mathematics

MDPI

Article

Fractional Adaptive Resonance Theory (FRA-ART): An Extension for a Stream Clustering Method with Enhanced Data Representation

Yingwen Zhu [1,*], Ping Li [2], Qian Zhang [1], Yi Zhu [1] and Jun Yang [3]

[1] School of Information Technology, Jiangsu Open University, Nanjing 210036, China; usstzhangqian@163.com (Q.Z.); zhuyi@jsou.edu.cn (Y.Z.)

[2] College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; ping.li@njupt.edu.cn

[3] College of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China; yangjun_nju@163.com

* Correspondence: zhu_ying_wen@163.com; Tel.: +86-139-138-13193

Abstract: Clustering data streams has become a hot topic and has been extensively applied to many real-world applications. Compared with traditional clustering, data stream clustering is more challenging. Adaptive Resonance Theory (ART) is a powerful (online) clustering method, it can automatically adjust to learn both abstract and concrete information, and can respond to arbitrarily large non-stationary databases while having fewer parameters, low computational complexity, and less sensitivity to noise, but its limited feature representation hinders its application to complex data streams. In this paper, considering its advantages and disadvantages, we present its flexible extension for stream clustering, called fractional adaptive resonance theory (FRA-ART). FRA-ART enhances data representation by fractionally exponentiating input features using self-interactive basis functions (SIBFs) and incorporating feature interaction through cross-interactive basis functions (CIBFs) at the cost only of introducing an additionally adjustable fractional order. Both SIBFs and CIBFs can be precomputed using existing algorithms, making FRA-ART easily adaptable to any ART variant. Finally, comparative experiments on five data stream datasets, including artificial and real-world datasets, demonstrate FRA-ART's superior robustness and comparable or improved performance in terms of accuracy, normalized mutual information, rand index, and cluster stability compared to ART and the state-of-the-art G-Stream algorithm.

Keywords: data stream clustering; fractional order; adaptive resonance theory; interactive basis functions; self-interactive; cross-interactive

MSC: 68W27

check for updates

Citation: Zhu, Y.; Li, P.; Zhang, Q.; Zhu, Y.; Yang, J. Fractional Adaptive Resonance Theory (FRA-ART): An Extension for a Stream Clustering Method with Enhanced Data Representation. *Mathematics* **2024**, *12*, 2049. https://doi.org/10.3390/math12132049

Academic Editor: Ioannis G. Tsoulos

Received: 5 June 2024
Revised: 24 June 2024
Accepted: 27 June 2024
Published: 30 June 2024

1. Introduction

The number of interconnected devices, including sensors, is steadily increasing and these devices continuously produce massive amounts of data streams at high speed [1,2]. A data stream is an ordered, unbounded sequence of data. Throughout its lifetime, it frequently undergoes rapid changes, necessitating fast and time-aware tools for analysis. Mining data streams [3–14] has become a hot issue in data analysis, including data stream clustering, data stream classification, etc. Among these, clustering data streams has been extensively applied to many applications, including network intrusion detection, healthcare monitoring, stock market analysis, Internet of Things device tracking, fraud detection, customer segmentation, and environmental surveillance. Traditional clustering methods are performed on a static dataset, whereas data stream clustering is subject to certain constraints, such as single-pass, real-time response, bounded memory, and the ability to detect concept drift, due to the dynamic nature of data streams. To tackle these challenges, the first approach that comes to mind is to adapt traditional clustering methods to handle data streams. As a result, numerous *data stream clustering* methods have been developed including partitioning methods (STREAM [15], streamKM++ [16], streamingKMeans [17], Adaptive Streaming k-Means [17], FEAC-Stream [18]), hierarchical methods (E-Stream [19], SWClustering [20], ClusTree [21]), density-based methods (incPreDecon [22], OPTICS-Stream [23], DenStream [24], CEDAS [25], DBStream [26], Improved Data Stream Clustering Algorithm [27], ACSC [28]), grid-based methods (MR-Stream [29], D-Stream [30], MuDi-Stream [31]), and model-based methods (GCPSOM [32], G-Stream [33], SWEM [34], RPGStream [35]). Table 1 provides a summary of the discussed stream clustering methods based on the following criteria: (1) the category to which it belongs, (2) the computational method used (two-phase learning or online learning), (3) the adaptivity of the number of clusters k, (4) the recognizability of the topological structure, (5) the detectability of concept drift, and (6) the adaptability to high-dimensional data.

Table 1. Comparison of different data stream clustering algorithms.

Algorithms	Category	Online	Adaptivity	Topology	Detectability	High-Dimensional
STREAM	partitioning	✓	×	×	×	×
streamKM++	partitioning	×	×	×	×	×
streamingKMeans	partitioning	✓	✓	×	✓	✓
Adaptive Streaming k-Means	partitioning	✓	✓	×	✓	✓
FEAC-Stream	partitioning	✓	✓	×	✓	✓
E-Stream	hierarchical	×	✓	×	✓	×
SWClustering	hierarchical	×	✓	×	✓	×
ClusTree	hierarchical	×	✓	×	✓	×
OPTICS-Stream	density	×	✓	×	✓	×
DenStream	density	×	✓	×	✓	×
incPreDecon	density	✓	✓	×	×	✓
CEDAS	density	✓	✓	×	✓	✓
DBStream	density	×	✓	×	✓	✓
IDS	density	×	✓	×	✓	✓
ACSC	density	✓	✓	×	✓	×
MR-Stream	grid	×	✓	×	✓	×
D-Stream	grid	×	✓	×	✓	×
MuDi-Stream	grid	×	✓	×	✓	×
G-Stream	model	✓	✓	✓	✓	×
GCPSOM	model	✓	✓	✓	✓	×
SWEM	model	✓	✓	×	✓	×
RPGStream	model	✓	✓	✓	✓	✓

From Table 1, it is evident that, despite the numerous clustering methods proposed, model-based algorithms are often the most promising and practical choice for handling constraints on streaming data, such as bounded memory, real-time response, and single-pass in general. GCPSOM [32], G-Stream [33], SWEM [34], and RPGStream [35] are all model-based algorithms. They are all capable of handling concept drift. Although these algorithms perform a single pass operation, they do not require the number of clusters to be specified in advance, and can identify clusters of arbitrary shapes, as well as representing the data structure through a graph. Unfortunately, they have many parameters to be tuned. In order to cope with this challenge, our aim is to develop a stream clustering algorithm that is user-friendly and straightforward to implement, while having fewer parameters.

Adaptive Resonance Theory (ART) [36] is considered to be the most advanced cognitive and neural theory in an ever-changing world. It is even capable of learning a data stream in one learning trial. The concept of vigilance control has been proposed as a means to adjust the generality of categories in ART. The advantages of ART are its fast speed to deal with data and the fewer iterations that are required to converge, thus naturally explaining the fact that ART is particularly suitable for handling large streaming datasets that cannot be stored in memory as a whole. Consequently, ART is particularly well suited for incremental on-line learning, and has been widely utilized in a variety of clustering approaches found in the literature. For example, ART1 [37] published in 1987 is a pioneering work, while other well-known extensions including ART2 [38], ART2-A [39], ART3 [40], Fuzzy ART [41], Gaussian ART [42], Hypersphere ART [43], TopoART [44], DDVFA [45], and DVFA [46] have been proposed. Among them, Fuzzy ART is demonstrably the most commonly used to date. It extends the capability of ART1 by incorporating computations from fuzzy set theory. Typically, it uses complement coding to pre-process samples. The authors of [47] demonstrate that, with complement coding, the vigilance parameter of a cluster in Fuzzy ART forms a hyperoctagon region within the high-dimensional feature space, known as a vigilance region (VR) and the clustering mechanism of Fuzzy ART is significantly changed. SA-ART [48] extends Fuzzy ART and provides a new strategy for modeling significant clustering features. Although ART has been greatly improved, it is still not flexible enough, and can not yield more useful features for clustering. To overcome these shortcomings, firstly, we find that using fractional order on a single variable can make its value larger and still fall between 0 and 1. So a fractionally self-interactive approch can be combined with Fuzzy ART to make it more flexible. Secondly, since feature interaction information can help enhance data representation, it is natural for us to extend the single variable to bivariate or multivariate [49]. Thus, fractionally cross-interactive base functions are introduced. Considering the above motivation, while keeping ART's advantages and adapting it to a data stream mining scenario, the aim of this study is to present a novel data stream clustering method based on ART that leverages fractional order information and interactive basis functions to enhance flexibility, data representation, and clustering performance. To summarize, the contributions of this paper are three-fold:

(1) Presenting a novel ART-based data stream clustering method in which we introduce the fractional order information into ART to make it more flexible and further promote the ability of data representation.
(2) Extending a Fuzzy ART utilizing novel method for creating flexible decision boundaries through the use of fractionally self-interactive or cross-interactive basis functions (SIBFs or CIBFs). Note our work is the first attempt to employ the interactive basis function for the data stream clustering problem. Furthermore, non-linear partitions can also be achieved based on the selected basis functions.
(3) Proposed SIBFs and CIBFs can be precomputed once independently and then used as an input to any variants of the ART with additional low complexity. As only the inputs to the algorithm vary, this suggests that basis functions can be utilized in numerous existing implementations of ART.

The remaining sections of this paper are organized as follows: Section 2 reviews related work, Section 3 introduces the FRA-ART model (the main contribution of this work), Section 4 presents the experimental results, and Section 6 concludes the paper.

2. Related Works

2.1. Elementary Frameworks of ART Models

The elementary ART models serve as a robust foundation for constructing intricate ART-based systems, enabling them to execute all three types of machine learning models, including unsupervised learning, supervised learning, and reinforcement learning [50]. The fundamental structure of the elementary ART model (Figure 1), comprises an input field F_1 for receiving input patterns and a cluster field F_2 for organizing these patterns. The generic network is described as follows.

- Input filed F_1: this is the input layer. The output $\mathbf{y}^{(F_1)}$ of this layer propagates the input samples $x \in R^d$ to the F_2 layer through the bottom-up long-term memory units (LTMs) θ^{bu}. The comparison layer F_1 compares the input sample x with the F_2's expectation and then sends the outcome $\mathbf{y}^{(F_1)}$ to the orienting subsystem.

