[Array 19 (2023) 100319](https://doi.org/10.1016/j.array.2023.100319)



Contents lists available at [ScienceDirect](https://www.elsevier.com/locate/array)

Array

journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array)

SSFuzzyART: A Semi-Supervised Fuzzy ART through seeding initialization and a clustered data generation algorithm to deeply study clustering solutions

Siwar Jendoubi [∗](#_bookmark0), Aurélien Baelde, Thong Tran

*UPSKILLS R&D, 16 Rue Marc Sangnier 94600, Choisy-Le-Roi, France*[1](#_bookmark1)

A R T I C L E I N F O A B S T R A C T

*Keywords:*

Semi-supervised learning FuzzyART

Initialization by seeding Clustering

SSFuzzyART

Clustered binary data generation

Semi-supervised clustering is a machine learning technique that was introduced to boost clustering performance when labeled data is available. Indeed, some labeled data are usually available in real use cases, and can be used to initialize the clustering process to guide it and to make it more efficient. Fuzzy ART is a clustering technique that is proved to be efficient in several real cases, but as an unsupervised algorithm, it cannot use available labeled data. This paper introduces a semi-supervised variant of the FuzzyART clustering algorithm (SSFuzzyART). The proposed solution uses the available labeled data to initialize clusters centers. In another hand, to deeply evaluate the characteristics of the proposed algorithm, a clustered binary data generation algorithm with controlled partitioning is also introduced in this paper. Indeed, the controlled generated clusters allows studying the characteristics of the proposed SSFuzzyART. Furthermore, a set of experiments is carried out on some available benchmarks. SSFuzzyART demonstrated better clustering prediction results than its classic counterpart.

# Introduction

Clustering techniques are useful to group similar unlabeled data instances together. It is used to explore unlabeled datasets and to discover similarity relationships between its instances. In some real cases, it is possible to have access to labeled data instances. These instances are, usually, not enough to learn a classifier. In fact, the available labels could represent a subset of possible labels, on only a tiny fraction of the dataset, preventing the training of a learner. However, usual clustering algorithms are not designed to consider this information about labels. In such cases, semi-supervised techniques can use the labeled instances to guide the clustering process.

Fuzzy Adaptive Resonance Theory (Fuzzy ART) is a classic cluster- ing algorithm. It was introduced by Carpenter et al. [[1](#_bookmark39)]. Fuzzy ART is

an unsupervised learning algorithm that incorporates the min operator

from the fuzzy set theory into an ART based neural network. This

algorithm can learn stable clusters from continuous input patterns. Besides, it is effective for large scaled data, and it runs fast compared to other clustering algorithms [[2](#_bookmark40),[3](#_bookmark41)]. These characteristics make Fuzzy ART able to detect clusters with different sizes. Then, small clusters, having few instances, could be detected and separated from larger clusters.

Fuzzy ART is a clustering algorithm that demonstrated good per- formance in many real cases [[2](#_bookmark40),[4](#_bookmark42),[5](#_bookmark43)]. However, in the case where some labeled data instances are available, Fuzzy ART cannot con- sider this information to improve the clustering quality. This paper presents a simple and efficient solution to initialize Fuzzy ART with available labeled data instances. Indeed, the proposed approach is a semi-supervised Fuzzy ART (SSFuzzyART) that considers available labeled data instances to initialize clusters.

Binary data is data composed of vectors of arbitrary size whose categories are drawn in a two-valued distribution only. With modalities 0 or 1, a vector of binary data can be 010111001010. They are used in various fields of application and to model several types of data or phenomena. For example, a text document can be modeled by a binary vector representing the presence or absence of terms or characters. In addition, several automatic learning techniques have been introduced for the exploitation of this particular data, [[6](#_bookmark44)–[9](#_bookmark46)]. In this article, we consider the binary controlled clustered data to deeply study the char- acteristics of the proposed SSFuzzyART algorithm. For that purpose, we need to be able to manage the data characteristics and clusters separability to evaluate the SSFuzzyART algorithm. Binary clustered data is adaptable for such cases. In fact, binary data has only two

∗ Corresponding author.

*E-mail addresses:* [siwar.jendoubi@upskills.ai](mailto:siwar.jendoubi@upskills.ai) (S. Jendoubi), [thong.tran@upskills.ai](mailto:thong.tran@upskills.ai) (T. Tran).

1 [http://www.upskills.com](http://www.upskills.com/)

<https://doi.org/10.1016/j.array.2023.100319>

Received 14 March 2023; Received in revised form 30 August 2023; Accepted 6 September 2023

Available online 9 September 2023

2590-0056/© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/) [nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

modalities, which make it easier to manage and generated clusters are better controlled.

In this article, an algorithm for generating binary data grouped by clusters of controlled variances is introduced. This algorithm allows the generation of binary data sets having characteristics known and mastered beforehand. These datasets are used to test and validate the effectiveness of binary data partitioning solutions. Being able to control the characteristics of the data generated also makes it possible to study the applicability of these methods and their weaknesses. These char- acteristics are also useful for choosing a machine learning technique. Indeed, the choice will be guided by the characteristics of the data and tests in circumstances similar to real cases. In addition, a study comparing dimension reduction algorithms was carried out to show the usefulness of the generated data in the choice of a dimension reduction algorithm, under the angle of the conservation of the characteristics of the clusters in space.

The contributions of this paper are the following: (1) the proposition of a semi-supervised version of the Fuzzy ART, SSFuzzyART, clustering algorithm. SSFuzzyART uses a set of available labeled instances to initialize the clusters centers. Also, a set of experiments is made to eval- uate the proposed algorithm and to study its strength and weakness.

(2) the proposition of a data generation algorithm and the study of the characteristics of the generated data and the use of the generated data to deeply study SSFuzzyART algorithm characteristics.

This paper is organized as follows: Section [2](#_bookmark2) presents an overview of the current state of the art on the semi-supervised clustering, Fuzzy ART and clustering data generation. Section [3](#_bookmark5) details the classic Fuzzy ART algorithm and the proposed SSFuzzyART. Section [4](#_bookmark12) presents the clustered binary data generation algorithm, and evaluates the gener- ated clusters through a set of experiments. Section [5](#_bookmark25) introduces the considered evaluation measures, evaluates the characteristics of SS- FuzzyART using the generated data, compares SSFuzzyART clustering to FuzzyART clustering and discusses the experiment results. Finally, Section [6](#_bookmark38) concludes the paper.

# Related works

* 1. *Semi-supervised clustering*

Semi-supervised clustering combines elements of both semi-super vised learning and clustering analysis. By incorporating class labels, semi-supervised clustering directs the clustering procedure, leading to improve performance. Over the past years, many remarkable initiatives have surfaced, focusing on both exploration and practical implementa- tion in various fields. [[10](#_bookmark47)].

Adaptive Resonance Theory (ART) and Semi-supervised learning has attracted several researchers’ attention. Kim et al. [[11](#_bookmark48)] introduced a label propagation ART algorithm. It is a semi-supervised approach. Another semi-supervised Message Passing Adaptive Resonance The- ory (MPART), [[12](#_bookmark49)], was also introduced. Recently, [[13](#_bookmark50)] proposed HANFA-ART with ACA which is a Hybrid Adaptive Neural-Fuzzy Al- gorithm based on Adaptive Resonant Theory with Adaptive clustering Algorithm. This model has nine major components, and the Adaptive Resonant Theory is one of them. Another interesting clustering solution was recently introduced by Masuyama et al. [[14](#_bookmark51)]. This work proposes a solution for better estimating the specification of a similarity threshold (vigilance parameter) which is an interesting parameter on which the prediction of the ART based models is highly dependent.

Fuzzy Adaptive Resonance Theory algorithm is a classic online clustering algorithm that is known to be sensitive to data order. To remedy this issue, several research papers introduced solutions for this problem. Elnabarawy et al. [[15](#_bookmark52)] used evolutionary computation to improve Fuzzy ART. Liew et al. [[4](#_bookmark42)] used a genetic algorithm for the same purpose.

Clusters initialization by seeding and semi-supervised clustering are usual machine learning techniques that are always used to make the

clustering prediction more efficient. Basu et al. [[16](#_bookmark53)] used labeled data to initialize clusters for k-means algorithm. They assume that for each possible cluster in a given problem, they have at least one instance belonging to that cluster. Besides, they initialize each cluster with the mean of instances belonging to that cluster. Taghizabet et al. [[17](#_bookmark54)] proposed ‘‘ConvexClust’’. It is a semi supervised clustering k-means based solution that uses labeled data and the geometric point of view to improve the clustering results.

According to [[18](#_bookmark55),[19](#_bookmark56)] that present detailed surveys about the semi- supervised clustering, most available semi-supervised clustering solu- tions consider the k-means clustering approach. However, to the best of our knowledge, there is no existing work that focus on a semi- supervised FuzzyART clustering approach. Indeed, Fuzzy ART is a specific clustering algorithm that showed its performance to detect clusters with different sizes and to overcome the case of imbalanced data. Then, a semi-supervised FuzzyART inherit from Fuzzy ART its characteristics and improves its results using labeled data.

* 1. *Clustered data generation*

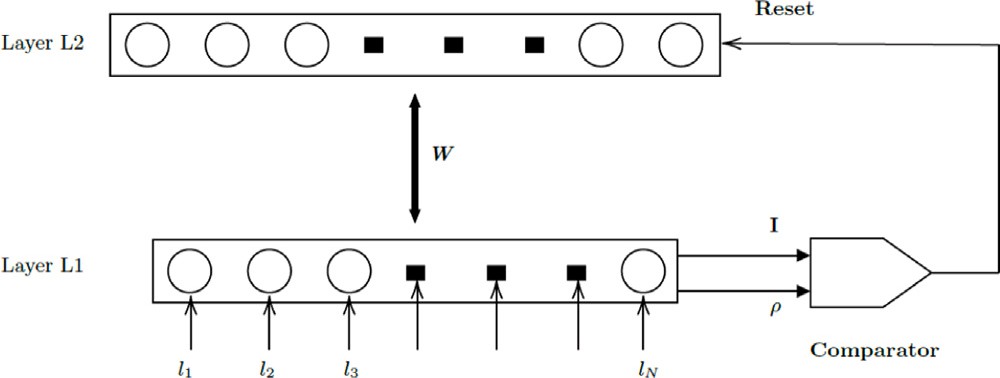
Data is ubiquitous and is often a resource in various situations. However, generating such a value depends on higher quality data. In addition, in situations involving sensitive confidential data such as business data or background, it is imperative to maintain data confi- dentiality by adhering to quality standards. The demand for top-notch data and privacy-friendly practices has become increasingly evident in recent years, as companies and researchers use it more extensively.

The literature presents several approaches of synthetic data gen- eration for clustering like [[20](#_bookmark57)–[24](#_bookmark59)] are introduced. Each of these ap- proaches is generally trying to replicate some real world phenomena and to generate the needed data to evaluate and validate some specific solutions. In addition, the generated data could have some special characteristics related to the generated clusters shapes or features, for example. Let us consider the work of [[24](#_bookmark59)]. It presents clustered data generation approach called ‘‘repliclust’’ that allows to generate clusters with different geometric characteristics. However, their solution is not useful to generate binary data and does not allow managing clusters sparsity.

Several binary data generation algorithms, [[23](#_bookmark58),[25](#_bookmark60)], aim to sim- ulate correlation phenomena between binary variables. These works seek to simulate a phenomenon observed in several fields of research such as the correlation between genetic variables in bioinformatics, binary questionnaires (yes/no) carried out for sociological studies, etc. Several algorithms for generating correlated binary modalities with varied marginal probabilities have been introduced, [[23](#_bookmark58),[25](#_bookmark60)]. The main objective of these algorithms is to be able to generate binary data with high dimensions and with variables presenting controlled correlations between them. These data generation algorithms are suitable for the study of data correlation. However, the data generated with these solutions is not suitable for the study of data clustering approaches.

Methods for generating longitudinally correlated binary data have been studied by [[26](#_bookmark61)]. These methods are useful for simulating large empirical surveys resulting in correlated longitudinal binary data where some dichotomous outcomes are measured on the same experimental units over time. For example, such data frequently arise in clinical trials where the same independent set of patients may be assessed for the presence or absence of a particular disease at different time points. The generation of this data considers the ‘‘cluster’’ aspect in the generation process. A cluster/group is defined by the fact that the measures within each cluster are correlated, but the measures of different clusters are independent. These generation methods are different from our solution. Indeed, these approaches do not control intra- and inter-cluster distances, but rather the correlation of data in each cluster.

