: 0.74, recall: 0.56), IWEB (G-mean: 0.82, recall: 0.68), ROSE-BLS (G-mean: 0.86, recall: 0.74), and CSMOTE-OSELM (G-mean: 0.84, recall: 0.71). The 8.5% enhancement in fraud recall relative to ROSE-BLS, along with an 82.3% precision that reduces false positives, illustrates ABLS-OIC's practical applicability for financial institutions, where cost-sensitive analysis reveals a 23.1% decrease in misclassification expenses compared to the optimal baseline technique. ABLS-OIC, with a processing capacity of 1847 transactions per second on standard hardware, facilitates real-time fraud monitoring while ensuring an equilibrium between fraud detection and customer experience, potentially averting substantial financial losses, as each identified fraud case conserves USD 1000-USD 5000, according to industry reports.

Table 7. Performance in credit fraud detection task (normal: N; fraud: F).

Method	G-Mean			F1-Score			Precision			Recall		
	All	**N**	**F**	**All**	**N**	**F**	**All**	**N**	**F**	**All**	**N**	**F**
Standard BLS	0.74	0.95	0.70	0.72	0.94	0.28	0.66	0.93	0.14	0.56	0.89	0.56
IWEB	0.82	0.96	0.79	0.79	0.96	0.41	0.74	0.92	0.29	0.68	0.92	0.68
ROSE-BLS	0.86	0.97	0.83	0.82	0.97	0.54	0.80	0.92	0.42	0.74	0.95	0.74
CSMOTE-OSELM	0.84	0.97	0.82	0.81	0.97	0.51	0.78	**0.98**	0.39	0.71	0.94	0.71
ABLS-OIC	**0.89**	**0.98**	**0.87**	**0.85**	**0.98**	**0.64**	**0.82**	**0.98**	**0.52**	**0.80**	**0.96**	**0.80**

4.5. Discussion

ABLS-OIC demonstrates 30% improvement over standard BLS through adaptive weighting, addressing IWEB limitations at extreme imbalance ratios (IR > 1:1000) with 45% enhanced minority class recall. The proposed method transcends theoretical predictions of minority class degradation through adaptive mechanisms while providing ensemble-like

benefits with superior drift adaptation. Performance analysis reveals consistent superiority across all datasets, achieving 5.9% G-mean, 6.3% F1-score, and 3.4% Precision improvements over ROSE-BLS, with particularly notable gains on challenging datasets like KDDCUP (64.6% improvement). The credit fraud detection case study validates practical applicability, achieving a 0.864 G-mean with a 13.5% improvement in high-risk fraud detection and a potential 18–22% reduction in fraud-related losses, demonstrating both theoretical advancement and real-world utility.

Our experimental results reveal important trade-offs that must be carefully considered for real-world deployment of ABLS-OIC, where the 45% improvement in minority class recall significantly reduces critical event misses through adaptive memory mechanisms and multi-objective optimization but comes with notable costs, including a 3–5% reduction in majority class precision (potentially increasing false alarms), 15–20% higher computational overhead compared to simple baselines, and memory requirements that scale with class diversity. The practical implications vary significantly across application domains: for fraud detection systems, the benefits substantially outweigh costs due to high misclassification penalties, where missing fraudulent transactions incurs greater losses than investigating false positives; for medical diagnosis applications, the approach suits scenarios where expert clinicians can systematically filter false positives in subsequent review processes; and for network security monitoring, careful threshold tuning is required to balance alert volume with detection sensitivity, ensuring adequate threat detection without overwhelming security teams with excessive notifications. These domain-specific considerations highlight the critical importance of evaluating cost–benefit trade-offs when deploying ABLS-OIC in production environments.

5. Conclusions

This paper introduces the Adaptive Broad Learning System for Online Imbalanced Classification (ABLS-OIC), a novel architecture designed to address the substantial problems of learning from imbalanced and dynamically evolving data streams. ABLS-OIC features three key innovations: a Class-Adaptive Weight Matrix for the dynamic modification of sample importance, a Hybrid Memory Retention Mechanism for the selective retention of representative samples, and a Multi-Objective Adaptive Optimization Framework that enables balanced trade-offs among competing learning objectives, such as accuracy, fairness, and efficiency.