- Category field F_2: this is the competitive layer. The network output $\mathbf{y}^{(F_2)}$ (short-term memory units) is yielded in this layer. The LTM related to category j is $\theta_j = \{\theta_j^{bu}, \theta_j^{td}\}$, $j = 1, \ldots, N$, where θ indicates the LTM of a given category.
- Orienting subsystem: this subsystem controls the learning and search mechanisms by inhibiting and permitting categories to resonate.

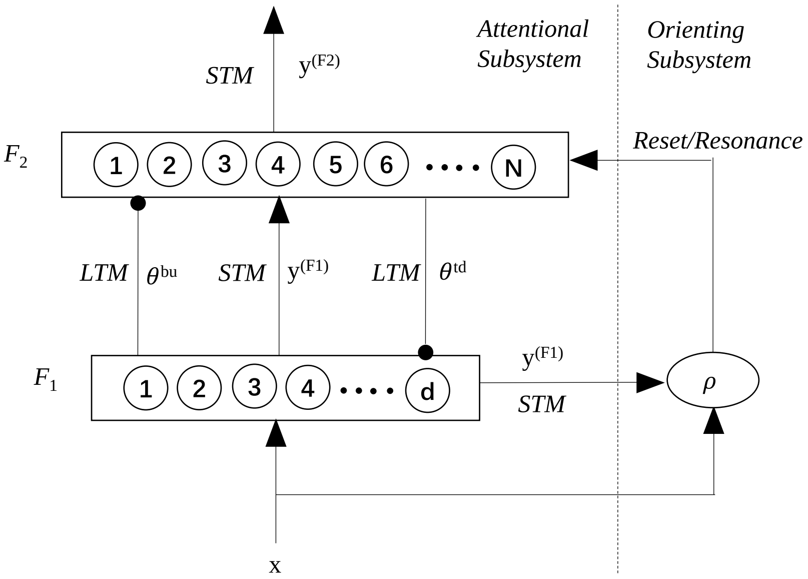


Figure 1. The fundamental structure of the ART model.

ART models are dynamic, self-organizing, modular and competitive networks. In general, each category represents a hypothesis. Since a new sample x is coming, the neuron J that maximizes the model's *activation function T* for this sample is selected

$$J = arg \max_j (T_j). \tag{1}$$

And a vigilance test is next required to evaluate the sufficiency of the chosen category, that is, the winner category must fulfill *a match function*. If the value of this test is greater than the vigilance parameter ρ, a resonance occurs, and learning is then permitted. Or else, category J is inhibited, the next winner is chosen among the rest of the categories in the category field, and the search continues. Finally, if no winner meets the necessary resonance condition (s), then a new category is generated. Note that the parameter ρ controls the network granularity.

A vigilance region (VR) of a given network category j can be denoted as:

$$VR_j = \{x : M_j(x) \; satisfies \; the \; resonance \; criteria\}, \tag{2}$$

where M_j is the *match function*. It is the region encompassing all points where the resonance criteria are fulfilled. Thus, it can be modeled for the sample which satisfies (or not) the vigilance test using

$$\mathbf{1}_{VR_j}(x) = \begin{cases} 1 & if \; x \in VR_j, \\ 0 & otherwise, \end{cases} \tag{3}$$

where $\mathbf{1}_{(\cdot)}(\cdot)$ is the indicator function.

2.2. Fuzzy ART

Fuzzy ART, as the foundational model studied in this paper, is demonstrably the most commonly used ART model so far. Let x denote the input sample in the input field F_1. Specifically, let $x = (x_1, \ldots, x_d)$, where $x_i \in [0, 1]$ for each index $i = 1, \ldots, d$.

With complement coding, the original input dimension is doubled while maintaining a constant norm by applying the transformation ($x \leftarrow [x, 1-x]$):

$$\|x\|_1 = \sum_{i=1}^{2d} x_i = \sum_{i=1}^{d}(1-x_i) + \sum_{i=1}^{d} x_i = d. \tag{4}$$

LTM. The LTM unit in each category is represented by a weight vector $\theta = w$. By applying complement coding, the weight vector can be further decomposed as $w = [u, v^c]$, where u represents the lower left corner of the feature range and v^c represents the upper right corner.

Activation. The activity (or choice) function for the jth cluster is defined as:

$$T_j = \frac{\|x \wedge w_j\|_1}{\alpha + \|w_j\|_1}, \tag{5}$$

where the component-wise fuzzy AND operation $\wedge$ is defined by $(p \wedge q)_i \equiv min(p_i, q_i)$, and $\alpha > 0$ is the choice parameter which can be considered as a regularization parameter and is related to the system's complexity. The activity function evaluates the fuzzy subsethood degree of w_j in x and leans towards smaller categories. When the winner node J is selected via the WTA (winner-takes-all) competition, the F_2 activity becomes

$$y_j^{(F_2)} = \begin{cases} 1, & if\ j = J, \\ 0, & otherwise. \end{cases} \tag{6}$$

Moreover, the F_1 activity follows:

$$y^{(F_1)} = \begin{cases} x, & if\ F_2\ is\ inactive, \\ x \wedge w_j, & otherwise. \end{cases} \tag{7}$$

Match and resonance. The similarity between the winner category J and the input sample x is assessed through a match function, which is formulated as follows:

$$M_J = \frac{\|y^{(F_1)}\|_1}{\|x\|_1} = \frac{\|x \wedge w_J\|_1}{\|x\|_1}, \tag{8}$$

where ρ is the vigilance parameter, and $\rho \in [0,1]$, $VR_J = \{x : M_J(x) \geq \rho\}$. Like the activity function (Equation (5)), the match function also evaluates the fuzzy subsethood degree, particularly of x in w_J. If learning takes place, the updated category will not exceed the maximum allowed size. Specifically, the category j's size is defined to be

$$R_j = \sum_{i=1}^{d}[(1 - w_{j,d+i}) - w_{j,i}] = d - \|w_j\|_1, \tag{9}$$

which is equal to the height plus the width of R_j. Learning increases the size of each R_j. The vigilance criterion specifies the upper bound of the size of the category represented by the vigilance parameter ρ

$$R_J \oplus x \leq d(1-\rho), \tag{10}$$

where $R_J \oplus x$ stands for the smallest hyperrectangle that can contain the input sample x and R_J. Thus, high vigilance ($\rho \simeq 1$) again leads to small R_J while low vigilance ($\rho \simeq 0$) permits large R_J. In a fast-learn Fuzzy ART, if j is an uncommitted category, $w_j = x = (a, a^c)$, and $\|w_j\| = 2d$, the corners of R_j are then given by a and $(a^c)^c = a$. Hence, R_j is just the point a, and so, $R_j = -d$.

Learning. If the vigilance test of the current winning category does not pass, it will be inhibited, and we need to select a new winner from the rest of categories. If no winner satisfies the vigilance criteria, a new category will be generated to encode the presented input sample. When category J satisfies the vigilance criterion, its corresponding weight vector will be updated through a learning function, which is specified as follows:

$$w_J^{(new)} = (1-\beta)w_J^{(old)} + \beta(x \wedge w_J^{(old)}), \tag{11}$$

where $\beta \in (0,1]$ is the learning parameter, and the *fast learning mode* requires $\beta = 1$.

3. The Fractional ART Model for Data Stream Clustering

3.1. Self-Interactive and Cross-Interactive Basis Functions (SIBFs and CIBFs)

3.1.1. Self-Interactive Basis Functions (SIBFs)

In this section, we detail self-interactive basis functions [49]. The basic vectors are constituted by the features used for training. Two features form a $2D$ basis, and so on. Then, a self-interactive basis function is a transformation, which can be denoted as:

$$f(X) = X \tag{12}$$

which is a specific instance of a polynomial function, with $a = 1$:

$$f(X) = X^a \tag{13}$$

$$f(X) = (1-X)^a \tag{14}$$

Furthermore, we can define other basis functions, such as the exponential function:

$$f(X) = e^X \tag{15}$$

Let us consider the general scenario where K real-valued functions f_i with $i = 1, \ldots, K$, each defined on $\mathbb{R} \to \mathbb{R}$, are available as candidate basis functions. And note that $\{f_1, f_2, \ldots, f_K\}$ are a set of basis functions. Subsequently, we expand the original set of d features by incorporating T new features, which are generated through the application of the candidate basis functions:

$$X^* = (X_1, \ldots, X_d, X_{d+1}, \ldots, X_{d+T}) \tag{16}$$

where $X^* \in \mathbb{R}^{d+T}$, and $X_{d+i} = f_{s_i}(X_{j_i})$, for $i = 1, \ldots, T$, $j_i \in \{1, \ldots, d\}$ and $s_i \in \{1, \ldots, K\}$. As an example, when the number of dimensions is $d = 2$, that is, the feature space consists of two dimensions, represented by $X = \{X_1, X_2\}$. Moreover, $K = 1$ and $T = 1$ with $f_1(x) = x^2$, thus $X^* = \{X_1, X_2, X_3 = X_1^2\}$. Whenever a split s in X_3 is selected by the partitioning mechanism, it is projected in X as $X_1 = \sqrt{X_3}$, which is a constant. Therefore, any partitioning of the basis function dimensionality is equivalent to discovering an orthogonal decision boundary in the original basis.

3.1.2. Cross-Interactive Basis Functions (CIBFs)

Our proposed approach involves utilizing the inter-feature interactions between two or more features to construct X^*. These interactions are different from self-interactive, and can be identified by a set of M functions. These functions reproduce functional interactions among the transformations of the features by the basis functions. Hence, cross-interactive basis functions are defined as:

$$g_i : \mathbb{R}^{dK} \to \mathbb{R} \quad (f_1(X_1), f_2(X_2) \ldots, f_K(X_d)) \to b \tag{17}$$

We define $X^* = (X_1, \ldots, X_d, X_{d+1}, \ldots, X_{d+M})$, with $X_{d+i} = g_i(f_1(X_1), f_2(X_2) \ldots, f_K(X_d))$, for $i = 1, \ldots, M$. Thus, by taking the interactions among the features into account, an oblique partition will be provided, which may also eventually become non-linear.