Synthetic binary data has been used to experiment and validate clustering algorithms, like in the work of [[8](#_bookmark45)]. However, the generated



**/ig. 1.** Fuzzy ART architecture.

Where ∧ is the Fuzzy AND operator defined as: *𝑥* ∧ *𝑦* = min(*𝑥, 𝑦*), *𝑥* is defined as *𝑥* = ∑*𝑘 𝑥𝑘*, *𝛼 >* 0 is a parameter of Fuzzy ART, and *𝑤𝑗* is the center of the *𝑗* category (neuron). After computing the score, *𝑇𝑗* , for all neurons in the layer *𝐿*2, the neuron, *𝐽* , having the maximum score

is selected:

| |

| |

*𝑇𝐽* = max(*𝑇𝑗* ) (2)

*𝑗*

Once a neuron (category) is selected from the layer *𝐿*2, Fuzzy ART checks the expectation between the input instance, *𝐼* , and the selected category. This expectation is compared to the vigilance threshold, *𝜌*, as

follows:

data used did not allow significant experimentation in terms of data

*𝜌* ≤ *𝐼* ∧ *𝑤𝐽*

*𝐼*

| |

(3)

partitioning. Indeed, the binary data generation method used only allowed the generation of high-dimensional data, but these data do not present detectable clusters. Thus, these datasets only allowed the study of the scalability of the proposed clustering algorithm.

To the best of our knowledge, the literature does not present a clustered data generation solution that could be useful to analyze all the characteristics of the proposed SSFuzzyART algorithm or that allows to generate similar clustered binary data to the generated data using the proposed algorithm. Our need for clustered binary data with the possibility to manage the intra and inter clusters distances and the data sparsity motivated us to introduce the data generation algorithm proposed in Section [4](#_bookmark12)

# Semi-supervised /uzzy ART (SS/uzzyART)

In real world problems, some labeled data instances may be avail- able. Fuzzy ART algorithm is not able to consider this available data in its clustering process. To resolve this problem and make Fuzzy ART output more efficient in such cases, this paper introduces the SSFuzzyART algorithm.

This section details first the Fuzzy ART algorithm. Next, it intro- duces the SSFuzzyART.

* 1. *Fuzzy ART algorithm*

Fuzzy ART [[1](#_bookmark39)] is a classic clustering algorithm that extends the ART1 algorithm. Indeed, ART1 algorithm categorizes only binary in- put instances. Then, Fuzzy ART extends this algorithm to categorize both binary and analog instances. For each input instance, the Fuzzy ART network looks for the ‘‘nearest’’ cluster that ‘‘resonates’’ with that instance. Next, it updates the center of the selected cluster with

the input instance. The vigilance parameter, *𝜌* ∈ [0*,* 1], is an input

parameter of Fuzzy ART that determines the similarity of instances in

the same cluster. When the vigilance parameter increases, the similarity of instances in the same cluster grows, which leads to a larger number of clusters, [[27](#_bookmark62)].

Fuzzy ART network is composed of two layers *𝐿*1 and *𝐿*2. [Fig.](#_bookmark3) [1](#_bookmark3)

shows the architecture of Fuzzy ART network. The layer *𝐿*1 is the input

the input instance. The layer *𝐿*2 is the category representation layer. presentation layer. It has as many neurons as the number of features in Then, neurons in the *𝐿*2 layer are added while the learning progresses.

the *𝐿*2 layer. Besides, the layer *𝐿*2 maintains one additional neuron in Indeed, for each new detected category, a new neuron is appended to

the uncommitted state. The uncommitted neuron is useful to maintain the vigilance criterion (equation ([3](#_bookmark4))) satisfied when a new category arises [[28](#_bookmark63)].

When a new data instance, *𝐼* , is presented in the layer *𝐿*1, each

neuron in the layer *𝐿*2 computes a score for the category choice, using

the following equation:

with the instance *𝐼* using the following equation: If this condition is satisfied, then the category neuron center is updated

*𝑤𝐽* = *𝛽*(*𝐼* ∧ *𝑤𝐽* ) + (1 − *𝛽*)*𝑤𝐽* (4)

Where *𝛽* ∈ [0*,* 1] is the learning rate.

If the condition of the formula ([3](#_bookmark4)) is not satisfied, Fuzzy ART re-

runs the neuron selection process until finding a neuron that satisfies the expectation condition (equation ([3](#_bookmark4))), next, it updates that neuron using formula ([4](#_bookmark6)). In the case where no committed neuron fits with the expected condition, then, the data instance represents a new category and the uncommitted neuron is selected and updated to fit the new detected category, then, its weight vector is initialized with the input data instance. Next, a new uncommitted neuron is created. This process is repeated on all available data instances until getting a stable partition in which no instance changes its category. To limit the running time, a maximum number of epochs could be defined.

The next section introduces the proposed semi-supervised Fuzzy ART algorithm.

* 1. *SSFuzzyART algorithm*

The semi-supervised Fuzzy ART (SSFuzzyART) is introduced to con- sider the case where some labeled data instances are available. These instances may represent all possible labels in a given problem, or just a subset of these labels. Consequently, this data is useful to make Fuzzy ART clustering more efficient and to get better quality results.

The SSFuzzyART has the same steps as the classic Fuzzy ART algorithm, and it differs from it in the algorithm initialization. In fact,

Fuzzy ART starts the clustering process with one neuron in the *𝐿*2 layer,

which is the commitment neuron. This initialization makes it sensitive

to the order of data entry. Indeed, the first coming instances will be used to initialize the clusters and then, will influence the clustering

the categories in the *𝐿*2 layer. Then, each available category in the performance. The SSFuzzyART uses the labeled instances to initialize

labeled data will be used to initialize its corresponding cluster, as shown in [Fig.](#_bookmark8) [2](#_bookmark8).

For a given category in the labeled data, a neuron in the *𝐿*2 layer

is created to represent that category. Then, available data instances

will be used to initialize the created neuron by computing its weights. Consequently, weights are computed by applying Eq. ([4](#_bookmark6)) on the set of data instances for a given category. [Table](#_bookmark10) [1](#_bookmark10) presents a computation example.

Three possible scenarios for the use of the labeled instances after the initialization step are possible:

neurons (layer *𝐿*2). Next, it is no longer used in the clustering 1. In the first scenario, the labeled data is used to initialize category

process.

category neurons (layer *𝐿*2). Next, their labels are discarded, 2. In the second scenario, the labeled data is used to initialize

*𝑇𝑗*

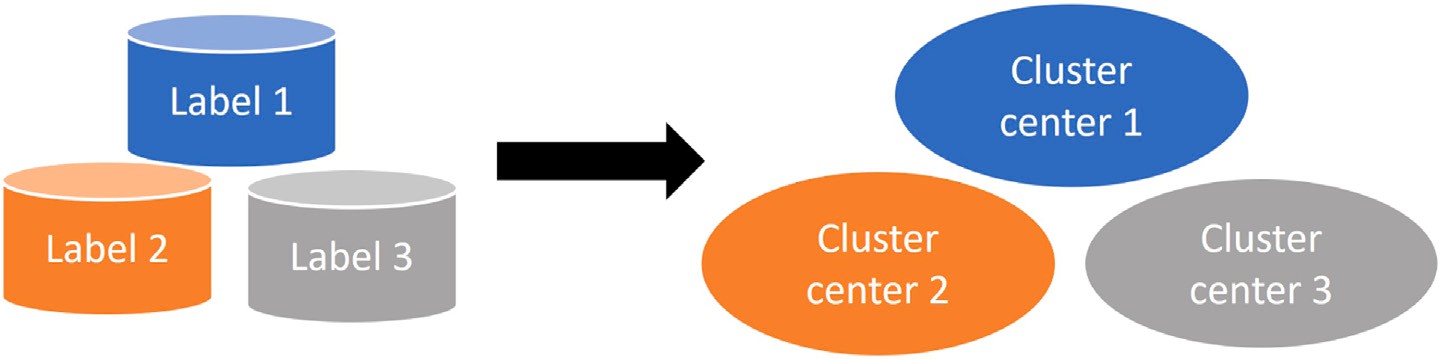
= *𝐼* ∧ *𝑤𝑗*

*𝛼* + *𝑤𝑗*

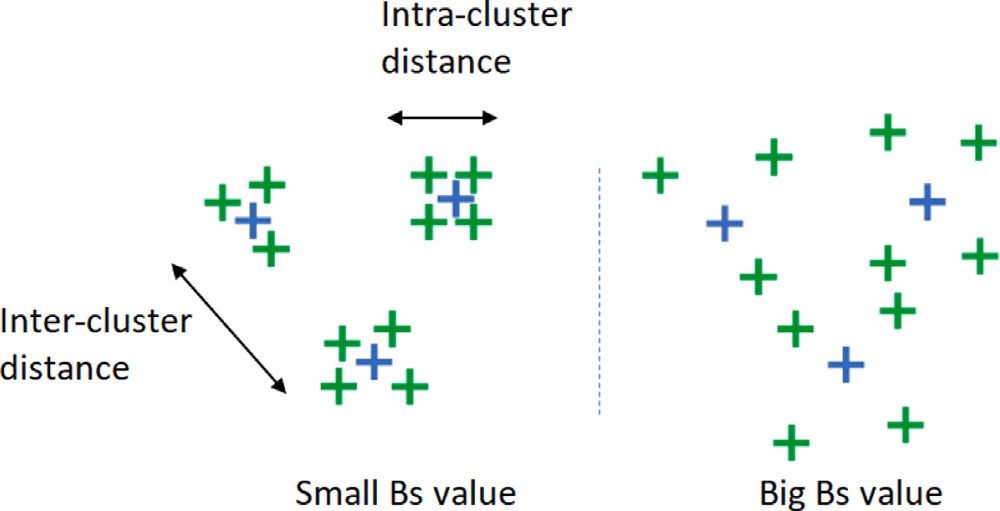
| |

(1)

and initialization instances are clustered with the rest of data regardless of their initial labels.



**/ig. 2.** SSFuzzyART initialization.

values are also possible without losing generality. We designate by *𝑏𝑖𝑡* of the binary modalities considered are either 1 or 0, but other pairs of

a modality that has exactly two possible values: 1 or 0.

**1** Input

* + - *𝐶*: Number of clusters
    - *𝑁𝑐*: *𝑁𝑐* = (*𝑛*1 *, 𝑛*2 *,* … *, 𝑛𝑚* ) list of clusters sizes
    - *𝐷*: Number of features, i.e. generated vector size
    - *𝐵𝑠*: Bit swapping rate of generated vectors
    - *𝑂𝑛𝑒*\_*𝑝𝑟*: Probability of having one value generated in the prototype

vector. The default value of this parameter is 0.5

**2 for** *cluster 𝑖 in 𝐶* **do**

**/ig. 3.** An explanatory example of intra and inter-cluster distances and the impact of

*𝐵𝑠* on the generated clusters.

**Table 1**

Category weights computation example through ap- plying the Fuzzy min operator on four instances of a

given label, the self-learning rate *𝛽* = 1 in this example.

**3**

**4**

**5**

**6**

**7 end**

Randomly generate a prototype of dimension *𝐷* whose values are

taken from a Bernoulli distribution of expectation *𝑂𝑛𝑒*\_*𝑝𝑟*

*Ber(𝑂𝑛𝑒\_𝑝𝑟)*.

**for** *vector 𝑗 in 𝑁𝑐𝑖* **do**

Copy the generated prototype and swap the state of *𝐵𝑠*%

randomly chosen bits

**end**

**Algorithm 1:** Clustering binary data generation algorithm with controlled variance

|  |  |
| --- | --- |
| Instance 1 | 000111 |
| Instance 2 | 101011 |
| Instance 3 | 010111 |
| Instance 4 | 100011 |
| **Category weights** | **000011** |

gory neurons (layer *𝐿*2). Next, initialization instances are clus- 3. In the third scenario, the labeled data is used to initialize cate-

tered with the rest of the data, but their labels are set and cannot change during the clustering process, thus guiding the clustering process.

The choice of the scenario depends on the data to be clustered and the problem to be solved. For example, if there is some doubt about the labeled instances, it could be useful to apply the second scenario and to keep SSFuzzyART cluster these instances again. In cases where the quality of labeled instances is high, then it is recommended to consider the first or the third scenario.

In this paper, **the second scenario** is considered. Then, labels are no longer considered in the clustering process and the instances used for initialization are clustered with the rest of the data regardless of their initial labels. This choice allows studying the ability of SSFuzzyART to correctly put the labeled instances in their clusters.

# Clustered binary data generation with controlled partitioning

This section present and explain the proposed data generation algo- rithm for binary clustered data. Besides, we present an empirical study of the generated clusters’ separability.

* 1. *Data generation algorithm*

This section introduces the data generation algorithm and the pa- rameters that allow controlling the characteristics of generated clusters. Algorithm [1](#_bookmark11) details the proposed data generation algorithm. The values

(*𝐶*), the list of the number of vectors per cluster (*𝑁𝑐* = (*𝑛*1*, 𝑛*2*,* … *, 𝑛𝑚*)), The input parameters of the algorithm are: the number of clusters the dimension of the binary vectors *𝐷* and the bit swapping rate ‘‘*𝐵𝑠*’’.

The bit-swapping rate controls the proximity of the vectors in their clusters (intra-cluster distance), while the average distance between clusters (inter-cluster distance) is defined by the prototypes. In this

algorithm, *𝐵𝑠* is the same for all clusters, as shown in [Fig.](#_bookmark9) [3](#_bookmark9). This means

The percentage of one value *𝑂𝑛𝑒*\_*𝑝𝑟* is an optional parameter that is that the intra-cluster distance is on average the same for each cluster.

set to 0.5 by default. This parameter could be specified to manage the percentage of one value in the generated prototype vector.