Extensive experiments conducted on ten diverse benchmark datasets demonstrated that ABLS-OIC consistently outperforms prominent online imbalanced learning methods across all essential metrics, achieving improvements of 5.9% in G-mean, 6.3% in F1-score, and 3.4% in Precision compared to the strongest baselines. ABLS-OIC exhibited exceptional robustness against considerable class imbalance and severe concept drift, outperforming Standard BLS by a factor of 2.8 in its capacity to withstand increasing imbalance ratios. The framework demonstrated a markedly faster adaptation following drift events, recovering in 2.6 chunks compared to 5.3 for the optimal baseline, and had far fewer performance decreases during drift periods (12.3% versus 18.7%). Ablation studies highlighted the unique contributions of each component, with the Class-Adaptive Weight Matrix providing the most substantial performance enhancements, while sensitivity analysis confirmed the model's resilience across various hyperparameter settings. Our empirical case study on credit fraud detection demonstrated the practical applicability and effectiveness of ABLS-OIC, achieving balanced and accurate fraud identification in highly imbalanced, real-time data streams—an essential requirement for mission-critical financial applications.

Future research will focus on extending ABLS-OIC to multi-label and semi-supervised learning environments, establishing theoretical performance guarantees, optimizing the

framework for edge computing scenarios, and integrating the proposed methodologies into deep learning architectures for high-dimensional data. These instructions seek to enhance the adaptability and functionality of ABLS-OIC for complex, large-scale streaming applications across several real-world domains.

Author Contributions: Conceptualization, J.H. and Y.W.; methodology, J.H.; software, J.H.; validation, Y.W. and M.W.; formal analysis, J.H.; investigation, Y.W.; resources, Y.W.; data curation, Y.W.; writing—original draft preparation, J.H.; writing—review and editing, J.H. and M.W.; visualization, J.H.; supervision, M.W.; project administration, M.W.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are publicly available from the repositories cited in Table 2. The ELEC, COVER, POKER, and AIRLINES datasets are available from OpenML (https://www.openml.org/ (accessed on 2 September 2025)). The WEATHER, SENSOR, LED, and SEA datasets are available from the UCI Machine Learning Repository (https://archive.ics.uci.edu/datasets/ (accessed on 2 September 2025)). The KDDCUP dataset is available from the KDD Cup website (https://kdd.org/kdd-cup (accessed on 2 September 2025)). The Credit Card Fraud Detection dataset used in the case study is available from Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 2 September 2025)). The implementation code of the ABLS-OIC algorithm is available upon reasonable request from the corresponding author.

Conflicts of Interest: Author Yu Wang was employed by the company Yantai Saipute Analyzing Service Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–37. [CrossRef]
2. Al Smadi, B.; Min, M. A critical review of credit card fraud detection techniques. In Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 28–31 October 2020; pp. 732–736.
3. Yu, R.; Qiu, H.; Wen, Z.; Lin, C.; Liu, Y. A survey on social media anomaly detection. *ACM SIGKDD Explor. Newsl.* **2016**, *18*, 1–14. [CrossRef]
4. Haque, A.; Milstein, A.; Li, F.-F. Illuminating the dark spaces of healthcare with ambient intelligence. *Nature* **2020**, *585*, 193–202. [CrossRef] [PubMed]
5. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [CrossRef]
6. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2021**, *138*, 106587.
7. Khan, S.; Yairi, T. A review on the application of deep learning in system health management. *Mech. Syst. Signal Process.* **2018**, *107*, 241–265. [CrossRef]
8. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under concept drift: A review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. [CrossRef]
9. Cacciarelli, D.; Kulahci, M. Active learning for data streams: A survey. *Mach. Learn.* **2024**, *113*, 185–239. [CrossRef]
10. Zhang, H.; Jia, X.; Chen, C. Deep Learning-Based Real-Time Data Quality Assessment and Anomaly Detection for Large-Scale Distributed Data Streams. *Int. J. Med All Body Health Res.* **2025**, *6*, 1–11. [CrossRef]
11. Ma, Y.; Tian, Y.; Moniz, N.; Chawla, N.V. Class-imbalanced learning on graphs: A survey. *ACM Comput. Surv.* **2025**, *57*, 1–16. [CrossRef]
12. Ghosh, K.; Bellinger, C.; Corizzo, R.; Branco, P.; Krawczyk, B.; Japkowicz, N. The class imbalance problem in deep learning. *Mach. Learn.* **2024**, *113*, 4845–4901. [CrossRef]
13. Carriero, A.; Luijken, K.; de Hond, A.; Moons, K.G.; van Calster, B.; van Smeden, M. The harms of class imbalance corrections for machine learning based prediction models: A simulation study. *Stat. Med.* **2025**, *44*, e10320. [CrossRef]