For example, when $d = 2$, that is $X = \{X_1, X_2\}$, $K = 1$ with $f_1(x) = x$, $M = 1$ with $g_1(f_1(X_1), f_1(X_2)) = f_1(X_1) + f_1(X_2) = X_1 + X_2$ and $X^* = (X_1, X_2, X_3)$, we achieve that the projection of $X_3 = s$ on the original basis plane as $X_2 = s - X_1$, thereby providing an oblique partition.

Note that the framework offered by CIBFs not only enables us to acquire oblique partitions, but also permits us to achieve non-linear decision boundaries. This is accomplished by the projection of the equation $g_i(f_1(X_1), f_1(X_2) \ldots, f_K(X_d)) = b$ in the subspace $X = \{X_1, X_2, \ldots, X_d\}$ spanned by the features in the dataset. For instance, the CIBF $g_1(f_1(X_1), f_1(X_2)) = f_1(X_1)f_1(X_2)$ with $f_1(x) = x$ leads to X_1X_2. When $X_1X_2 = s$ is fixed by a split s and, therefore, $X_2 = s/X_1$, thus a hyperbolic partition is generated. As one other example, the CIBF $g_1(f_1(X_1), f_1(X_2)) = f_1(X_1) + f_1(X_2)$ with $f_1(x) = x^2$ leads to $X_1^2 + X_2^2$. When $X_1^2 + X_2^2 = s$ is fixed by a split s in the dimension of CIBF, and, therefore, $X_2 = \sqrt{s - X_1^2}$, thus a radial partition is created.

3.2. The FRA-ART Algorithm

In this section, we present a novel unsupervised self-organizing incremental neural network, called fractional adaptive resonance theory (FRA-ART), to cluster data streams. The intuition is that by a fractional order conversion of the raw input features, we induce a single hyperparameter but make the FRA-ART more flexible and can further promote the ability of data representation.

We assume that we have a data stream consisting of a sequence $DS = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$ of n (potentially infinite) data arriving in times $t_1, t_2, \ldots$, where $\boldsymbol{x}_i = (x_i^1, x_i^2, \ldots, x_i^d)$ is an original vector in the $\mathbb{R}^d$ space. The FRA-ART algorithm extends Fuzzy ART with a new strategy through the use of fractionally self-interactive or cross-interactive basis functions (SIBFs or CIBFs). Our objective is to assess how these SIBFs or CIBFs perform in FRA-ART. When a sample $\boldsymbol{x} = (x^1, x^2, \ldots, x^d)$ is coming, note that each feature is normalized in $[0, 1]$. In terms of the SIBFs and CIBFs, We expand the set of d features with d new features:

$$\boldsymbol{x}^* = (x^1, x^2, \ldots, x^d, x^{d+1}, x^{d+2}, \ldots, x^{2d}) \tag{18}$$

where $\boldsymbol{x}^* \in \mathbb{R}^{2d}$, $x^{d+j} = f_p(x^j)$, $p \in \{1, \ldots, K\}$ when used SIBFs and $x^{d+j} = g_1(f_1(x^1),$ $f_2(x^2), \ldots, f_K(x^d))$ when used CIBFs. We consider the following functions:

$$f_1(x^j) = (x^j)^a \tag{19}$$

$$f_2(x^j) = (1 - x^j)^a \tag{20}$$

$$f_3(x^j) = (e^{x^j})^a \tag{21}$$

$$x^{d+j} = g_1(f_1(x^1), f_2(x^2), \ldots, f_K(x^d)) = \max \min_{m \in [1,d]\ m \neq j} (f_1(x^j), f_1(x^m)) \tag{22}$$

$$x^{d+j} = g_2(f_1(x^1), f_2(x^2), \ldots, f_K(x^d)) = \sum_{m=1, m \neq j}^{d} (f_1(x^j) f_1(x^m))/(d-1) \tag{23}$$

where a is a fractional order.

As a new data point x_i arrives, the proposed FRA-ART firstly uses IBFs (SIBFs or CIBFs) to expand the set of d features with d new features. Therefore, the data stream $\mathcal{DS} = \{x_1, x_2, \ldots\}$ is transformed to $\underline{X^*} = \{x_1^*, \underline{x_2^*}, \ldots\}$ one-by-one, $x_i^* \in [0, 1] (i = 1, 2, \underline{\ldots})$. By using complement coding, x_i^* and $\overline{x_i^*}$, where $\overline{x_i^*} = 1 - x_i^*$, results in the vector $I = (x_i^*, \overline{x_i^*})$. Then, we use Fuzzy ART to cluster this transformed data stream. The specific procedure is shown in Algorithm 1 and Figure 2.

Algorithm 1 FRA-ART

```
Require: $\mathcal{DS} = \{x_1, x_2, x_3, \ldots\}$.
Ensure: set of nodes $\mathcal{C} = \{c_1, c_2, c_3, \ldots\}$ and their prototypes $\boldsymbol{W} = \{w_{c_1}, w_{c_2}, w_{c_3}, \ldots\}$.
 1: for each input vector $x_i$ do
 2:     Compute $x_i^*$ using Equations (18)–(23);
 3:     Complement coding $x_i^*$ to $I = (x_i^*, \overline{x_i^*})$ ;
 4:     Compute activation function using Equation (5) to get the active nodes $\Omega$ ($\Omega \subseteq C$) ;
 5:     Select winner node $J$:  $J = arg_{j \in \Omega} max(T_j)$;
 6:     Compute match function using Equation (8);
 7:     if $M_J \geq \rho$ then
 8:         Update category $J$ using Equation (11);
 9:     else
10:         Deactivate category $J$: $\Omega \leftarrow \Omega - J$;
11:         if $\Omega \neq \emptyset$ then
12:             Go to step 5;
13:         else
14:             $J = |C| + 1$;
15:             Create new category: $C \leftarrow C \cup J$.
16:             Initialize new category: $w_J = I$.
17:         end if
18:     end if
19: end for
```

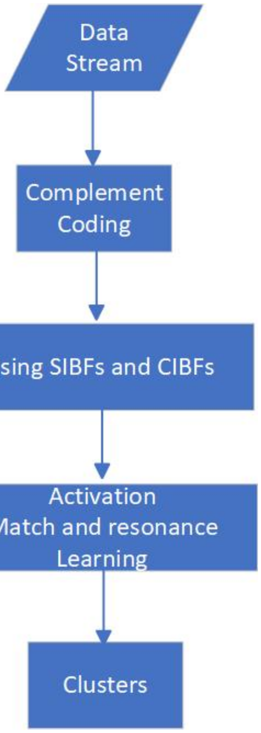


Figure 2. Flowchart of the FRA-ART algorithm.

3.3. Complexity Analysis

In this section, we discuss the time complexity of FRA-ART from two aspects: using IBFs to expand the set of d features and extending the Fuzzy ART algorithm to cluster. For the first case, the addition of SIBFs increases the computational complexity by $O(nd)$, while CIBFs further increase the complexity by $O(nd^2)$, where n is the number of samples, and d is the number of features. For the second case, Fuzzy ART undergoes three procedures including complement coding, activity function-based matching cluster identification, and match function-based template matching, as well as prototype learning or the creation of new clusters. The corresponding time complexities are $O(d)$, $O(kd)$, and $O(d)$, where k denotes the number of clusters and d denotes the number of features, respectively. Consequently, the time complexity of Fuzzy ART is linear. Specifically, it can be expressed as

$O(ndk)$ [51]. Therefore, the overall complexity of FRA-ART is less than $O(nd^2) + O(ndk) = O(nd(d+k))$.

4. Experiments

In this section, we present the experimental evaluation of the FRA-ART algorithm. We compare our algorithm with the stream clustering algorithm G-Stream and the Fuzzy ART algorithm, which are all model-based stream clustering algorithms. The performance of G-Stream has been shown to be better than that of many well-known data stream clustering algorithms. As explained in Section 2, Fuzzy ART is demonstrably the most commonly used ART model, and we use an online version of Fuzzy ART for data stream clustering. The experiments were conducted on a PC with Core(TM) i5-3470 running at 3.20 GHz processors, and 16 GB of RAM, which runs the Windows 7 Professional operating system, using the MATLAB platform.

4.1. Datasets and Parameters Setting

We evaluate the clustering quality and clustering scalability of the FRA algorithm using not only artificial datasets but also real datasets, the details of which are given in Tables 2 and 3 below.

Table 2. The details of five datasets.

Dataset	Letter4	Kddcup99	CoverType	Powersupply	Sensor
#samples	9344	494,021	581,012	29,928	2,219,803
#features	2	41	54	2	5
#classes	7	23	7	24	54

- letter4: letter4 is an artificial dataset. It is generated by a Java Code (https://github.com/feldob/Token-Cluster-Generator, accessed on 1 September 2023). It contains 7 classes and 9344 samples with 2 dimensions.
- Kddcup99 (http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data, accessed on 1 September 2023): Kddcup99 stream is the dataset used in the KDD CUP challenge held in 1999. It consists of 23 classes and 494,021 samples with 41 dimensions.
- CoverType (https://archive.ics.uci.edu/ml/datasets/Covertype, accessed on 1 September 2023): The vegetation coverage type dataset includes four wilderness areas in northern Colorado, USA, located in the Roosevelt National Forest. The total number of samples is 581,012, each with 54 features and 7 types.
- Powersupply (http://www.cse.fau.edu/~xqzhu/stream.html, accessed on 1 September 2023): Powersupply stream contains hourly power supply data of an Italian electricity company which records the power from two sources: power supply from the main grid and power transformed from other grids. This stream contains three-year power supply records from 1995 to 1998. The concept drift in this stream mainly comes from season, weather, hours of a day (e.g., morning and evening), and the differences between working days and weekends. It consists of 29,928 samples, 2 attributes, and 24 classes.
- Sensor (http://www.cse.fau.edu/~xqzhu/stream.html, accessed on 1 September 2023): Sensor stream contains information (temperature, humidity, light, and sensor voltage) collected from 54 sensors deployed in the Intel Berkeley Research Lab. The whole stream contains consecutive information recorded over a 2-month period (1 reading per 1–3 min). It consists of 2,219,803 samples, 5 attributes, and 54 classes.