The data generation algorithm is detailed in the algorithm [1](#_bookmark11). Thus,

*𝐶* random vectors are randomly generated and are used as ‘‘prototypes’’ or ‘‘centers’’ of future clusters. For each cluster *𝑖*, *𝑁𝑐𝑖* vectors are

of an arbitrarily defined *𝐵𝑠*% fraction of randomly chosen bits. For generated by duplicating the prototype vector, then swapping the value

a *𝐵𝑠* = 20%, the following two vectors can be generated by randomly example, given the following 10-bit reference vector 1001101101 and

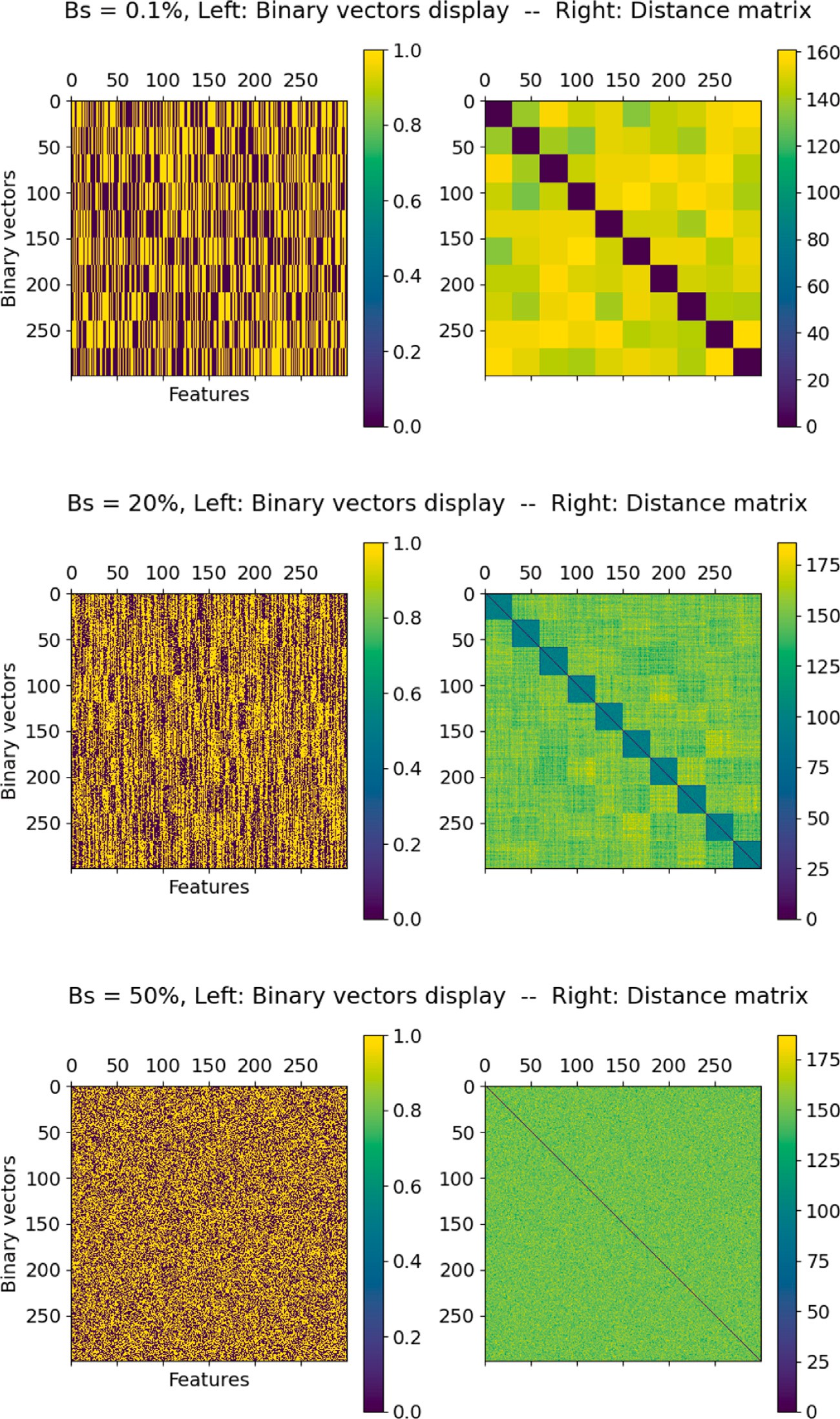
swapping two bits of the initial vector: 10011**1**1**0**01 and 10**1**1101**0**01.

* 1. *Evaluation of the generated clusters separation*

The binary data generation algorithm introduced in the [4.1](#_bookmark13) section is evaluated in this section. Indeed, in order to use these artificially generated data with confidence, it is important to control the quality of their generation. This quality control is defined through the control of the distance between the vectors in the same cluster (intra-cluster distance), and the distance between clusters (inter-cluster distance).

The set of generated vectors is called *𝐸*. This set is partitioned into

*𝐶* clusters, the *𝑖*th of which is named *𝐶𝑖*.



**/ig. 4.** Display of the generated binary vectors and their corresponding distance matrices for three values of *𝐵𝑠*: 0.1%, 20% and 50% respectively. Number of clusters *𝐶* = 10, Dimension *𝐷* = 300 and *𝑁𝑐* = 30. The Manhattan distance is used.

The binary data generation algorithm is, also, compared to a com- monly used solution in the literature that is considered in this paper as a baseline solution. This solution consist of generating real-valued ran- dom clustered data and then binarizing the generated clusters against a threshold. For this purpose, we used the available function make\_blobs[2](#_bookmark16)

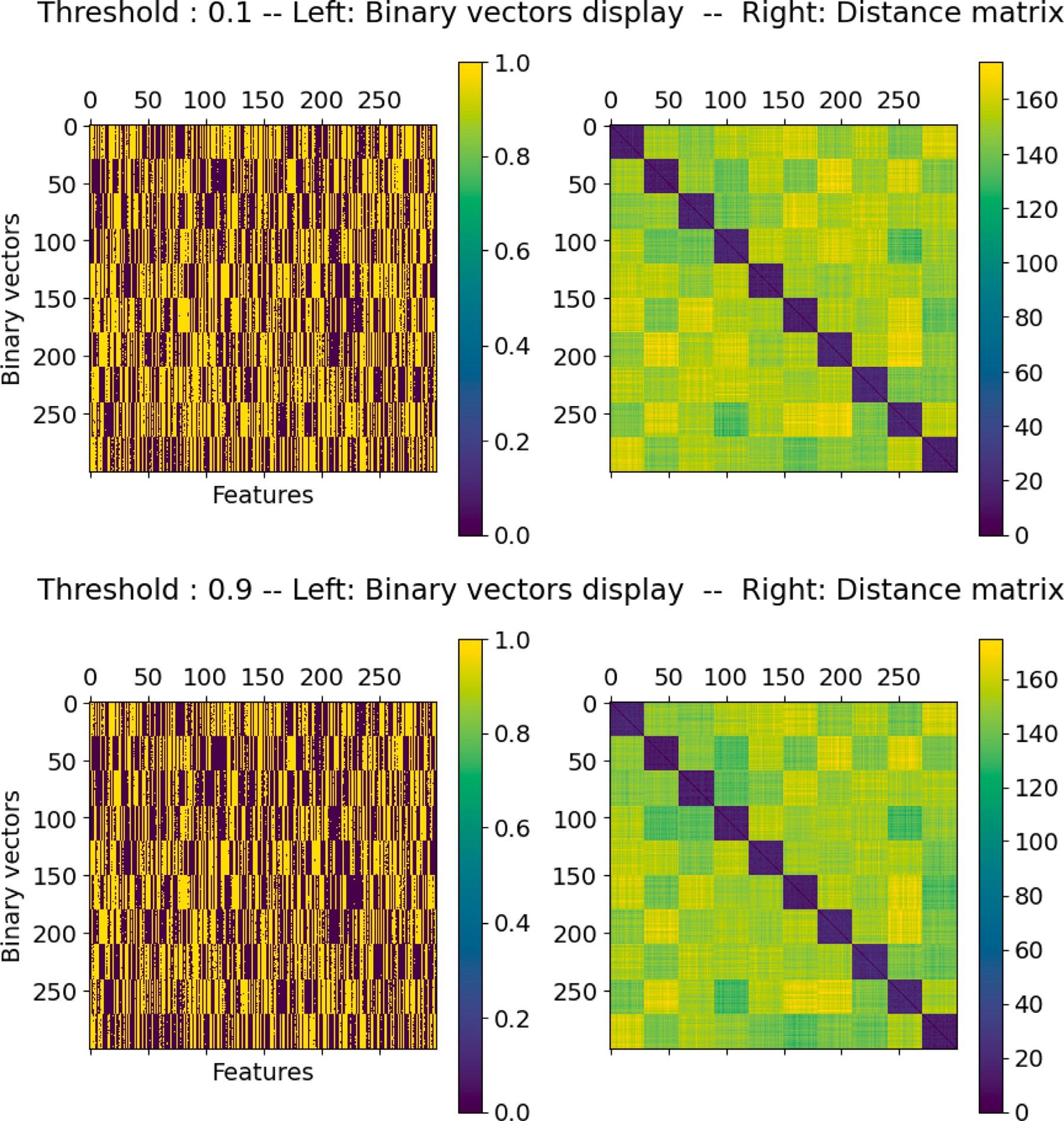
2 [make\_blobslink](https://shorturl.at/lMN12)

from the python library scikit learn. The generated samples follow the normal distribution. Then, the idea is to generate a set of real-valued random clusters. Next, a threshold is used to transform these real-values to binary values. Then, if a given real value is lower than the threshold,

the binary value will be 0 else it will be 1. This algorithm is called

Baseline data generation in the experiments.

We note that the algorithms and all the experiments are imple- mented with Python.



respectively. Number of clusters *𝐶* = 10, Dimension *𝐷* = 300 and *𝑁𝑐* = 30. The Manhattan distance is used. **/ig. 5.** Display of the generated binary vectors using the Baseline data generation algorithm and their corresponding distance matrices for two values of threshold: 0.1 and 0.9

distance matrices with three-bit swapping rates *𝐵𝑠* = 0*.*1%, *𝐵𝑠* = 20% [Fig.](#_bookmark15) [4](#_bookmark15) shows a visualization of three data sets and corresponding and, *𝐵𝑠* = 50% respectively. Besides, [Fig.](#_bookmark17) [5](#_bookmark17) shows a visualization of two

data sets generated with the Baseline data generation algorithm. In this experiment, we generated one real valued data set as explained above, and we used two different binarization threshold values, 0.1 and 0.9 respectively. The distance matrix consists of calculating the distance

between each pair of vectors (*𝑥, 𝑦*). The distance used in this paper is

the Manhattan distance:

*𝐷*

*𝑀𝑎𝑛*(*𝑥, 𝑦*) = *𝑥𝑖* − *𝑦𝑖* (5)

∑

*𝑖*=1

The vectors in [Fig.](#_bookmark15) [4](#_bookmark15) have a dimension *𝐷* = 300. Besides, 10 clusters

low (*𝐵𝑠* = 0*.*1%), the clusters are easily visible and separable. In coher- containing 30 vectors each are generated. When the bit-swapping rate is

ence, the distance matrix shows that the data vectors belonging to the same clusters are close to each other. If the bit swapping rate increases