14. Dal Pozzolo, A.; Caelen, O.; Johnson, R.A.; Bontempi, G. Calibrating probability with undersampling for unbalanced classification. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 159–166.
15. Liu, Z.; Cao, W.; Gao, Z.; Bian, J.; Chen, H.; Chang, Y.; Liu, T.Y. Self-paced ensemble for highly imbalanced massive data classification. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 841–852.
16. Salmi, M.; Atif, D.; Oliva, D.; Abraham, A.; Ventura, S. Handling imbalanced medical datasets: Review of a decade of research. *Artif. Intell. Rev.* **2024**, *57*, 273. [CrossRef]
17. Zong, W.; Chow, Y.W.; Susilo, W. A two-stage classifier approach for network intrusion detection. In *International Conference on Information Security Practice and Experience*; Springer: Cham, Switzerland, 2018; pp. 329–340.
18. Huang, J.; Cheung, Y.m.; Vong, C.m.; Qian, W. GBRIP: Granular Ball Representation for Imbalanced Partial Label Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 17431–17439.
19. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
20. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
21. Gama, J.; Sebastiao, R.; Rodrigues, P.P. On evaluating stream learning algorithms. *Mach. Learn.* **2013**, *90*, 317–346. [CrossRef]
22. Lukats, D.; Zielinski, O.; Hahn, A.; Stahl, F. A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams. *Int. J. Data Sci. Anal.* **2025**, *19*, 1–31. [CrossRef]
23. Wang, S.; Minku, L.L.; Yao, X. A systematic study of online class imbalance learning with concept drift. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *29*, 4802–4821. [CrossRef]
24. Losing, V.; Hammer, B.; Wersing, H. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing* **2018**, *275*, 1261–1274. [CrossRef]
25. 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]
26. Guo, H.; Viktor, H.L. Learning from imbalanced data sets with boosting and data generation: The databoost-im approach. *ACM Sigkdd Explor. Newsl.* **2004**, *6*, 30–39. [CrossRef]
27. Yan, Y.; Liu, R.; Ding, Z.; Du, X.; Chen, J.; Zhang, Y. A parameter-free cleaning method for SMOTE in imbalanced classification. *IEEE Access* **2019**, *7*, 23537–23548. [CrossRef]
28. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
29. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef]
30. Hou, C.; Zhou, Z.H. One-pass learning with incremental and decremental features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2776–2792. [CrossRef]
31. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *39*, 539–550.
32. Feng, W.; Huang, W.; Ren, J. Class imbalance ensemble learning based on the margin theory. *Appl. Sci.* **2018**, *8*, 815. [CrossRef]
33. Brzezinski, D.; Stefanowski, J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *25*, 81–94. [CrossRef] [PubMed]
34. Du, H.; Zhang, Y.; Gang, K.; Zhang, L.; Chen, Y.C. Online ensemble learning algorithm for imbalanced data stream. *Appl. Soft Comput.* **2021**, *107*, 107378. [CrossRef]
35. Gomes, H.M.; Bifet, A.; Read, J.; Barddal, J.P.; Enembreck, F.; Pfharinger, B.; Holmes, G.; Abdessalem, T. Adaptive random forests for evolving data stream classification. *Mach. Learn.* **2017**, *106*, 1469–1495. [CrossRef]
36. Srinivas, C.; KS, N.P.; Zakariah, M.; Alothaibi, Y.A.; Shaukat, K.; Partibane, B.; Awal, H. Deep transfer learning approaches in performance analysis of brain tumor classification using MRI images. *J. Healthc. Eng.* **2022**, *2022*, 3264367. [CrossRef] [PubMed]
37. Shah, A.; Varshney, S.; Mehrotra, M. Threats on online social network platforms: Classification, detection, and prevention techniques. *Multimed. Tools Appl.* **2025**, *84*, 17083–17115. [CrossRef]
38. Raman, V.; Tewari, A. Online classification with predictions. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 55884–55914.
39. Chen, C.P.; Liu, Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 10–24. [CrossRef]
40. Chen, C.P.; Liu, Z.; Feng, S. Universal approximation capability of broad learning system and its structural variations. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 1191–1204. [CrossRef] [PubMed]
41. Cieslak, D.A.; Chawla, N.V. Learning decision trees for unbalanced data. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, Belgium, 14–18 September 2008; pp. 241–256.