Table 3. The parameters of the algorithms.

	Dataset	**Letter4**	**Kddcup99**	**CoverType**	**Powersupply**	**Sensor**
FRA-ART	ρ	0.8	0.8	0.8	0.95	0.8
Fuzzy ART	ρ	0.8	0.8	0.8	0.95	0.8
G-Stream	$a_{max} = 250$, $weight_{min} = 2$, $\epsilon_b = 0.01$, $\epsilon_n = 0.001$, $\lambda_1 = 0.2$, $\lambda_2 = 0.2$ $\lvert reservoir \rvert = 400$, $\lvert window \rvert = 600$, and $\beta = 300$					

4.2. Evaluation Criteria

We adopt the common evaluation criteria, including Accuracy (Acc), Normalized Mutual Information (NMI), and the Rand index (RI), to evaluate the performance of clustering, following the approach of [33].

1. Accuracy is calculated by:

$$Acc = \frac{\sum_{i=1}^{K} \frac{|C_i^d|}{|C_i|}}{K} \times 100\% \tag{24}$$

where K denotes the number of clusters. $|C_i^d|$ denotes the number of points with the dominant class label in cluster i. $|C_i|$ denotes the number of points in cluster i. Acc measures the purity of the clusters in terms of the true cluster (class) labels that are known for our datasets, and the value of it is in the range [0, 1].

2. Normalized Mutual Information evaluates the similarity of two clustering results from an information theoretic point of view. The value is in the range [0,1]. A higher value means that more information is shared with the true results, and the clustering performance is better. Unlike Acc, NMI is independent of the absolute values of the labels, a permutation of the class or cluster label values will not change the score value in any way. Considering the two clustering labels, $m^{(a)}$ and $m^{(b)}$, associated with the groups of $k^{(a)}$ and $k^{(b)}$, respectively. We can denote the number of objects in cluster C_h corresponding to $m^{(a)}$ as $n_h^{(a)}$, and the number of objects in cluster C_l corresponding to $m^{(b)}$ as $n_l^{(b)}$. Furthermore, let us denote the number of objects that fall into both clusters C_h and C_l as $n_{h,l}$. Next, the NMI is denoted as:

$$NMI(m^{(a)}, m^{(b)}) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{l=1}^{k^{(b)}} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h^{(a)} n_l^{(b)}}\right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^{(a)}}{n}\right)\left(\sum_{l=1}^{k^{(b)}} n_l^{(b)} \log \frac{n_l^{(b)}}{n}\right)}} \tag{25}$$

3. The Rand index is defined as:

$$RI = \frac{n_{11} + n_{00}}{C_n^2}. \tag{26}$$

where n_{11} is the number of pairs of points that exist in the same cluster in both given correct labels and the clustering result, n_{00} is the number of pairs of points that exist in different subsets for both given correct labels and the clustering result. The value of RI is also in the range [0, 1].

4.3. Comparison on Clustering Performance

We first compare the FRA-ART algorithm with the G-Stream algorithm and Fuzzy ART algorithm, respectively, on various datasets to evaluate the clustering performance of the FRA-ART algorithm. In terms of the SIBFs and CIBFs, we use Equations (13)–(15) and the following five functions:

$$x^{d+j} = g_1(f_1(x^1), f_1(x^2), \ldots, f_1(x^d)) = \max \min_{m \in [1,d]\ m \neq j} (f_1(x^j), f_1(x^m)) \tag{27}$$

$$x^{d+j} = g_1(f_2(x^1), f_2(x^2), \ldots, f_2(x^d)) = \max \min_{m\in[1,d]\ m\neq j} (f_2(x^j), f_2(x^m)) \tag{28}$$

$$x^{d+j} = g_1(f_3(x^1), f_3(x^2), \ldots, f_3(x^d)) = \max \min_{m\in[1,d]\ m\neq j} (f_3(x^j), f_3(x^m)) \tag{29}$$

$$x^{d+j} = g_2(f_1(x^1), f_1(x^2), \ldots, f_1(x^d)) = \sum_{m=1, m\neq j}^{d} (f_1(x^j) f_1(x^m))/(d-1) \tag{30}$$

$$x^{d+j} = g_3(f_1(x^1), f_1(x^2), \ldots, f_1(x^d)) = \sum_{m=1, m\neq j}^{d} \min(f_1(x^j), f_1(x^m))/(d-1) \tag{31}$$

The clustering results are shown in Tables 4–6. From Tables 4–6, we can discover the following noteworthy observations:

(1) The Accuracy values, NMI values and RI values of Fuzzy ART are all higher than those of the G-Stream algorithm except for the Powersupply dataset. However, FRA-ART using fractional order has a different degree of improvement compared with Fuzzy ART on five datasets. Interestingly, FRA-ART with CIBFs achieves much better performance than the counterpart with SIBFs. The reason may be that after the process of normalization and fractionally cross-interactive basis functions, the value of each feature is magnified while also falling between 0 and 1, which combines many individual features and provides information useful for clustering, thus making the FRA-ART more flexible.

(2) As shown in Table 4, the proposed FRA-ART using Equation (28) obtains better performance than Fuzzy ART across letter4, Kddcup99, Powersupply datasets in terms of Accuracy. Surprisingly, on letter4, it achieves the best result 0.9998 with $a = 1/4$, on Kddcup99, it achieves the best result 0.9874 with $a = 1/5$, on Powersupply, it achieves the best result 0.1786 with $a = 1/4$. Meanwhile, the proposed FRA-ART using Equation (31) performs significantly better than Fuzzy ART on CoverType and Sensor. On CoverType, it obtains the best result 0.6014 with $a = 1/2$ when using Equation (13), on Sensor, it obtains the best result 0.0704 with $a = 1/3$ when using Equation (30). The most striking implication of the results in the table is that when using an exponential function, the results are significantly better.

(3) As seen from the Table 5, the NMI value of the proposed FRA-ART using Equation (31) is higher than that of Fuzzy ART on letter4, CoverType, and Sensor. In particular, letter4 obtains the best result 0.8558 with $a = 1/4$, which is about 10% higher than that of Fuzzy ART.

(4) Table 6 shows the RI value of the comparison among three algorithms. The RI value on letter4 is significantly improved except Equation (15). FRA-ART has been improved in Equations (27)–(29) on Kddcup99, and Equations (29)–(31) on CoverType, Powersupply, and Sensor.

(5) As Tables 4 and 6 show, although the Accuracy and RI values of Fuzzy ART are lower than those of G-Stream on Powersupply, the performance of FRA-ART has been greatly improved.

Table 4. The comparison results of FRA-ART and other data stream clustering algorithms measured by Accuracy(Better results in the table are shown in bold).

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
G-Stream	-	0.9515	0.9814	0.5201	0.1752	0.0551
Fuzzy ART	-	0.96	0.9827	0.5935	0.1748	0.0676
FRA-ART (SIBFs using Equation (13))	a = 1/2	**0.9891**	0.9009	**0.6014**	0.1748	0.0616
	a = 1/3	**0.996**	**0.985**	0.573	0.174	0.067
	a = 1/4	**0.972**	0.901	0.569	0.173	0.0608
	a = 1/5	0.9403	**0.9862**	**0.6008**	0.17	0.0606

Table 4. *Cont.*

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
FRA-ART (SIBFs using Equation (14))	a = 1/2	**0.983**	**0.9855**	0.5908	**0.1774**	0.0583
	a = 1/3	**0.9967**	**0.9868**	0.5841	0.174	0.0597
	a = 1/4	**0.9832**	**0.9858**	0.5876	0.1748	0.0584
	a = 1/5	**0.9608**	**0.9874**	0.5914	0.1715	0.0575
FRA-ART (SIBFs using Equation (15))	a = 1/2	0.9599	0.9802	**0.5963**	0.1735	0.0658
	a = 1/3	0.9599	**0.9838**	**0.596**	**0.1755**	0.0665
	a = 1/4	**0.9635**	**0.9849**	**0.5962**	**0.1761**	0.0635
	a = 1/5	**0.9633**	**0.9852**	0.5883	**0.1757**	0.0626
FRA-ART (CIBFs using Equation (27))	a = 1	0.9142	**0.984**	0.5935	0.1746	0.064
	a = 1/2	0.9129	**0.9835**	**0.6014**	**0.1758**	0.0626
	a = 1/3	0.9585	0.9818	0.5734	**0.1757**	0.0652
	a = 1/4	**0.961**	**0.9833**	0.5689	0.173	0.0659
	a = 1/5	**0.973**	0.9822	**0.6008**	0.175	0.0626
FRA-ART (CIBFs using Equation (28))	a = 1	**0.9889**	0.9827	0.5935	0.1749	**0.0677**
	a = 1/2	**0.9605**	**0.9855**	0.5908	**0.1777**	0.0627
	a = 1/3	**0.9991**	**0.9868**	0.5841	**0.1757**	0.061
	a = 1/4	**0.9998**	**0.9858**	0.5876	**0.1786**	0.0657
	a = 1/5	**0.9891**	**0.9874**	0.5914	**0.1766**	0.055
FRA-ART (CIBFs using Equation (29))	a = 1	0.9401	0.8981	0.5726	0.1734	0.0639
	a = 1/2	0.9224	0.9805	**0.5963**	0.1735	0.0663
	a = 1/3	0.9258	**0.9838**	**0.596**	0.1742	0.0668
	a = 1/4	0.9318	**0.9838**	**0.5962**	0.1731	0.0649
	a = 1/5	0.9185	0.9664	0.5883	0.173	**0.0679**
FRA-ART (CIBFs using Equation (30))	a = 1	0.9458	0.9799	0.576	0.1739	0.0663
	a = 1/2	0.9045	**0.9854**	0.5601	**0.1762**	**0.0687**
	a = 1/3	0.9343	**0.9843**	0.5918	0.1733	**0.0704**
	a = 1/4	**0.9652**	0.9168	0.577	0.173	0.0635
	a = 1/5	**0.9891**	**0.9874**	**0.5997**	**0.1756**	0.062
FRA-ART (CIBFs using Equation (31))	a = 1	0.9142	0.9164	0.5833	0.1746	**0.0704**
	a = 1/2	0.9129	0.9533	**0.5978**	**0.1758**	**0.0674**
	a = 1/3	0.935	**0.9842**	0.5838	**0.1757**	0.0636
	a = 1/4	**0.961**	**0.9849**	**0.5959**	0.173	0.0658
	a = 1/5	**0.973**	0.9536	**0.5973**	0.175	**0.0691**

Table 5. The comparison results of FRA-ART and other data stream clustering algorithms measured by NMI(Better results in the table are shown in bold).