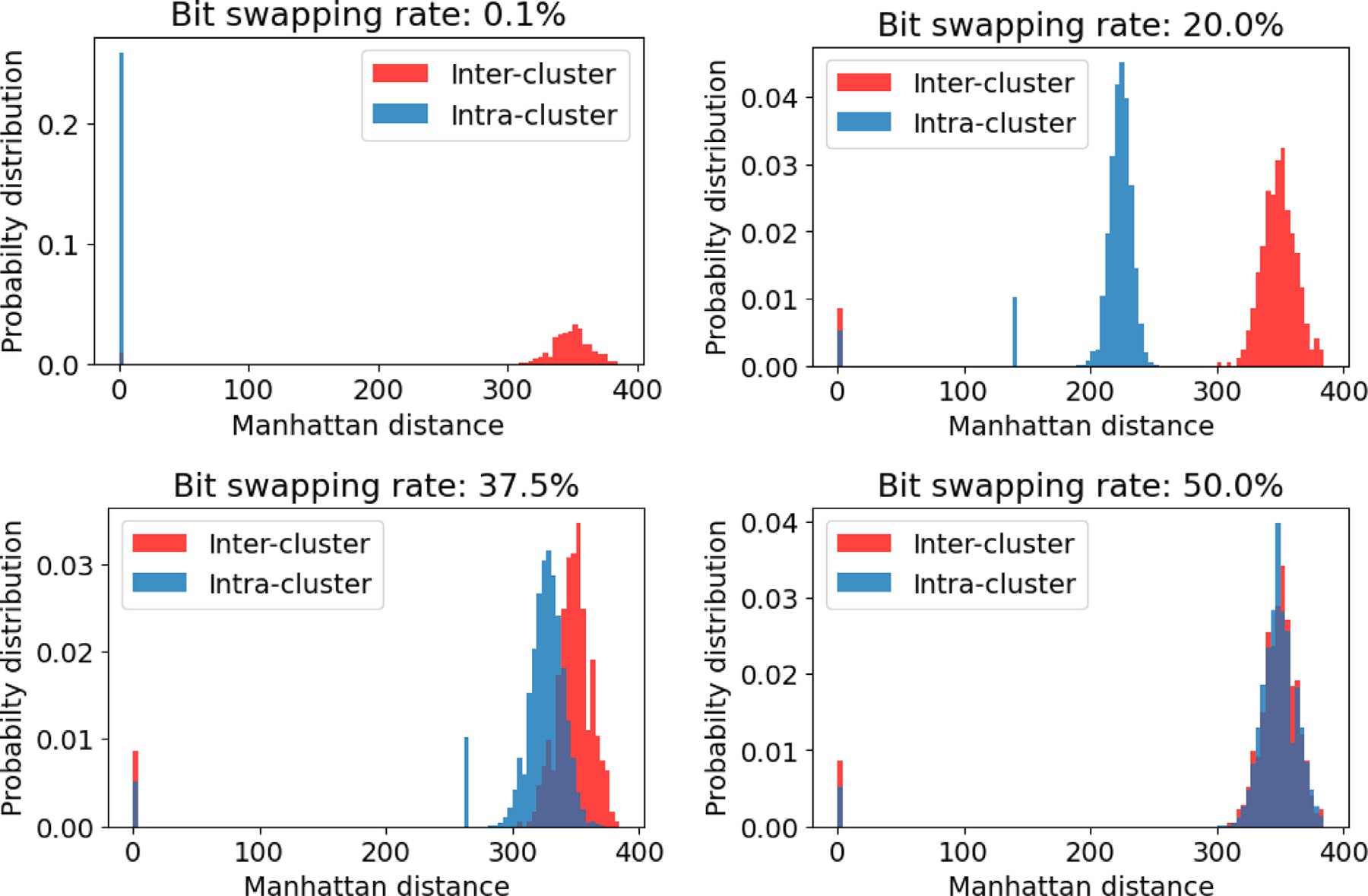
to *𝐵𝑠* = 20% the clusters remain visible but the average intra-cluster

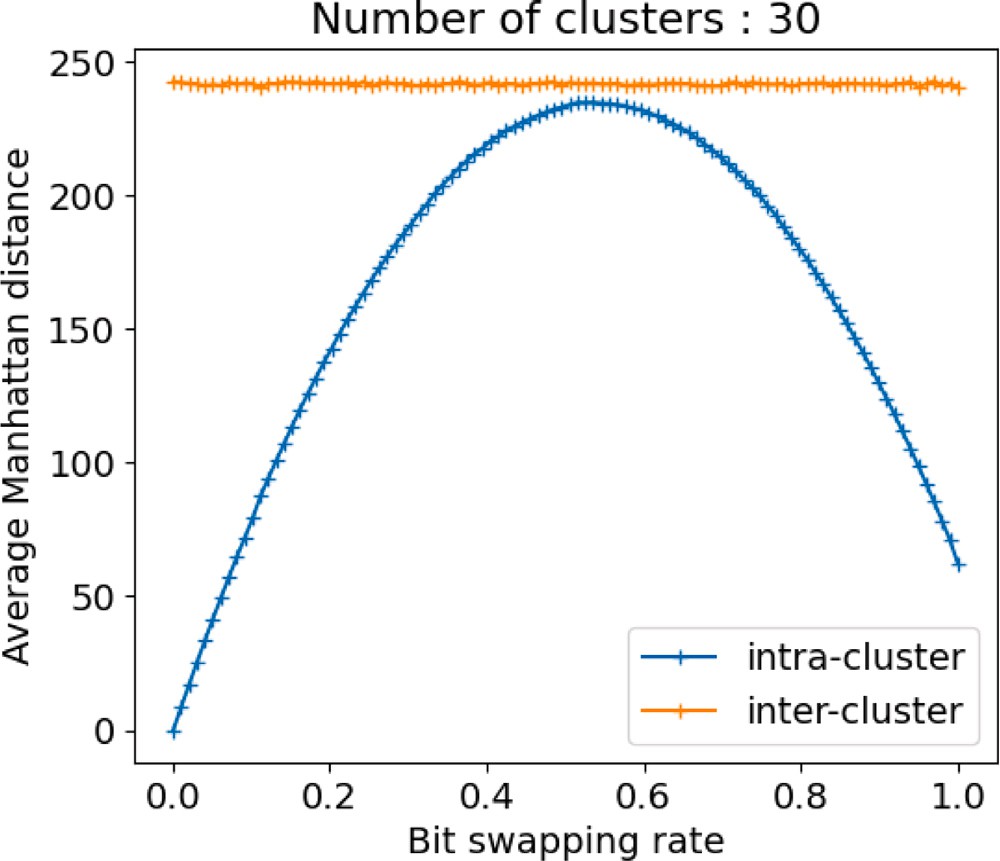
each other. With *𝐵𝑠* of 50%, there are no more visible clusters in the distance increases, leading to significantly more different vectors from

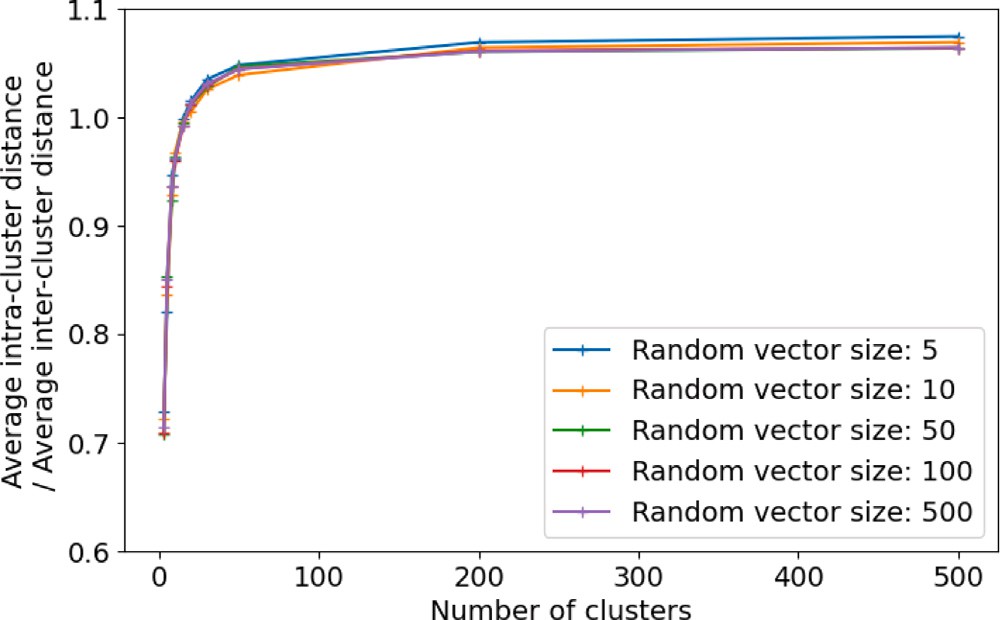
matrices. All the vectors are random, the average intra-cluster distance is equal to the average inter-cluster distance: the notion of the cluster has no longer any meaning in this case. The same configuration was considered in [Fig.](#_bookmark17) [5](#_bookmark17), then the generated vectors have a dimension

*𝐷* = 300. Besides, 10 clusters containing 30 vectors each are generated.

and two threshold values are considered, 0.1 and 0.9. According to this figure, the clusters are easily visible and separable and the distance matrices show that the data vectors belonging to the same clusters are close to each other. However, we have the same visualization for the two threshold values. Consequently, the binarization threshold has no effect on the clusters. The Baseline data generation algorithm does not offer the possibility to manage the clusters properties, as we need to study the clustering algorithm characteristics.



**/ig. 6.** Intra- and inter-cluster distance distributions Evolution in the function of bit swapping rate. *𝐶* = 30, *𝐷* = 700, and *𝑁𝑐* = 50. The distance Manhattan is used.



of the intra-cluster distance (several values of *𝐵𝑠* were used to determine the maximum **/ig. 8.** Ratio between the average value of the inter-cluster distance and the maximum and the average), according to the number of clusters *𝐶* and the dimension of the vector

*𝐷*. The Manhattan distance is used.

of bit swapping. Number of clusters *𝐶* = 30, number of vectors per cluster *𝑁𝑐* = 15 **/ig. 7.** Evolution of the intra and inter-cluster average distance according to the rate

and dimension of the binary vector = 500. The Manhattan distance used.

The experiment illustrated by [Fig.](#_bookmark18) [6](#_bookmark18) confirms these observations. This experiment consists in plotting the histograms of the intra- and inter-cluster distances as a function of the average bit-swapping rate.

to the same cluster *𝐶𝑖*, defined as follows: An intra-cluster distance is a distance between two vectors belonging

*𝐷𝑖𝑛𝑡𝑟𝑎*(*𝑥, 𝑦*) = *𝑀𝑎𝑛*(*𝑥, 𝑦*) | ∀*𝐶𝑖* ∈ *𝐶, 𝑥, 𝑦* ∈ *𝐶𝑖* (6)

different clusters. Let *𝐶𝑖* and *𝐶𝑗* be two different clusters, this distance Conversely, inter-cluster distances relate to vectors belonging to

is expressed as follows:

*𝐷𝑖𝑛𝑡𝑒𝑟*(*𝑥, 𝑦*) = *𝑀𝑎𝑛*(*𝑥, 𝑦*) ∣ ∀*𝐶𝑖, 𝐶𝑗* ∈ *𝐶, 𝐶𝑖* ≠ *𝐶𝑗 , 𝑥* ∈ *𝐶𝑖 𝑒𝑡 𝑦* ∈ *𝐶𝑗* (7)

Indeed, with a low *𝐵𝑠*, the histogram of intra-cluster distances

other. As the rate of bit swapping *𝐵𝑠* increases, the histogram of the is located at 0, because the vectors are extremely similar to each

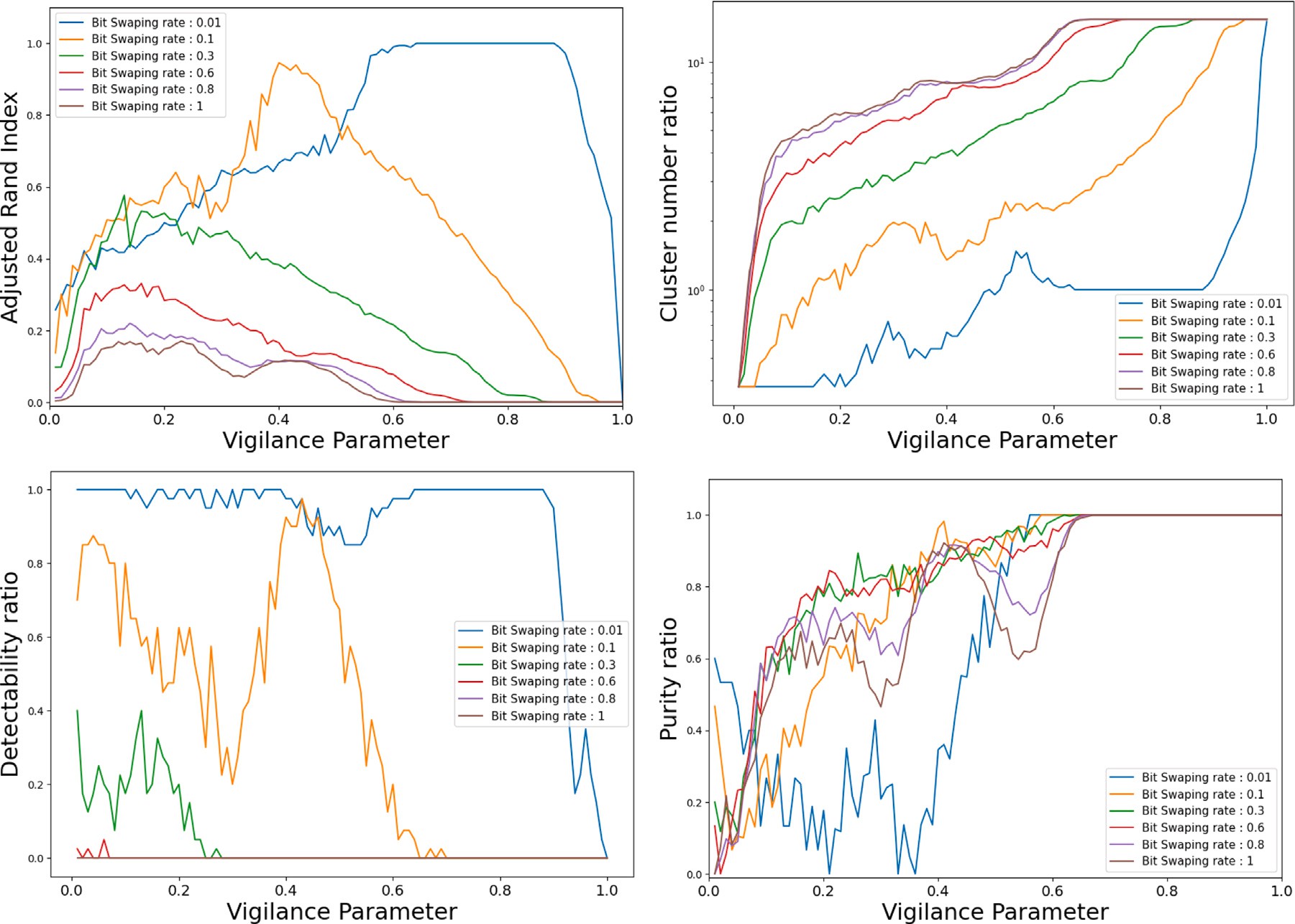
intra-cluster distances widens and approaches the histogram of the inter-cluster distances. Indeed, the clusters grow because the number