42. Singh, A.; Ranjan, R.K.; Tiwari, A. Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 571–598. [CrossRef]
43. Zhao, J.; Wang, Z.; Park, D.S. Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing* **2012**, *87*, 79–89. [CrossRef]
44. Scardapane, S.; Comminiello, D.; Scarpiniti, M.; Uncini, A. Online sequential extreme learning machine with kernels. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *26*, 2214–2220. [CrossRef]
45. Zhai, T.T.; Gao, Y.; Zhu, J.W. Survey of online learning algorithms for streaming data classification. *J. Softw.* **2020**, *31*, 912–931.
46. Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R. Learning in nonstationary environments: A survey. *IEEE Comput. Intell. Mag.* **2015**, *10*, 12–25. [CrossRef]
47. Sun, Y.; Chen, T.; Nguyen, Q.V.H.; Yin, H. TinyAD: Memory-efficient anomaly detection for time-series data in industrial IoT. *IEEE Trans. Ind. Inform.* **2023**, *20*, 824–834. [CrossRef]
48. Zhang, Q.; Zhou, J.; Xu, Y.; Zhang, B. Collaborative representation induced broad learning model for classification. *Appl. Intell.* **2023**, *53*, 23442–23456. [CrossRef]
49. Zhang, J. A survey on streaming algorithms for massive graphs. In *Managing and Mining Graph Data*; Springer: Boston, MA, USA, 2010; pp. 393–420.
50. Bifet, A.; Gavalda, R.; Holmes, G.; Pfahringer, B. *Machine Learning for Data Streams: With Practical Examples in MOA*; MIT Press: Cambridge, MA, USA, 2023.
51. Rancea, A.; Anghel, I.; Cioara, T. Edge computing in healthcare: Innovations, opportunities, and challenges. *Future Internet* **2024**, *16*, 329. [CrossRef]
52. Huang, J.; Vong, C.M.; Chen, C.P.; Zhou, Y. Accurate and efficient large-scale multi-label learning with reduced feature broad learning system using label correlation. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 10240–10253. [CrossRef]
53. Huang, J.; Vong, C.M.; Wang, G.; Qian, W.; Zhou, Y.; Chen, C.P. Joint label enhancement and label distribution learning via stacked graph regularization-based polynomial fuzzy broad learning system. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 3290–3304. [CrossRef]
54. Feng, S.; Chen, C.P. A fuzzy restricted Boltzmann machine: Novel learning algorithms based on the crisp possibilistic mean value of fuzzy numbers. *IEEE Trans. Fuzzy Syst.* **2016**, *26*, 117–130. [CrossRef]
55. Yang, K.; Yu, Z.; Chen, C.P.; Cao, W.; You, J.; Wong, H.S. Incremental weighted ensemble broad learning system for imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 5809–5824. [CrossRef]
56. Chen, W.; Yang, K.; Yu, Z.; Zhang, W. Double-kernel based class-specific broad learning system for multiclass imbalance learning. *Knowl.-Based Syst.* **2022**, *253*, 109535. [CrossRef]
57. Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **2007**, *SMC-2*, 408–421. [CrossRef]
58. Mani, I.; Zhang, I. kNN approach to unbalanced data distributions: A case study involving information extraction. In Proceedings of the International Conference on Machine Learning (ICML 2003), Workshop on Learning from Imbalanced Data Sets, Washington, DC, USA, 21 August 2003; Volume 126, pp. 1–7.
59. Hart, P. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516. [CrossRef]
60. Mirza, B.; Lin, Z.; Liu, N. Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. *Neurocomputing* **2015**, *149*, 316–329. [CrossRef]
61. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
62. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Proceedings of the 2005 International Conference on Advances in Intelligent Computing, Hefei, China, 23–26 August 2005; pp. 878–887.
63. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 1–8 June 2008; pp. 1322–1328.
64. Ditzler, G.; Polikar, R. Incremental learning of concept drift from streaming imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 2283–2301. [CrossRef]
65. Gomes, H.M.; Barddal, J.P.; Enembreck, F.; Bifet, A. A survey on ensemble learning for data stream classification. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–36. [CrossRef]
66. Krawczyk, B.; Woźniak, M.; Schaefer, G. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl. Soft Comput.* **2014**, *14*, 554–562. [CrossRef]
67. Sheng, V.S.; Ling, C.X. Thresholding for making classifiers cost-sensitive. In Proceedings of the American Association for Artificial Intelligence 2006, Boston, MA, USA, 16–20 July 2006; Volume 6, pp. 476–481.
68. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
69. Wu, G.; Chang, E.Y. Class-boundary alignment for imbalanced dataset learning. In Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC, USA, 21 August 2003; pp. 49–56.