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
G-Stream	-	0.5991	0.6449	0.0919	0.1732	0.0679
Fuzzy ART	-	0.7368	0.7347	0.1732	0.186	0.0879
FRA-ART (SIBFs using Equation (13))	a = 1/2	**0.842**	0.6316	0.1709	0.1833	0.078
	a = 1/3	**0.8545**	**0.7365**	0.1477	0.1844	0.0816
	a = 1/4	**0.8162**	0.617	0.1548	0.1854	0.0762
	a = 1/5	**0.8106**	**0.7454**	0.1583	**0.1865**	0.0686
FRA-ART (SIBFs using Equation (14))	a = 1/2	**0.8135**	**0.7835**	0.1558	**0.1865**	0.0704
	a = 1/3	**0.8372**	**0.7756**	0.1465	0.185	0.0733
	a = 1/4	**0.8152**	0.7294	0.1494	**0.1863**	0.0782
	a = 1/5	**0.8457**	**0.7613**	0.1496	0.1859	0.0705

Table 5. *Cont.*

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
FRA-ART (SIBFs using Equation (15))	a = 1/2	**0.7387**	0.7086	0.1593	0.1856	0.0816
	a = 1/3	**0.7381**	**0.748**	0.1676	**0.1878**	0.0835
	a = 1/4	**0.7351**	0.7287	0.165	0.186	0.0813
	a = 1/5	**0.74**	**0.7568**	0.1678	0.1851	0.0767
FRA-ART (CIBFs using Equation (27))	a = 1	0.7308	0.7194	0.1732	0.186	0.0791
	a = 1/2	**0.7392**	0.7292	0.1709	0.1845	0.0749
	a = 1/3	**0.7641**	0.7301	0.1477	0.1829	0.0803
	a = 1/4	**0.796**	**0.7397**	0.1548	0.1834	0.078
	a = 1/5	**0.8235**	0.7209	0.1583	0.1824	0.0753
FRA-ART (CIBFs using Equation (28))	a = 1	**0.7897**	0.7347	0.1732	0.184	0.0866
	a = 1/2	**0.7398**	**0.7835**	0.1558	0.186	0.0795
	a = 1/3	**0.8078**	**0.7756**	0.1465	0.1844	0.0803
	a = 1/4	**0.8558**	0.7294	0.1494	0.1834	0.0838
	a = 1/5	**0.842**	**0.7613**	0.1496	0.1819	0.0693
FRA-ART (CIBFs using Equation (29))	a = 1	**0.7403**	0.6063	0.1624	0.1849	0.0746
	a = 1/2	0.7364	0.7275	0.1593	0.1843	0.0808
	a = 1/3	0.7328	**0.7658**	0.1676	0.1833	0.0798
	a = 1/4	**0.7558**	**0.7468**	0.165	0.1851	0.0769
	a = 1/5	**0.7446**	0.6952	0.1678	0.1857	0.0848
FRA-ART (CIBFs using Equation (30))	a = 1	**0.7738**	0.7165	0.1725	**0.1878**	**0.0939**
	a = 1/2	0.7205	0.7317	**0.1746**	0.185	**0.0971**
	a = 1/3	0.7222	0.7029	**0.1834**	0.1847	**0.0941**
	a = 1/4	**0.7793**	0.6214	0.1718	0.184	**0.0928**
	a = 1/5	**0.842**	**0.7613**	**0.1771**	0.1833	0.0755
FRA-ART (CIBFs using Equation (31))	a = 1	0.7308	0.6226	**0.1849**	0.186	**0.1014**
	a = 1/2	**0.7392**	0.6625	**0.1812**	0.1845	**0.1078**
	a = 1/3	**0.7781**	0.7106	**0.1787**	0.1829	**0.0971**
	a = 1/4	**0.796**	0.7206	**0.1747**	0.1834	0.0852
	a = 1/5	**0.8235**	0.6655	0.1684	0.1824	**0.0932**

Table 6. The comparison results of FRA-ART and other data stream clustering algorithms measured by RI(Better results in the table are shown in bold).

Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
G-Stream	-	0.8095	0.817	0.6158	0.9464	0.8703
Fuzzy ART	-	0.8588	0.9363	0.6492	0.9412	0.9122
FRA-ART (SIBFs using Equation (13))	a = 1/2	**0.9089**	0.8599	0.6482	0.9387	**0.9208**
	a = 1/3	**0.9133**	**0.9455**	0.6394	0.9361	**0.921**
	a = 1/4	**0.9028**	0.856	0.6383	0.9324	0.9119
	a = 1/5	**0.9032**	**0.9494**	0.6371	0.9277	0.8789
FRA-ART (SIBFs using Equation (14))	a = 1/2	**0.8957**	**0.9546**	0.6449	0.933	0.8885
	a = 1/3	**0.8956**	**0.954**	0.6302	0.9314	**0.9208**
	a = 1/4	**0.8901**	0.9352	0.6386	0.9303	0.893
	a = 1/5	**0.9116**	**0.9497**	0.6302	0.927	0.8664
FRA-ART (SIBFs using Equation (15))	a = 1/2	0.8568	0.9253	**0.6493**	0.9422	**0.9166**
	a = 1/3	0.8561	**0.946**	**0.6525**	0.9411	**0.9244**
	a = 1/4	0.8524	**0.9422**	0.6441	0.9376	**0.9201**
	a = 1/5	0.854	**0.9504**	**0.6618**	0.9426	**0.9145**

Table 6. *Cont.*

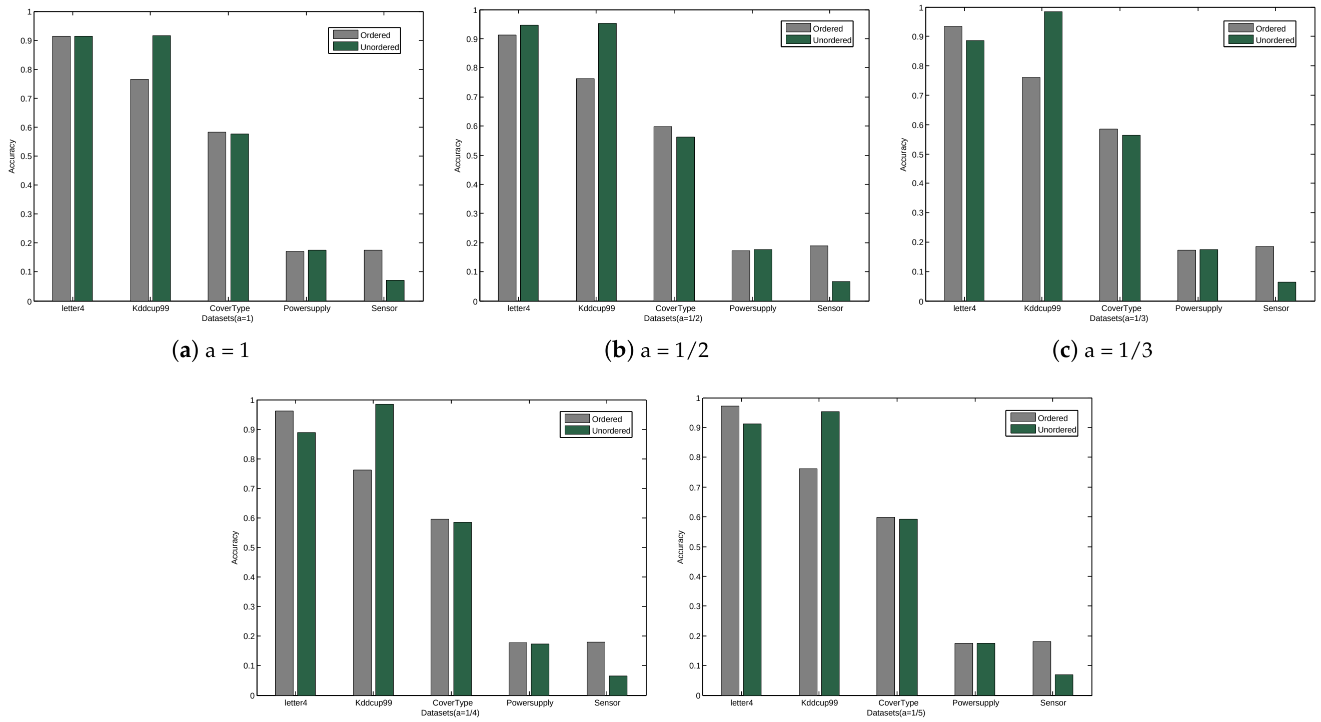
Algorithm	Datasets	Letter4	Kddcup99	CoverType	Powersupply	Sensor
FRA-ART (CIBFs using Equation (27))	a = 1	0.8588	0.9294	0.6492	**0.9476**	**0.9359**
	a = 1/2	**0.8642**	0.9363	0.6482	0.9457	0.8966
	a = 1/3	**0.8781**	**0.9411**	0.6394	0.9447	**0.9233**
	a = 1/4	**0.8893**	**0.9432**	0.6383	0.9405	0.9004
	a = 1/5	**0.9063**	**0.9425**	0.6371	0.9406	0.9039
FRA-ART (CIBFs using Equation (28))	a = 1	**0.8842**	0.9363	0.6492	**0.948**	**0.9137**
	a = 1/2	**0.8619**	**0.9546**	0.6449	0.9438	0.9082
	a = 1/3	**0.8809**	**0.954**	0.6302	0.9434	0.8868
	a = 1/4	**0.9111**	0.9352	0.6386	0.9364	0.9053
	a = 1/5	**0.9089**	**0.9497**	0.6302	0.9372	0.8523
FRA-ART (CIBFs using Equation (29))	a = 1	**0.8626**	0.8418	**0.6542**	0.9458	**0.9275**
	a = 1/2	**0.8654**	0.9341	**0.6493**	**0.948**	**0.9331**
	a = 1/3	**0.8638**	**0.9519**	**0.6525**	**0.9476**	**0.9214**
	a = 1/4	**0.8731**	**0.9426**	0.6441	**0.9506**	**0.9388**
	a = 1/5	**0.8706**	0.9201	**0.6618**	**0.9487**	**0.935**
FRA-ART (CIBFs using Equation (30))	a = 1	**0.8814**	0.9358	**0.6837**	**0.9487**	**0.9395**
	a = 1/2	0.8523	0.9345	**0.6897**	**0.9503**	**0.9176**
	a = 1/3	**0.862**	0.9296	**0.6802**	**0.9486**	**0.9438**
	a = 1/4	**0.8814**	0.8572	**0.6773**	**0.9481**	**0.9343**
	a = 1/5	**0.9089**	**0.9497**	**0.6732**	0.9454	**0.9291**
FRA-ART (CIBFs using Equation (31))	a = 1	**0.8653**	0.8527	**0.6955**	**0.9476**	**0.9365**
	a = 1/2	**0.8642**	0.9034	**0.6801**	0.9457	**0.9492**
	a = 1/3	**0.8847**	0.9321	**0.6838**	0.9447	**0.9245**
	a = 1/4	**0.8893**	0.9343	**0.6579**	0.9405	**0.9299**
	a = 1/5	**0.9063**	0.8973	**0.6566**	0.9406	**0.9249**