of swapped bits increases. For a high bit swapping rate (*𝐵𝑠* = 50%),

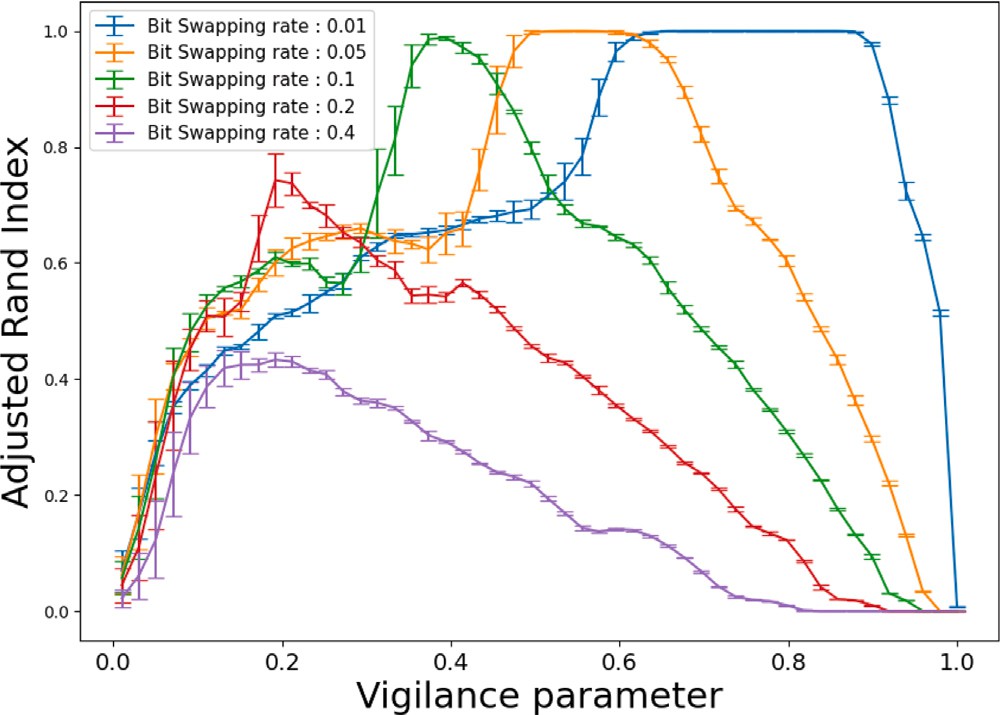
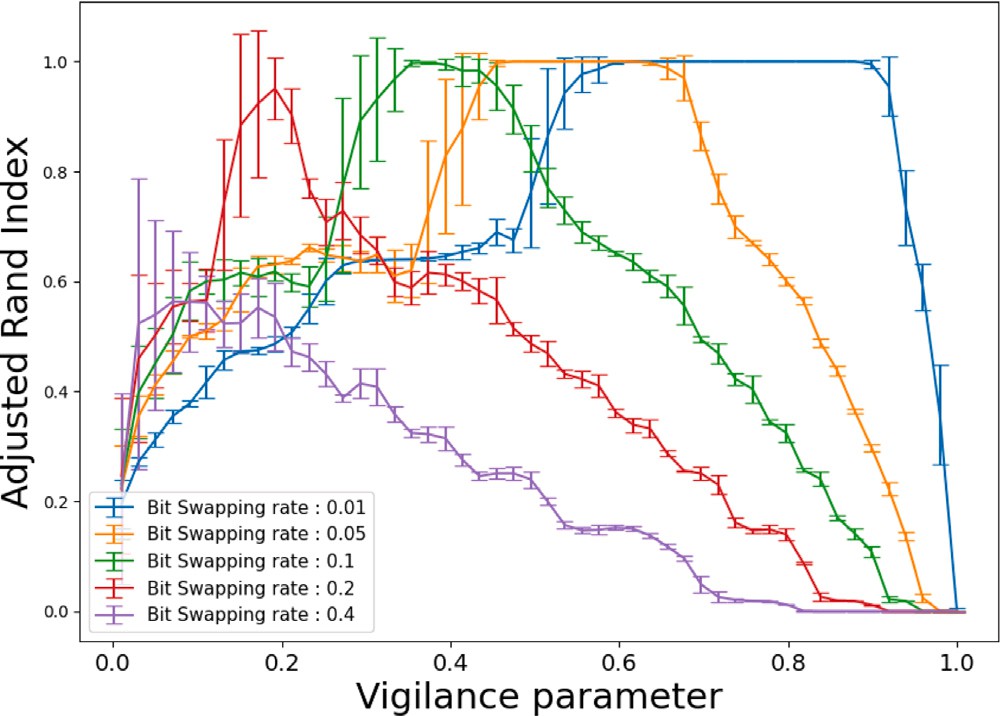
the two histograms are merged: there are no more clusters, only a uniform cloud of points. Moreover, we note that the histogram of the

inter-cluster distances does not depend on the *𝐵𝑠* rate. In fact, these

distances are fixed during the generation of the initial prototypes and,



**/ig. 9.** SSFuzzyART clustering quality (compared to ground truth) versus the vigilance parameter for several bit-swapping rates.



**/ig. 10.** Adjusted Rand Index versus the vigilance parameter for several values of bit swapping rate and binary vector feature number.

the rate *𝐵𝑠*. in coherence with the definition of the algorithm, are independent of

To better understand the impact of *𝐵𝑠* on intra- and inter-cluster dis-

tances, we introduce the notions of average distances by the following

definitions:

**/ig. 11.** Adjusted Rand Index versus the vigilance parameter for several values of bit swapping and cluster number. The error bar shows the variability associated with the cluster number.

The [Fig.](#_bookmark20) [7](#_bookmark20) shows the impact of the bit swapping rate *𝐵𝑠* on the intra and inter-cluster average distances. In coherence with [Fig.](#_bookmark18) [6](#_bookmark18), the inter-

cluster average distance is independent of the bit swapping rate *𝐵𝑠*. Moreover, the intra-cluster average distance increases until *𝐵𝑠* = 50%

*𝐷𝑖𝑛𝑡𝑟𝑎*

= 1

*𝑁𝑐𝑜𝑢𝑝𝑙𝑒𝑠 𝑖𝑛𝑡𝑟𝑎*

∑ *𝐷𝑖𝑛𝑡𝑟𝑎*

*𝑥,𝑦*∈*𝐸*

(*𝑥, 𝑦*) (8)

with the value of the inter-cluster average distance. For low *𝐵𝑠*, the then decreases beyond. Moreover, its maximum is almost confused

intra-cluster distance is much lower than the inter-cluster distance: the

1

*𝐷* =

*𝑖𝑛𝑡𝑒𝑟*

∑ *𝐷𝑖𝑛𝑡𝑒𝑟*(*𝑥, 𝑦*) (9)

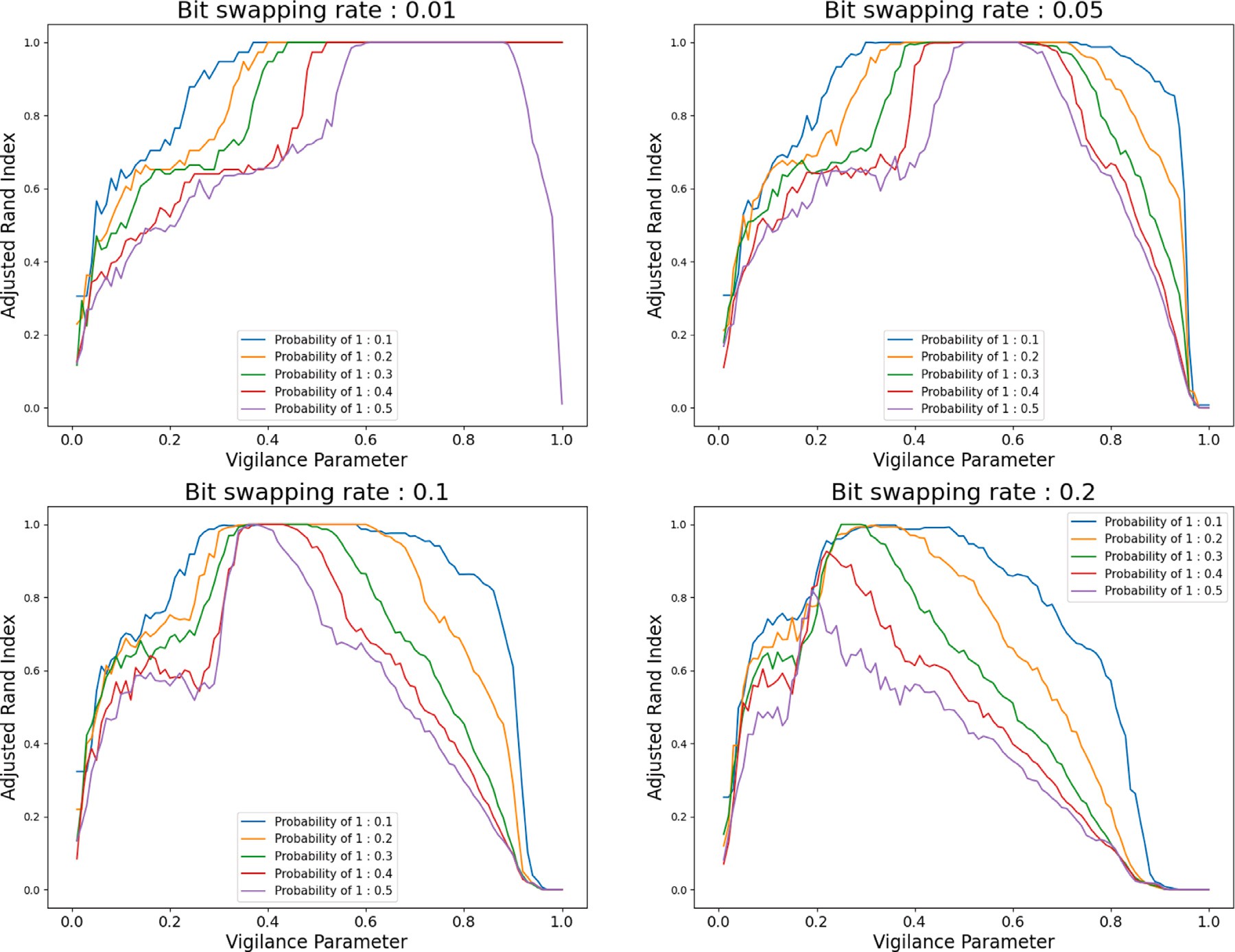
clusters are very compact and far from each other. As the bit swapping

*𝑁𝑐𝑜𝑢𝑝𝑙𝑒𝑠 𝑖𝑛𝑡𝑒𝑟 𝑥,𝑦*∈*𝐸*

These average distances correspond to the arithmetic mean of the histograms presented in [Fig.](#_bookmark18) [6](#_bookmark18).

When *𝐵𝑠* = 50%, the intra-cluster distance reaches its maximum, the rate increases, the clusters grow at a constant inter-cluster distance.

radius of the clusters is close to the inter-cluster distance, so there are



**/ig. 12.** Adjusted Rand Index for several bit swapping rates, for several occurrence probabilities of 1 in binary vectors.

no more clusters defined. The value of the maximum at *𝐵𝑠* = 50% is

used for the generation of the prototypes. As *𝐵𝑠* continues to increase, linked to the arbitrary choice of the expectation of Bernoulli’s law at 0.5

the inter-cluster distance decreases because the bit-swap potential has been exceeded, and any further bit change leads to the complementary

vector of the initial vector. In the case where *𝐵𝑠* = 100%, all the vectors

are equal to the complement of the prototype, except the prototype

Next, [Fig.](#_bookmark19) [8](#_bookmark19) illustrates the impact of the number of clusters *𝐶* and itself, hence the fact that the intra-cluster distance is not strictly zero. the dimension of the vectors *𝐷* on the intra- and inter-cluster distances.

This figure presents the ratio between the average value of the inter- cluster distance and the maximum of the average intra-cluster distance, defined as follows:

# Experiments

Several experiments are made to evaluate the proposed SSFuzz- yART. All experiments presented in this paper are implemented with Python3. Then a set of evaluation measures is introduced in Section [5.1](#_bookmark26). A first set of experiments evaluates the characteristics of SSFuzzyART (Section [5.2](#_bookmark31). Finally, an evaluation of SSFuzzyART and a compari- son with FuzzyART using three available benchmarks is presented in Section [5.3](#_bookmark34).