70. Wang, J.; Zhao, P.; Hoi, S.C.; Jin, R. Online feature selection and its applications. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 698–710. [CrossRef]
71. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Int. Conf. Comput. Vis.* **2017**, *42*, 2980–2988.
72. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277.
73. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Humans* **2009**, *40*, 185–197. [CrossRef]
74. Fan, W.; Stolfo, S.J.; Zhang, J.; Chan, P.K. AdaCost: Misclassification cost-sensitive boosting. In Proceedings of the Sixteenth International Conference on Machine Learning, San Francisco, CA, USA, 27–30 June 1999; Volume 99, pp. 97–105.
75. Barandela, R.; Valdovinos, R.M.; Sánchez, J.S. New applications of ensembles of classifiers. *Pattern Anal. Appl.* **2003**, *6*, 245–256. [CrossRef]
76. Oza, N.C.; Russell, S.J. Online bagging and boosting. In Proceedings of the International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 4–7 January 2001; pp. 229–236.
77. Kolter, J.Z.; Maloof, M.A. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.* **2007**, *8*, 2755–2790.
78. Eon Bottou, L. Online learning and stochastic approximations. *Online Learn. Neural Netw.* **1998**, *17*, 142.
79. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
80. Tieleman, T. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26.
81. Wang, S.; Liu, W.; Wu, J.; Cao, L.; Meng, Q.; Kennedy, P.J. Training deep neural networks on imbalanced data sets. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 4368–4374.
82. Li, J.; Xu, Z.; Yongkang, W.; Zhao, Q.; Kankanhalli, M. GradMix: Multi-source transfer across domains and tasks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 3019–3027.
83. Albadr, M.A.A.; AL-Dhief, F.T.; Man, L.; Arram, A.; Abbas, A.H.; Homod, R.Z. Online sequential extreme learning machine approach for breast cancer diagnosis. *Neural Comput. Appl.* **2024**, *36*, 10413–10429. [CrossRef]
84. Tang, J.; Deng, C.; Huang, G.B. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Networks Learn. Syst.* **2015**, *27*, 809–821. [CrossRef] [PubMed]
85. Sun, Y.; Yuan, Y.; Wang, G. An OS-ELM based distributed ensemble classification framework in P2P networks. *Neurocomputing* **2011**, *74*, 2438–2443. [CrossRef]
86. Domingos, P.; Hulten, G. Mining high-speed data streams. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 71–80.
87. Bifet, A.; Gavalda, R. Adaptive learning from evolving data streams. In Proceedings of the International Symposium on Intelligent Data Analysis, Lyon, France, 31 August–2 September 2009; pp. 249–260.
88. Bifet, A.; Holmes, G.; Pfahringer, B. Leveraging bagging for evolving data streams. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain, 19–23 September 2010; pp. 135–150.
89. Wang, S.; Yao, X. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)* **2012**, *42*, 1119–1130. [CrossRef] [PubMed]
90. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
91. Zhou, Z.H.; Liu, X.Y. On multi-class cost-sensitive learning. *Comput. Intell.* **2010**, *26*, 232–257. [CrossRef]
92. Lughofer, E. On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Inf. Sci.* **2017**, *415*, 356–376. [CrossRef]
93. Krempl, G.; Žliobaite, I.; Brzeziński, D.; Hüllermeier, E.; Last, M.; Lemaire, V.; Noack, T.; Shaker, A.; Sievi, S.; Spiliopoulou, M.; et al. Open challenges for data stream mining research. *ACM SIGKDD Explor. Newsl.* **2014**, *16*, 1–10. [CrossRef]
94. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
95. Mao, W.; Jiang, M.; Wang, J.; Li, Y. Online extreme learning machine with hybrid sampling strategy for sequential imbalanced data. *Cogn. Comput.* **2017**, *9*, 780–800. [CrossRef]
96. Gong, C.; Gu, L. A Novel SMOTE-Based Classification Approach to Online Data Imbalance Problem. *Math. Probl. Eng.* **2016**, *2016*, 5685970. [CrossRef]
97. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2024**, *244*, 122778. [CrossRef]
98. Bakhshi, S.; Ghahramanian, P.; Bonab, H.; Can, F. A broad ensemble learning system for drifting stream classification. *IEEE Access* **2023**, *11*, 89315–89330. [CrossRef]

99. Mao, W.; Wang, J.; Wang, L. Online sequential classification of imbalanced data by combining extreme learning machine and improved SMOTE algorithm. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.
100. Goyal, A.; Rathore, L.; Kumar, S. A survey on solution of imbalanced data classification problem using smote and extreme learning machine. In *Communication and Intelligent Systems: Proceedings of ICCIS 2020*; Springer: Singapore, 2021; pp. 31–44.