4.4. *Effect of Non-Stationarity*

In addition to the above experiments, we also explore the effectiveness of the proposed algorithm in dealing with non-stationary data streams. In many real-world scenarios, the distribution of data changes dynamically, making the data non-stationary. For example, after all data points of the first class have arrived, all data points of the second class have also arrived, and finally the third class, and so on. In this case, with the arrival of new data points, old concepts will disappear, new concepts may be generated, and concept drift will likely occur [52].

4.4.1. Concept Drift

We assess the effectiveness of FRA-ART in handling streaming datasets, which arrive in a sequential manner based on class labels, by utilizing CIBFs for performance evaluation. We use the CIBFs function Equation (31) and set the vigilance parameter $\rho = 0.95$ for the Powersupply dataset and $\rho = 0.8$ for the other datasets. Figures 3–5 display the results of the FRA-ART measured by Accuracy, NMI, and RI, respectively, as the value of parameter a varies, both with and without class label ordering. Our results indicate that the FRA-ART with class label ordering can attain results that are in line with, or even comparable to, those achieved without ordering on the letter4, CoverType, Powersupply, and Sensor datasets. Only on Kddcup99, the Accuracy and NMI of FRA-ART decrease slightly. Furthermore, on letter4 and Sensor, our results demonstrate that the FRA-ART with class label ordering outperforms the version without ordering. With the change in parameter a, the performance of FRA-ART remains consistent. Based on the results presented above, it can be inferred that the FRA-ART effectively handles concept drift during the study, achieving comparable clustering results regardless of whether the classes are arranged in a specific order or not.



(**a**) a = 1

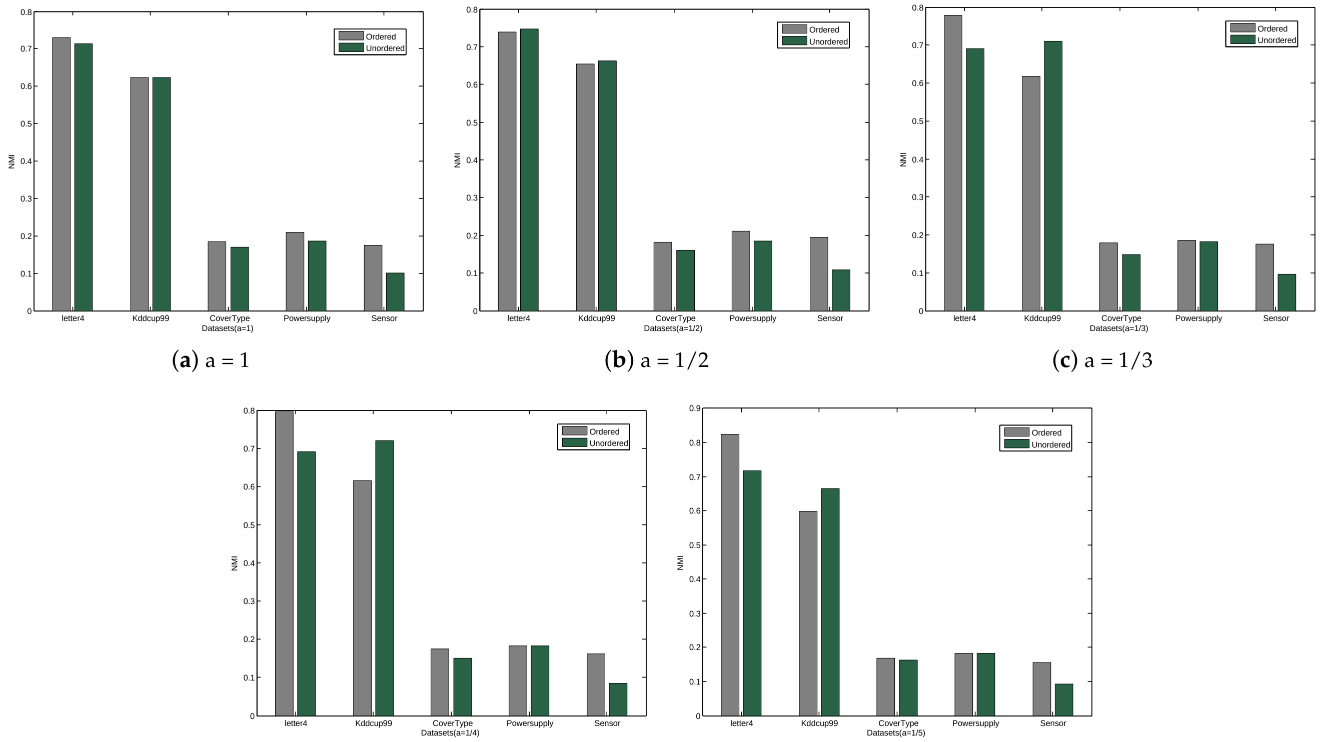
(**b**) a = 1/2

(**c**) a = 1/3

(**d**) a = 1/4

(**e**) a = 1/5

Figure 3. Accuracy of FRA-ART with and without class label ordering.



(**a**) a = 1

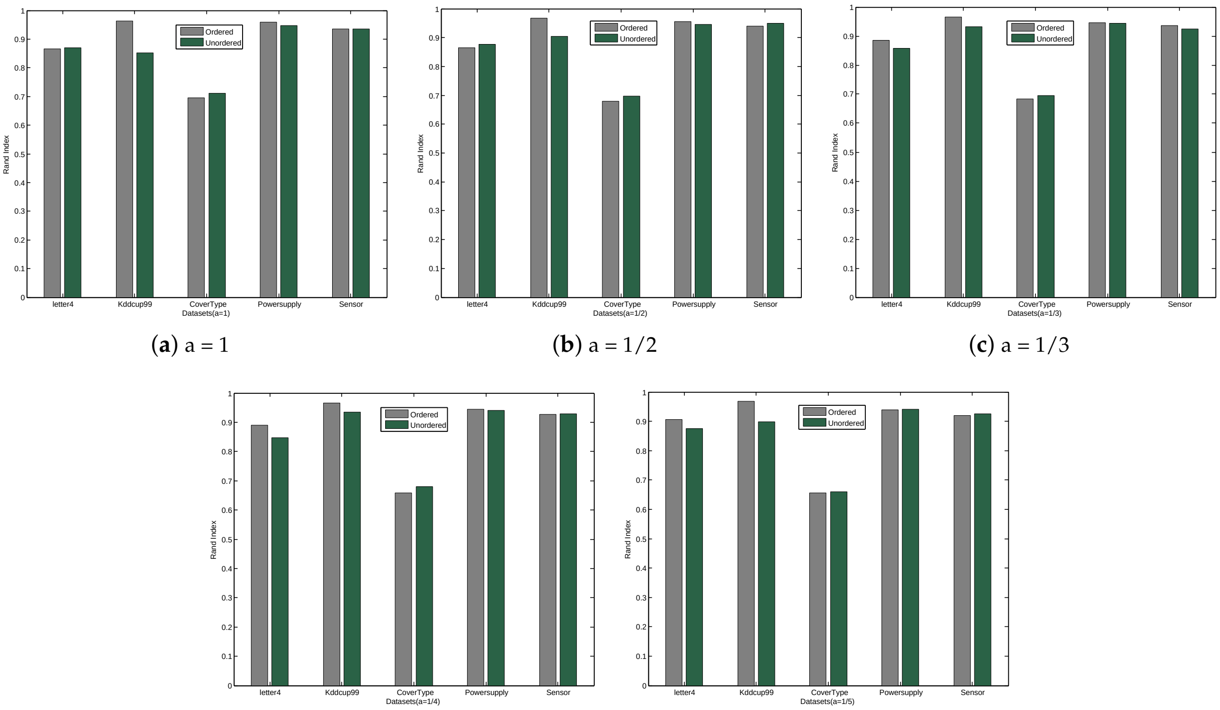
(**b**) a = 1/2

(**c**) a = 1/3

(**d**) a = 1/4

(**e**) a = 1/5

Figure 4. NMI of FRA-ART with and without class label ordering.