* 1. *Evaluation measures*

Four evaluation measures are considered to evaluate the proposed

*𝑅* = *𝐷𝑖𝑛𝑡𝑒𝑟*

*𝑚𝑎𝑥𝑥,𝑦*∈*𝐸 𝐷𝑖𝑛𝑡𝑟𝑎*(*𝑥, 𝑦*)

(10)

SSFuzzyART algorithm, and to compare it to the classic Fuzzy ART. The considered measures are defined as follows:

depending on the number of clusters *𝐶* and the dimension of the vector,

*𝐷*.

inter and intra-cluster distances. On the other hand, *𝑅* depends on the This figure shows that the vector dimension has no impact on the number of clusters only if this number is less than 100. When the *𝑅*

indicator is less than 1, the clusters disappear for the bit swapping rates

*𝐵𝑠* less than 50%. Controlling the clusters concentration is therefore difficult in this case. In the case where the *𝑅* indicator is greater than 1, the clusters disappear from a *𝐵𝑠* equal to 50%. Therefore, the clusters

tering in the range [−1*,* 1]. It considers all pairs of samples in • *Adjusted Rand index*[3](#_bookmark27): it is a similarity score between two clus-

same or different clusters. When this score is equal to 1, then the the prediction and target clustering and checks if pairs are in the

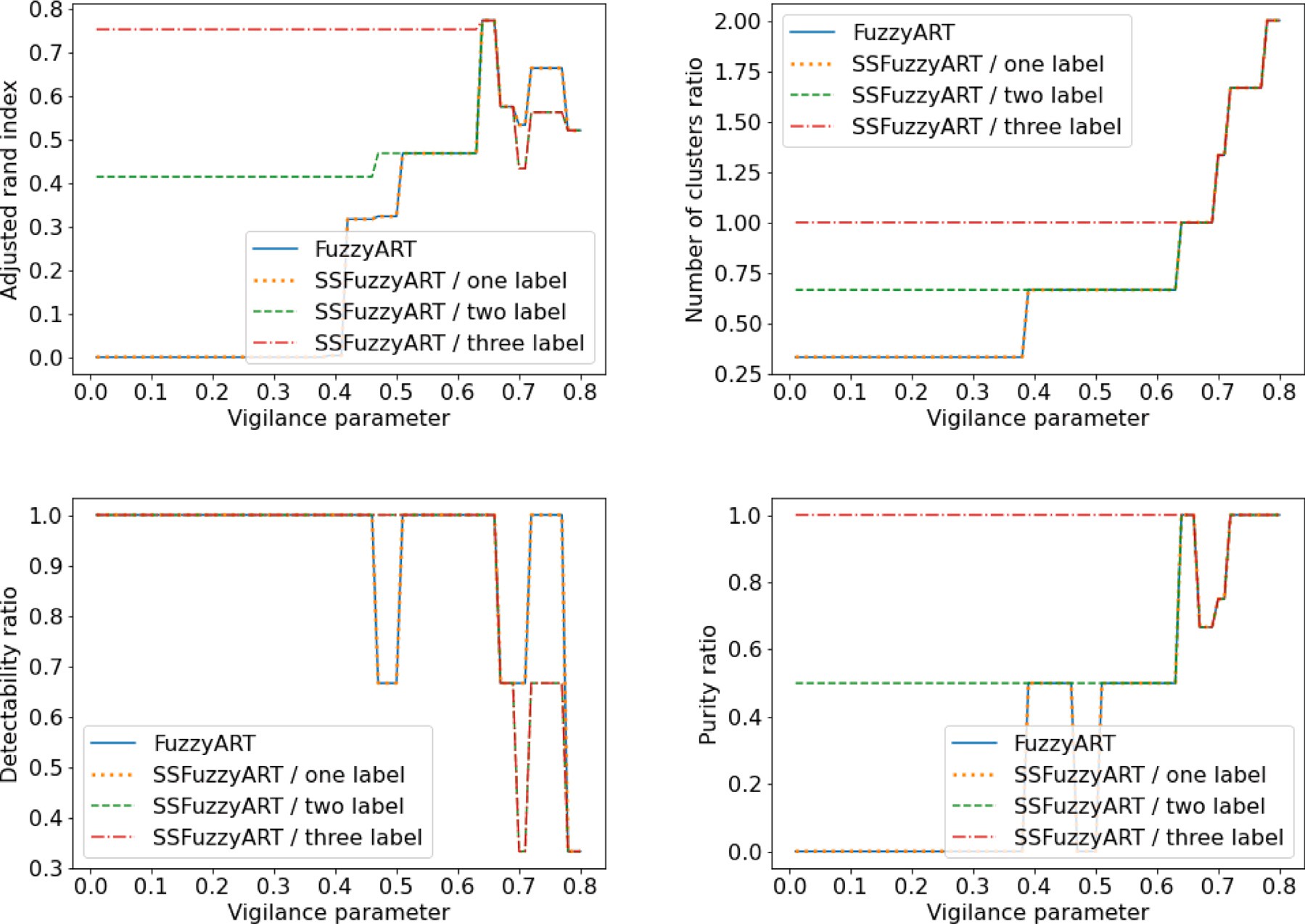
clustering is perfect. When it is near zero, then the clustering is near the random clustering.

the target clusters number. If this ratio is equal to 1, then the • *Number of cluster ratio*: the ratio of predicted clusters number by

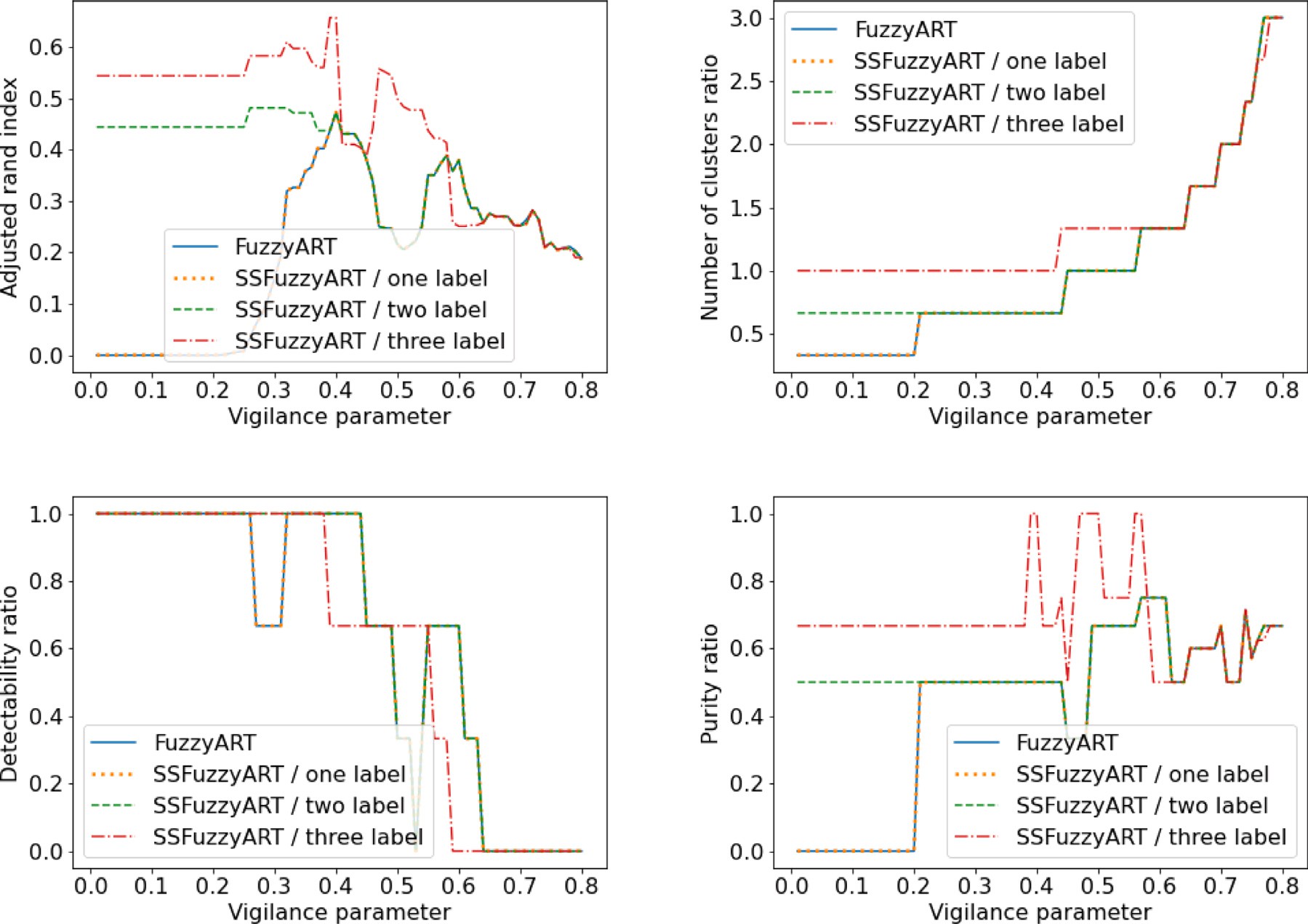
can be controlled independently of the *𝐷* dimension of the vectors.

Moreover, the more the number of generated clusters increases, the easier it is to control the concentration of the vectors.

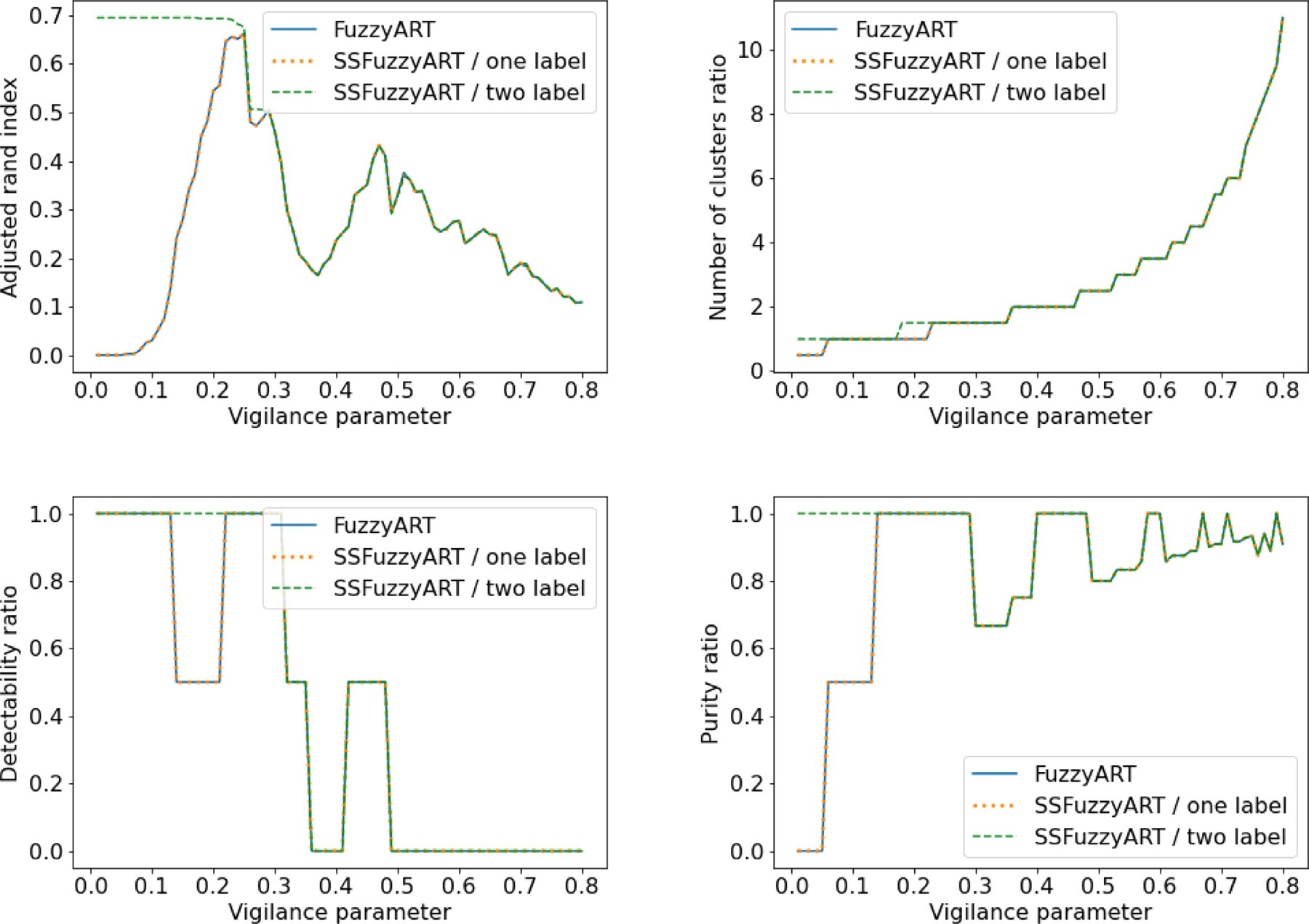
3 [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html) [adjusted\_rand\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html)



**/ig. 13.** SSFuzzyART VS Fuzzy ART on Iris dataset. In this experiment, three initialization samples were considered. The first one contains 5 instances having the same label. The second one contains 10 instances having two labels (five instances per label). The third one contains 15 instances having three labels (five instances per label).



**/ig. 14.** SSFuzzyART VS Fuzzy ART on Wine dataset. In this experiment, three initialization samples were considered. The first one contains 5 instances having the same label. The second one contains 10 instances having two labels (five instances per label). The third one contains 15 instances having three labels (five instances per label).



**/ig. 15.** SSFuzzyART VS Fuzzy ART on Breast cancer dataset. In this experiment, two initialization samples were considered. The first one contains 30 instances having the same label. The second one contains 60 instances having two labels (30 instances per label).

predicted clustering has exactly the same number of clusters as the target clustering. This measure is useful to evaluate Fuzzy ART and SSFuzzyART, as both do not use the number of clusters as input.

* *Cluster purity ratio*: the ratio of pure clusters, where a cluster is considered pure if at least 70% of its elements have the same label.
* *Label detectability ratio*: the ratio of detectable labels, where a label is considered detectable if at least 70% of data instances having that label are in the same cluster.

The considered cluster purity ratio and label detectability ratio have some tolerance on the definition of pure cluster and detectable label, respectively. This tolerance is defined by the fact that a cluster is pure if at least 70% of its elements have the same label, and a label is detectable if at least 70% of data instances having that label are in the same cluster. This tolerance allows these two measures to accept a clustering error of 30%. The choice of the tolerance value is defined according to the clustering problem, then for some problems it could be higher and for other problems it could be lower than 70%. In this paper, the clustering tolerance is fixed to 70%. A good prediction should have high cluster purity ratio and high label detectability ratio. This tolerance is useful to find the best compromise between these two measures. These two measures are influenced by the number of clusters. Consequently, when the number of clusters is high, these clusters are usually pure. However, in that case, the detectability of labels is low. With a high number of clusters, a given label instances could be scattered in several clusters.

impact of the layer *𝐿*2 initialization on the final prediction of the The next section presents and discusses experiment results and the

SSFuzzyART.

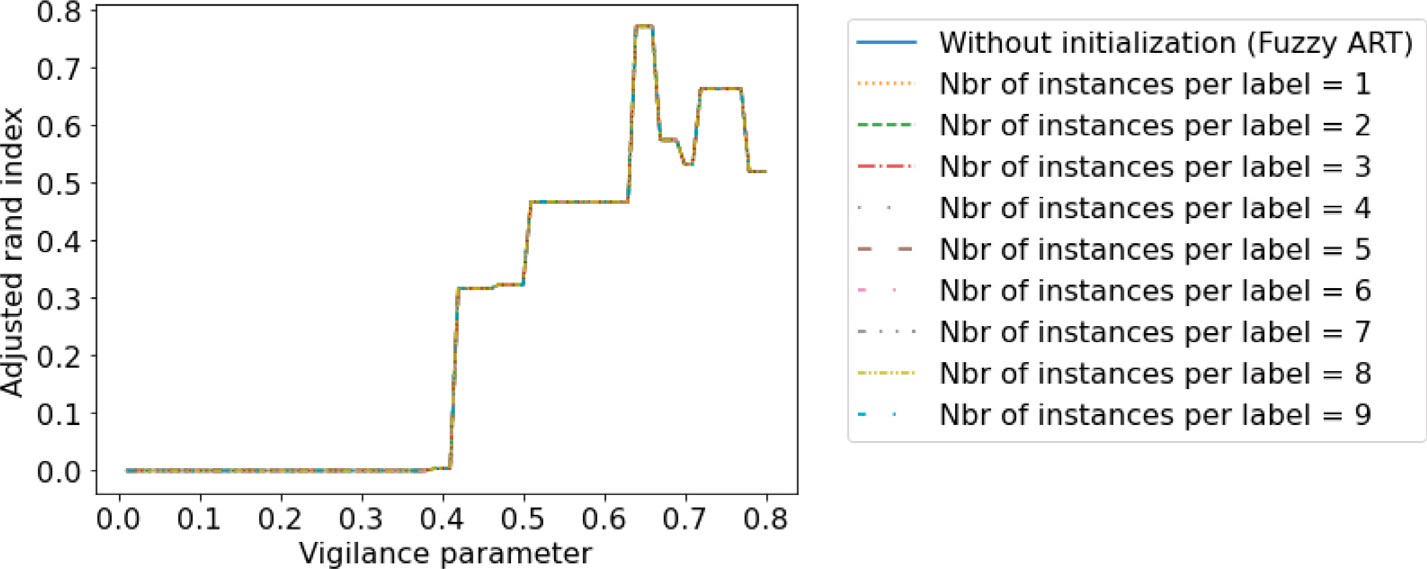
* 1. *SSFuzzyART characteristics study on generated data*

SSFuzzyART is tested on a simulated dataset generated using the al- gorithm introduced in Section [4.2](#_bookmark14). It consists of binary vectors grouped in clusters whose variance can be controlled with the bit-swapping parameter. This parameter ranges from 0 to 1 and the cluster variance is the highest when the bit-swapping rate is equal to 0.5 (in most ‘‘normal’’ cases). This means that clusters are created with an assigned root cause.

As the binary vector dataset is simulated, the cluster in which each point belong is known. This allows for assessing the ability of SSFuzzyART to effectively cluster binary vectors. The quality of the SSFuzzyART clustering is assessed using the Adjusted Rand Index, which compares the SSFuzzyART clustering with the known vector assignment from the generation process.

The Adjusted Rand Index, the number of cluster ratio, the de- tectability ratio, and the purity ratio are computed for several values of bit swapping rates and vigilance parameters. The maximum number of epochs for SSFuzzyART is set at 20. The number of created clusters is 40 with 15 vectors each and the vectors have 200 features. Besides, 75 initialization vectors having 15 labels (5 vectors per label) are con- sidered in the initialization set of SSFuzzyART. Results are presented in [Fig.](#_bookmark21) [9](#_bookmark21).

According to the adjusted rand index curves in [Fig.](#_bookmark21) [9](#_bookmark21), for low values of bit swapping (meaning clusters are well separated), SSFuzzyART can perfectly cluster the binary vectors for a large range of vigilance param- eters (from 0.6 to 0.9). If the vigilance parameter is too small, then the created clusters are too coarse and contain several individuals/vectors having different types. If the vigilance parameter is too close to 1, then the clusters are too granular and individuals/vectors having the same type are spread on multiple clusters. If the bit swapping rate increases,



**/ig. 16.** Impact of the number of considered instances per label in the initialization sample of the SSFuzzyART compared to Fuzzy ART (without initialization). Iris dataset is used, only one label is considered in this experiment.

meaning that clusters are bigger and closer to one another, SSFuzzyART is less able to accurately cluster the data. Moreover, the maximum of the Rand index moves to the left: as the generated cluster is bigger and closer, some begin to merge and the vigilance parameter allows for the maximum Rand index is lower to allow coarse clusters to be identified. According to the cluster number ratio curves in [Fig.](#_bookmark21) [9](#_bookmark21), whatever the variance of clusters (the bit swapping rate), the number of clusters created is globally increasing with the vigilance parameter. This is coherent with the internal working of SSFuzzyART. Moreover, the vigilance parameter values allow the creation of the exact same number of clusters than the ground truth decreases when the bit-swapping rate increases. This means that for coarser cluster structures, using a lower vigilance parameter allows for better retrieval of the cluster structure. According to the detectability ratio curves in [Fig.](#_bookmark21) [9](#_bookmark21), the label detectability decreases when the vigilance parameter increases. The more clusters are created, the more the labels are spread in numerous

clusters, therefore reducing their detectability.

According to the purity ratio curves in [Fig.](#_bookmark21) [9](#_bookmark21), cluster purity is high when the vigilance parameter is high. This makes sense because when a lot of clusters are created, they contain only a few individuals that are likely to have the same root cause.

To summarize, provided clusters are well-defined and separated, SSFuzzyART can retrieve the clustering structure of binary vectors

In the second experiment, the effect of the number of features (the binary vector length) is studied by computing the Adjusted Rand Index for several values of the feature number (200, 500, 1000, 2000) and different values of bit swapping. The generated dataset contains 40 clusters with 15 vectors each, and the vectors have 200 features. Besides, 75 initialization vectors having 15 labels (5 vectors per label) are considered in the initialization set of SSFuzzyART. Results are displayed in [Fig.](#_bookmark22) [10](#_bookmark22).

According to [Fig.](#_bookmark22) [10](#_bookmark22), the feature number has a relatively small impact on the Adjusted Rand Index, mainly for vigilance parameter values that are less than 0.5. Consequently, the SSFuzzyART clustering process can be achieved for different values of feature numbers. The feature number has no significant impact on clustering results.

The effect of the cluster number to detect (the number of root causes) is displayed in [Fig.](#_bookmark23) [11](#_bookmark23). The number of features of the binary vector is 200, the number of vectors per cluster is 15, and 75 initial- ization vectors having 15 labels (5 vectors per label) are considered in the initialization set of SSFuzzyART. According to [Fig.](#_bookmark23) [11](#_bookmark23), the number of clusters has little impact on the behavior of SSFuzzyART. Consequently, SSFuzzyART’s capacity to cluster data is not dependent on the number of clusters, but more on the relative distance between clusters (accounted by the bit-swapping parameter).

In [Fig.](#_bookmark24) [12](#_bookmark24), we investigate the effect of the occurrence probability

**Table 2**

Considered benchmark datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Benchmark | Number of categories | Number of instances | Number of features |
| Iris | 3 | 150 | 4 |
| Wine | 3 | 178 | 13 |
| Breast cancer | 2 | 569 | 30 |

the adjusted Rand index for several occurrence probabilities of 1 in random vectors, for several bit swapping rate. Having vectors biased toward 0 (meaning having less 1 values than 0 values), characterized by low values of occurrence probability of 1, allows to SSFuzzyART to find optimal partitioning of the vectors for a larger or different range of vigilance parameters.

* 1. *SSFuzzyart evaluation and comparison with FuzzyART*

To evaluate the proposed SSFuzzyART algorithm and compare its performance to the classic FuzzyART, a set of experiments is made to compare Fuzzy ART results to the SSFuzzyART results using three benchmark datasets. The considered datasets were selected from UCI machine learning repository.[4](#_bookmark35) [Table](#_bookmark33) [2](#_bookmark33) presents the considered datasets. To evaluate SSFuzzyART, a set of experiments on three benchmarks ([Table](#_bookmark33) [2](#_bookmark33)) is carried out. The clustering results of SSFuzzyART are com- pared to Fuzzy ART results according to the four considered evaluation measures. In each experiment, a sample from the dataset is selected to

initialize the *𝐿*2 layer in SSFuzzyART.

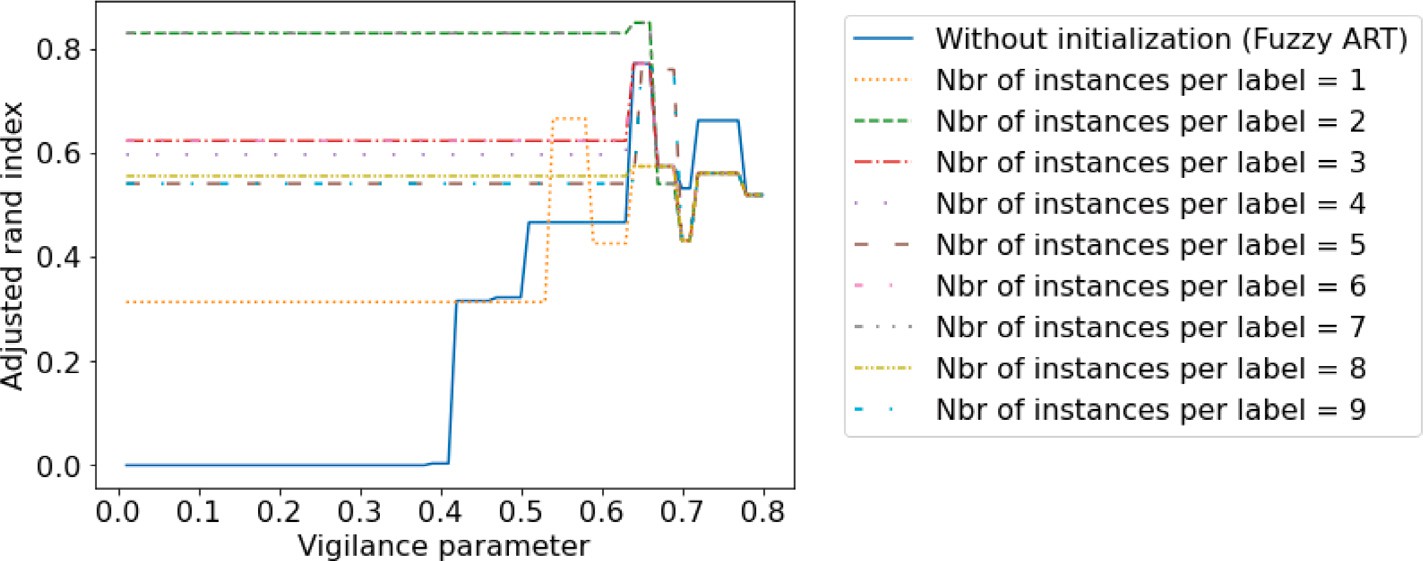
[Fig.](#_bookmark28) [13](#_bookmark28), [Fig.](#_bookmark29) [14](#_bookmark29) and [Fig.](#_bookmark30) [15](#_bookmark30) presents the experiment results on respectively Iris, Wine and Breast cancer datasets. In these experiments, the Adjusted Rand score, the number of clusters ratio, the purity ratio and the detectability ratio were measured in function of the vigilance

parameter *𝜌* to compare the SSFuzzyART to Fuzzy ART.