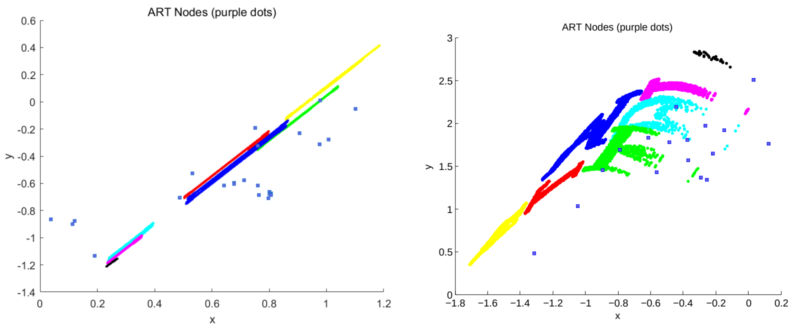
(**a**) a = 1 (**b**) a = 1/2 (**c**) a = 1/3

(**d**) a = 1/4 (**e**) a = 1/5

Figure 5. Rand index of FRA-ART with and without class label ordering.

4.4.2. Visualization

Figure 6 shows the set of nodes by applying Fuzzy ART and FRA-ART on the letter4 dataset (seven colors of points represent data points of the data stream and purple dots are the set of nodes). Since letter4 is a 2-dimensional dataset, with complement coding and CIBFs, the number of dimensions changes to 8. In order to visualize the results in 2-dimensional space, we use random projection [35] to reduce the dimensions and draw the diagram. Meanwhile, by using the same random matrix, the cluster nodes are projected into the same space, as Figure 6 shows. As illustrated in this figure, we can see that the FRA-ART algorithm manages to recognize the clusters of the data stream and can separate these clusters with an optimal visualization. Comparing the two plots in Figure 6, it can be observed that, compared to Fuzzy ART, FRA-ART, through the introduction of SIBFs and CIBFs, has effectively enhanced the data representation capability, leading to tighter clustering of the same class and better separation between different classes. This indicates that the FRA-ART algorithm can better identify the clustering structure of the data stream and provide a more optimal visualization.

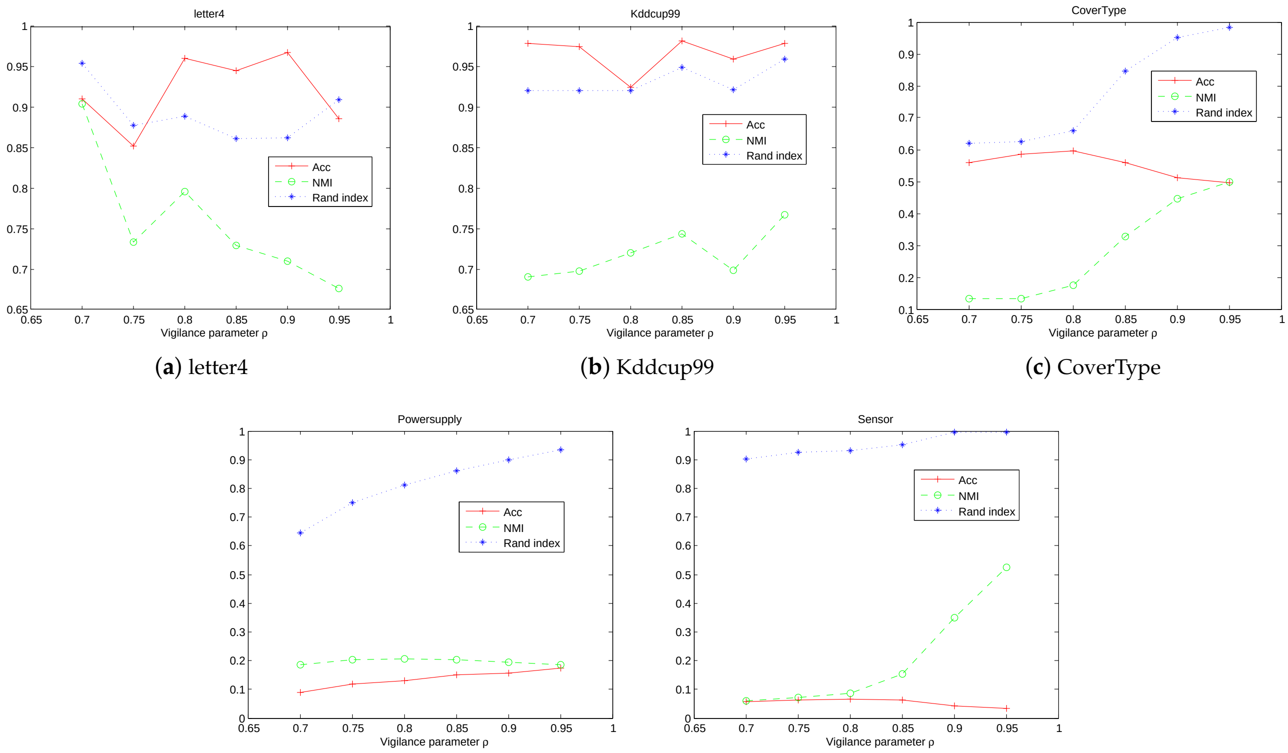


(**a**) Fuzzy ART (**b**) FRA-ART

Figure 6. Nodes are the set of node created by Fuzzy ART and FRA-ART in dataset letter4.

4.5. Effect of Vigilance Parameter

We assess the effectiveness of FRA-ART as the vigilance parameter ρ varies, which controls when resonance happens and whether or not to allow learning samples of that class. Choosing an appropriate vigilance value ρ can identify useful clusters without adjusting many sensitive parameter values. Figure 7 shows the sensitivity of FRA-ART to the vigilance parameter ρ using Accuracy, NMI, and RI on five datasets. We use the CIBFs function Equation (31). As shown in Figure 7, the values of Accuracy, NMI, and RI decrease with increase in ρ on the letter4 and Kddcup99 datasets. The values of NMI and RI increase with increase in ρ on the CoverType and Sensors datasets, but the Accuracy values decrease. On Powersupply, the value of NMI decreases a little bit.



(**a**) letter4 (**b**) Kddcup99 (**c**) CoverType

(**d**) Powersupply (**e**) Sensor

Figure 7. Sensitivity of FRA-ART to the vigilance parameter ρ.

5. Discussion

Through the above experimental comparisons, the proposed FRA-ART algorithm outperforms the traditional Fuzzy ART algorithm and G-Stream algorithm in clustering performance on various datasets. Firstly, the FRA-ART algorithm incorporates fractional order and cross-interactive basis functions (CIBFs) to improve clustering performance. The use of fractional order enables the FRA-ART algorithm to adapt to the changing data distribution, thereby enhancing its ability to handle concept drift in streaming datasets. Moreover, the CIBFs function magnifies the value of each feature, making it easier to combine individual features and provide information useful for clustering. This results in better Accuracy, NMI, and RI values compared with the traditional Fuzzy ART algorithm and G-Stream algorithm. Secondly, the FRA-ART algorithm shows promising results when using different equations. Specifically, on the letter4, CoverType, and Sensor datasets, the NMI value of the proposed FRA-ART using Equation (31) is significantly higher than that of the traditional Fuzzy ART algorithm. On Kddcup99, FRA-ART has been improved in several equations, and on CoverType, Powersupply, and Sensor, FRA-ART has been significantly improved in multiple equations. Lastly, the use of SIBFs and CIBFs is attractive because it allows different shapes of segmentation to be implemented with low computational cost.

Furthermore, the underlying recursive partitioning algorithm is not modified, which means IBFs can be used with any existing ART algorithm variant software implementation.

Despite its advantages, the FRA-ART algorithm has some limitations that need to be addressed in future research. Firstly, the FRA-ART algorithm requires careful selection of the vigilance parameter ρ and the parameter a to achieve optimal clustering performance. Future research can focus on developing more efficient methods for automatically selecting these parameters. Secondly, the FRA-ART algorithm may not be suitable for high-dimensional datasets. In the future, the FRA-ART algorithm can be extended to handle high-dimensional data by incorporating dimensionality reduction techniques.

In conclusion, the proposed FRA-ART algorithm shows promising results in clustering performance on various datasets, demonstrating its ability to handle concept drift in streaming datasets. However, further research is needed to address its limitations and improve its computational efficiency. Furthermore, Multiview data stream clustering is an emerging topic to be explored further.

6. Conclusions

In this paper, we propose FRA-ART, a novel unsupervised self-organizing incremental neural network for clustering evolving data streams. The contributions of this paper are three-fold. Firstly, we introduce the fractional order operation into ART to make its representation more flexible and richer. Secondly, for the first time, the fractionally self-interactive or cross-interactive basis functions (SIBFs or CIBFs) are employed for data stream clustering. Thirdly, the proposed SIBFs and CIBFs can be precomputed once independently and thus used in any variants of the ART. Finally, comparative experiments on five data stream sets show that FRA-ART is more robust than Fuzzy ART, as well as the state-of-the-art data stream clustering algorithm G-Stream, and it consistently achieves results better or comparable to their performances.

Author Contributions: Methodology, Y.Z. (Yingwen Zhu); Software, P.L. and Q.Z.; Validation, Y.Z. (Yi Zhu) and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation: 62206114 and 62006126, the Natural Science Foundation of Jiangsu Province: BK20200740, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China: 22KJB110012, the Jiangsu Province Education Science Planning Project: B/2023/01/50, and the Higher Education Teaching Reform Project of Jiangsu Province: 2023JSJG690.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Zhang, Q.; Zhu, Y.; Yang, M.; Jin, G.; Zhu, Y.; Chen, Q. Cross-to-merge training with class balance strategy for learning with noisy labels. *Expert Syst. Appl.* **2024**, *249*, 123846. [CrossRef]
2. Jin, G.; Liu, C.; Chen, X. An efficient deep neural network framework for COVID-19 lung infection segmentation. *Inf. Sci.* **2022**, *612*, 745–758. [CrossRef] [PubMed]
3. Silva, J.A.; Faria, E.R.; Barros, R.C.; Hruschka, E.R.; de Carvalho, A.C.; Gama, J. Data stream clustering: A survey. *ACM Comput. Surv.* **2013**, *46*, 1–13. [CrossRef]
4. Li, Y.; Yang, G.; He, H.; Jiao, L.; Shang, R. A study of large-scale data clustering based on fuzzy clustering. *Soft Comput.* **2016**, *20*, 3231–3242. [CrossRef]
5. Zhang, P.; Shen, Q. Fuzzy c-means based coincidental link filtering in support of inferring social networks from spatiotemporal data streams. *Soft Comput.* **2018**, *22*, 7015–7025. [CrossRef]
6. Stratos, M.; Eirini, N.; Nikos, P.; Yannis, T. An evaluation of data stream clustering algorithms. *Stat. Anal. Data Min.* **2018**, *11*, 167–187.
7. Laurinec, P.; Lucká, M. Interpretable multiple data streams clustering with clipped streams representation for the improvement of electricity consumption forecasting. *Data Min. Knowl. Discov.* **2019**, *33*, 413–445. [CrossRef]
8. Forestiero, A.; Pizzuti, C.; Spezzano, G. A single pass algorithm for clustering evolving data streams based on swarm intelligence. *Data Min. Knowl. Discov.* **2013**, *26*, 1–26. [CrossRef]

9. Appice, A.; Ciampi, A.; Malerba, D. Summarizing numeric spatial data streams by trend cluster discovery. *Data Min. Knowl. Discov.* **2015**, *29*, 84–136. [CrossRef]
10. Wang, H.B.; Hui, X.B.; Lin, J.F. The research of data stream mining and application in fault diagnosis of equipment. In *Mechanical Engineering and Control Systems: Proceedings of the 2016 International Conference on Mechanical Engineering and Control System (MECS2016), Wuhan, China, 27–29 October 2016*; World Scientific: Singapore, 2017; pp. 101–107.
11. Souza, V.M.; dos Reis, D.M.; Maletzke, A.G.; Batista, G.E. Challenges in benchmarking stream learning algorithms with real-world data. *Data Min. Knowl. Discov.* **2020**, *34*, 1805–1858. [CrossRef]
12. Huang, L.; Wang, C.D.; Chao, H.Y.; Yu, P.S. MVStream: Multiview Data Stream Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3482–3496. [CrossRef] [PubMed]
13. Zubaroğlu, A.; Atalay, V. Data stream clustering: A review. *Artif. Intell. Rev.* **2021**, *54*, 1201–1236. [CrossRef]
14. Zhou, Z.H. Stream efficient learning. *arXiv* **2023**, arXiv:2305.02217.
15. O'callaghan, L.; Mishra, N.; Meyerson, A.; Guha, S.; Motwani, R. Streaming-data algorithms for high-quality clustering. In Proceedings of the International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2002; IEEE: Piscataway, NJ, USA, 2002; pp. 685–694.
16. Ackermann, M.R.; Märtens, M.; Raupach, C.; Swierkot, K.; Lammersen, C.; Sohler, C. Streamkm++ a clustering algorithm for data streams. *J. Exp. Algorithm.* **2012**, *17*, 2.1–2.3.
17. Puschmann, D.; Barnaghi, P.; Tafazolli, R. Adaptive Clustering for Dynamic IoT Data Streams. *IEEE Internet Things J.* **2016**, *4*, 64–74. [CrossRef]
18. de Andrade Silva, J.; Hruschka, E.R.; Gama, J. An evolutionary algorithm for clustering data streams with a variable number of clusters. *Expert Syst. Appl.* **2017**, *67*, 228–238. [CrossRef]
19. Udommanetanakit, K.; Rakthanmanon, T.; Waiyamai, K. E-stream: Evolution-based technique for stream clustering. In Proceedings of the International Conference on Advanced Data Mining and Applications, Harbin, China, 6–8 August 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 605–615.
20. Zhou, A.; Cao, F.; Qian, W.; Jin, C. Tracking clusters in evolving data streams over sliding windows. *Knowl. Inf. Syst.* **2008**, *15*, 181–214. [CrossRef]
21. Kranen, P.; Assent, I.; Baldauf, C.; Seidl, T. The clustree: Indexing micro-clusters for anytime stream mining. *Knowl. Inf. Syst.* **2011**, *29*, 249–272. [CrossRef]
22. Kriegel, H.P.; Kröger, P.; Ntoutsi, I.; Zimek, A. Density based subspace clustering over dynamic data. In Proceedings of the International Conference on Scientific and Statistical Database Management, Portland, OR, USA, 20–22 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 387–404.
23. Tasoulis, D.K.; Ross, G.; Adams, N.M. Visualising the cluster structure of data streams. In Proceedings of the International Symposium on Intelligent Data Analysis, Ljubljana, Slovenia, 6–8 September 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 81–92.
24. Cao, F.; Estert, M.; Qian, W.; Zhou, A. Density-based clustering over an evolving data stream with noise. In Proceedings of the 2006 SIAM International Conference on Data Mining, Bethesda, MD, USA, 20–22 April 2006; SIAM: Philadelphia, PA, USA, 2006; pp. 328–339.
25. Hyde, R.; Angelov, P.; Mackenzie, A.R. Fully online clustering of evolving data streams into arbitrarily shaped clusters. *Inf. Sci.* **2016**, *382–383*, 96–114. [CrossRef]
26. Hahsler, M.; Bolaños, M. Clustering data streams based on shared density between micro-clusters. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1449–1461. [CrossRef]
27. Yin, C.; Xia, L.; Zhang, S.; Sun, R.; Wang, J. Improved clustering algorithm based on high-speed network data stream. *Soft Comput.* **2018**, *22*, 4185–4195. [CrossRef]
28. Fahy, C.; Yang, S.; Gongora, M. Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams. *IEEE Trans. Cybern.* **2018**, *49*, 2215–2228. [CrossRef] [PubMed]
29. Wan, L.; Ng, W.K.; Dang, X.H.; Yu, P.S.; Zhang, K. Density-based clustering of data streams at multiple resolutions. *ACM Trans. Knowl. Discov. Data* **2009**, *3*, 14. [CrossRef]
30. Chen, Y.; Tu, L. Density-based clustering for real-time stream data. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; ACM: New York, NY, USA, 2007; pp. 133–142.
31. Amini, A.; Saboohi, H.; Herawan, T.; Wah, T.Y. MuDi-Stream: A multi density clustering algorithm for evolving data stream. *J. Netw. Comput. Appl.* **2016**, *59*, 370–385. [CrossRef]
32. Smith, T.; Alahakoon, D. Growing self-organizing map for online continuous clustering. In *Foundations of Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 4, pp. 49–83.
33. Ghesmoune, M.; Lebbah, M.; Azzag, H. A new growing neural gas for clustering data streams. *Neural Netw.* **2016**, *78*, 36–50. [CrossRef] [PubMed]
34. Dang, X.H.; Lee, V.; Ng, W.K.; Ciptadi, A.; Ong, K.L. An EM-based algorithm for clustering data streams in sliding windows. In Proceedings of the International Conference on Database Systems for Advanced Applications, Brisbane, Australia, 21–23 April 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 230–235.

35. Zhu, Y.; Chen, S. Growing neural gas with random projection method for high-dimensional data stream clustering. *Soft Comput.* **2020**, *24*, 9789–9807. [CrossRef]
36. Wunsch, D.C. Admiring the Great Mountain: A Celebration Special Issue in Honor of Stephen Grossberg's 80th Birthday. *Neural Netw.* **2019**, *120*, 1–4. [CrossRef] [PubMed]
37. Carpenter, G.A.; Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput. Vision Graph. Image Process.* **1987**, *37*, 54–115. [CrossRef]
38. Carpenter, G.A.; Grossberg, S. ART 2: Self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.* **1987**, *26*, 4919–4930. [CrossRef]
39. Carpenter, G.A.; Grossberg, S.; Rosen, D.B. Art 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Netw.* **1991**, *4*, 493–504. [CrossRef]
40. Carpenter, G.A.; Grossberg, S. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Netw.* **1990**, *3*, 129–152. [CrossRef]
41. Carpenter, G.A.; Grossberg, S.; Rosen, D.B. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Netw.* **1991**, *4*, 759–771. [CrossRef]
42. Williamson, J.R. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Netw.* **1996**, *9*, 881–897. [CrossRef] [PubMed]
43. Anagnostopoulos, G.C.; Georgiopulos, M. Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 24–27 July 2000; Volume 6, pp. 59–64.
44. Tscherepanow, M. TopoART: A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece, 15–18 September 2010; pp. 157–167.
45. Silva, L.E.B.D.; Elnabarawy, I.; Wunsch, I.D.C. Distributed dual vigilance fuzzy adaptive resonance theory learns online, retrieves arbitrarily-shaped clusters, and mitigates order dependence. *Neural Netw.* **2020**, *121*, 208–228. [CrossRef]
46. Silva, L.E.B.D.; Elnabarawy, I.; Wunsch, D.C. Dual vigilance fuzzy adaptive resonance theory. *Neural Netw.* **2019**, *109*, 1–5. [CrossRef] [PubMed]
47. Meng, L.; Tan, A.; Wunsch, D.C. Adaptive Scaling of Cluster Boundaries for Large-Scale Social Media Data Clustering. *IEEE Trans. Neural Netw.* **2016**, *27*, 2656–2669. [CrossRef] [PubMed]
48. Meng, L.; Tan, A.; Miao, C. Salience-aware adaptive resonance theory for large-scale sparse data clustering. *Neural Netw.* **2019**, *120*, 143–157. [CrossRef]
49. Paez, A.; Lopez, F.A.; Ruiz, M.; Camacho, M. Inducing non-orthogonal and non-linear decision boundaries in decision trees via interactive basis functions. *Expert Syst. Appl.* **2019**, *122*, 183–206. [CrossRef]
50. Da Silva, L.E.B.; Elnabarawy, I.; Wunsch Ii, D.C. A survey of adaptive resonance theory neural network models for engineering applications. *Neural Netw.* **2019**, *120*, 167–203. [CrossRef]
51. Granger, E.; Savaria, Y.; Lavoie, P.; Cantin, M.A. A comparison of self-organizing Neural Netw. for fast clustering of radar pulses. *Signal Process.* **1998**, *64*, 249–269. [CrossRef]
52. Webb, G.I.; Hyde, R.; Cao, H.; Nguyen, H.L.; Petitjean, F. Characterizing concept drift. *Data Min. Knowl. Discov.* **2016**, *30*, 964–994. [CrossRef]

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