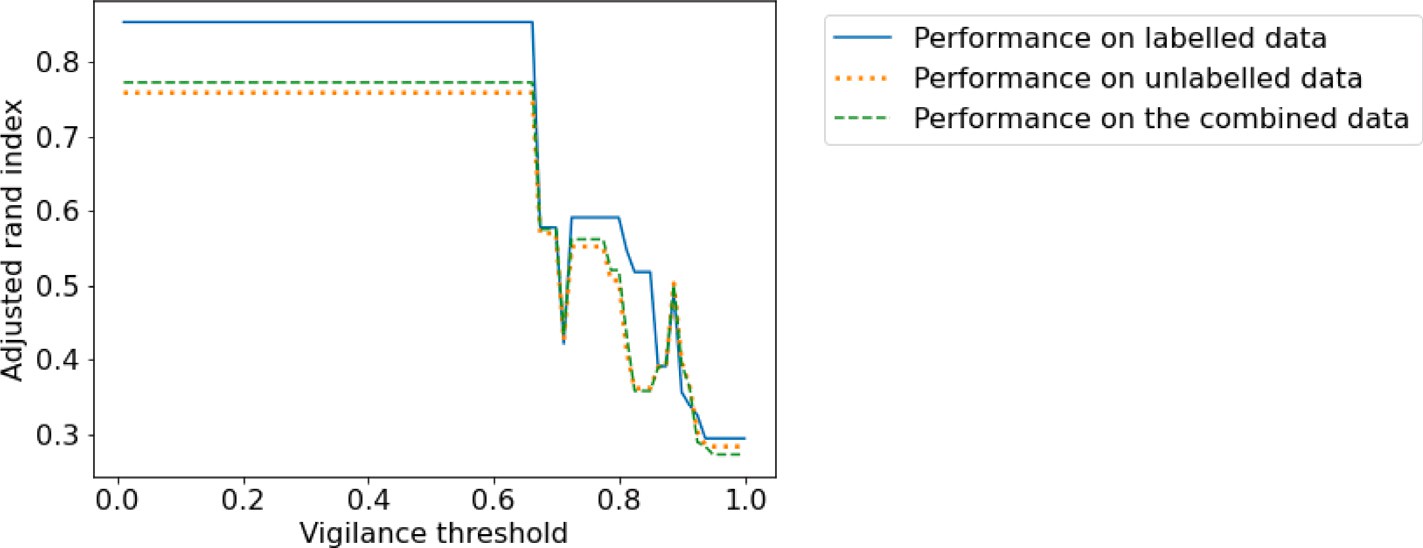
Moreover, to study the impact of the number of labels/category used in the initialization sample, a set of experiments was carried out using different samples containing a different number of labels, depending on the selected dataset. The number of selected instances depends also on the number of labels, to ensure that labels are equally present in the sample. For example, the Iris dataset contains three labels. The first sample out of the Iris Dataset contains five instances all having the same label, the second sample contains ten instances belonging to any of two different labels (five instances per label) and the third sample contains fifteen instances from the three available labels (five instances per label). In a given sample, all considered labels are represented with the same number of instances.

of 1 in binary vectors. In fact, real data are known to be sparse, with

more 0 values than 1 values in their feature vectors. [Fig.](#_bookmark24) [12](#_bookmark24) displays 4 <https://archive.ics.uci.edu/ml/datasets.php>



**/ig. 17.** Impact of the number of considered instances per label in the initialization sample of the SSFuzzyART compared to Fuzzy ART (without initialization). Iris dataset is used, and three labels are considered.



**/ig. 18.** SSFuzzyART Adjusted Rand index on the labeled data (seeds), the unlabeled data and the combined dataset. Iris dataset is used, and three labels are considered with 7 random instances per label.

According to the results in [Fig.](#_bookmark28) [13](#_bookmark28), [Fig.](#_bookmark29) [14](#_bookmark29) and [Fig.](#_bookmark30) [15](#_bookmark30), the SS- FuzzyART leads to better results than Fuzzy ART when the vigilance

parameter, *𝜌*, values are low, and they yield similar results when *𝜌*

values are higher. Indeed, when the *𝜌* parameter is high, the tolerance

of the algorithm to create a new cluster is higher, which increases the

clustering process is loosed. However, when the *𝜌* parameter is low, number of clusters, then the initialization that is made to guide the

the tolerance to create a new cluster is low too. Then, the number of cluster is also low, and the algorithm will not be allowed to add many clusters, especially when most possible labels are represented

with corresponding neurons in the *𝐿*2 layer.

Another important conclusion from these experiments is that the available number of labels in the initialization sample has an impact on the output prediction. When only one label is available in the initialization sample, the SSFuzzyART and Fuzzy ART have almost the same output results. However, when the number of available labels in the initialization sample increases, the quality of the clustering increases too. Besides, according to the results in [Fig.](#_bookmark28) [13](#_bookmark28) and [Fig.](#_bookmark29) [14](#_bookmark29), even with two available labels from three in the labeled data, the SSFuzzyART is able to improve the clustering performance compared to the classic Fuzzy ART results.

Furthermore, the classic Fuzzy ART algorithm is sensitive to data order and its results are impacted with it. Indeed, clusters are cre- ated gradually with the arriving of data instance, as explained in above. With the initialization step, the algorithm is provided with some information that helps to better detect the clusters. Consequently, SSFuzzyART improves the clustering performance thanks to this initial- ization.

A second experiment is made to study the impact of number of instances in the initialization sample if there is only one available label

of instances in these samples is in the range [1*,* 9] and all samples are in the data. Nine different samples were considered, and the number

selected from instances that belong to only one label. [Fig.](#_bookmark32) [16](#_bookmark32) presents the results of this experiment. [Fig.](#_bookmark32) [16](#_bookmark32) shows overlapping curves, i.e., all curves are equals. These results mean that a sample with only one label is not having any impact on the clustering results, even with many instances having that label. This experiment confirms the results of [Fig.](#_bookmark28) [13](#_bookmark28). In fact, having only one label in the initialization sample have no impact on the clustering results of SSFuzzyART. To have some improvement, the number of available labels must be greater than one.

A third experiment is carried out to study the impact of the number of instances used during initialization on the clustering results. [Fig.](#_bookmark36) [17](#_bookmark36) shows the Adjusted Rand index with different vigilance threshold val- ues and nine different samples. All these samples contain random instances from the three labels. The number of instances per label

is in the range [1*,* 9]. According to [Fig.](#_bookmark36) [17](#_bookmark36), increasing the number

of initialization instances always improves the Adjusted Rand index.

Indeed, even with one instance per label, SSFuzzyART has a Rand index that is greater than or equals to the Fuzzy ART Rand index. Furthermore, a number of instances per label that is greater than 2 has always a positive impact on the Rand index, according to these results.

In the previous experiments, the labeled instances that are used to initialize SSFuzzyART are clustered with the rest of the data.

Then, a last experiment is carried out to compare the performance of SSFuzzyART on the labeled data (labeled instances that are used as seeds), the unlabeled data (the rest of the dataset) and the combined dataset. Iris dataset is considered in this experiment. [Fig.](#_bookmark37) [18](#_bookmark37) presents the results of this experiment. According to the results in [Fig.](#_bookmark37) [18](#_bookmark37), the highest Adjusted Rand score is obtained on the labeled data. This is an expected result as the labeled data is used to initialize SSFuzzyART second layer neurons. The Adjusted Rand score on the unlabeled data is almost equal to the Adjusted Rand score of the entire dataset. These results confirm that SSFuzzyART has a positive impact on the performance of the clustering. Besides, the initialization of the layer

*𝐿*2 is useful to obtain a better clustering prediction.

# Conclusion

This paper introduces a SSFuzzyART algorithm. It is a Fuzzy ART semi supervised clustering algorithm that can use a set of available labeled data to improve the clustering prediction. SSFuzzyART uses

the labeled instances to initialize its categories in the *𝐿*2 layer. This

initialization is useful to guide the clustering process, and it influences

positively the clustering prediction. Besides, the classic Fuzzy ART algorithm is sensitive to the choice of the vigilance parameter that manage the tolerance of the algorithm to create a new cluster, this sensitivity has been inherited by SSFuzzyART. Then, it is generally

recommended choosing a relatively low *𝜌* to get good performance. The

*𝜌* value could depend on the number of categories that are present in the initialization data. Then, if all categories are present, a lower *𝜌* is

recommended.

A set of experiments was made and discussed to deeply study the SSFuzzyART characteristics and to compare it to the classic Fuzzy ART algorithm. According to these experiments, SSFuzzyART initialization

improved the clustering performance, especially with smaller *𝜌* values.

Besides, the number of clusters present in the initialization sample has

an important impact on the algorithm performance. Finally, the number of instances per label in the initialization sample has no impact on the clustering performance.

In the future work, it will be interesting to test SSFuzzyART in a real world problem and to study its impact on real data. A second perspective is to compare the three possible scenarios introduced in Section [3.2](#_bookmark7), and to study their impact on real world problems. A Third perspective of this work could be to find a solution to influence the Fuzzy ART clustering process with available labeled data.

# CRediT authorship contribution statement

**Siwar Jendoubi:** Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Investigation, Writing – review & editing. **Aurélien Baelde:** Software, Conceptualization, Validation, Investigation, Writing – review & editing. **Thong Tran:** Supervision.

# Declaration of competing interest

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

# Data availability

Data will be made available on request.

# References

1. [Carpenter GA, Grossberg S, Rosen DB. Fuzzy ART: Fast stable learning and](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb1) [categorization of analog patterns by an adaptive resonance system. Neural Netw](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb1) [1991;4(6):759–71.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb1)
2. Bingwen C, Wenwei W, Qianqing Q. Infrared target detection based on fuzzy ART neural network. In: 2010 Second international conference on computational intelligence and natural computing. IEEE; 2010, [http://dx.doi.org/10.1109/cinc.](http://dx.doi.org/10.1109/cinc.2010.5643745) [2010.5643745](http://dx.doi.org/10.1109/cinc.2010.5643745).
3. Ilhan S, Duru N, Adali E. Improved fuzzy art method for initializing K-means. Int J Comput Intell Syst 2010;3(3):274. <http://dx.doi.org/10.2991/ijcis.2010.3.3.3>.
4. Liew WS, Loo CK, Wermter S. Emotion recognition using explainable genetically optimized fuzzy ART ensembles. IEEE Access 2021;9:61513–31. [http://dx.doi.](http://dx.doi.org/10.1109/access.2021.3072120) [org/10.1109/access.2021.3072120](http://dx.doi.org/10.1109/access.2021.3072120).
5. Djellali C, adda M, Moutacalli MT. A comparative study on fuzzy clustering for cloud computing. Taking web service as a case. Procedia Comput Sci 2021;184:622–7. <http://dx.doi.org/10.1016/j.procs.2021.04.024>.
6. Bouguila N. On multivariate binary data clustering and feature weighting. Comput Stat Data Anal 2010;54(1):120–34. [http://dx.doi.org/10.1016/j.csda.](http://dx.doi.org/10.1016/j.csda.2009.07.013) [2009.07.013](http://dx.doi.org/10.1016/j.csda.2009.07.013).
7. Wang X, Kaban A. Finding uninformative features in binary data. In: Lecture notes in computer science. Springer Berlin Heidelberg; 2005, p. 40–7. [http:](http://dx.doi.org/10.1007/11508069_6)

[//dx.doi.org/10.1007/11508069\_6](http://dx.doi.org/10.1007/11508069_6).

1. Ordonez C. Clustering binary data streams with K-means. In: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery - DMKD '03. ACM Press; 2003, [http://dx.doi.org/10.1145/882082.](http://dx.doi.org/10.1145/882082.882087) [882087](http://dx.doi.org/10.1145/882082.882087).
2. Kaban A, Girolami M. Initialized and guided EM-clustering of sparse binary data with application to text based documents. In: Proceedings 15th international conference on pattern recognition. IEEE Comput. Soc; 2000, [http://dx.doi.org/](http://dx.doi.org/10.1109/icpr.2000.906182) [10.1109/icpr.2000.906182](http://dx.doi.org/10.1109/icpr.2000.906182).
3. [Cai J, Hao J, Yang H, Zhao X, Yang Y. A review on semi-supervised clustering.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb10) [Inform Sci 2023.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb10)
4. [Kim T, Hwang I, Kang GC, Choi W-S, Kim H, Zhang BT. Label propaga-](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb11) [tion adaptive resonance theory for semi-supervised continuous learning. In:](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb11) [ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb11) [processing. IEEE; 2020, p. 4012–6.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb11)
5. [Kim T, Hwang I, Lee H, Kim H, Choi W-S, Lim JJ, et al. Message passing adaptive](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb12) [resonance theory for online active semi-supervised learning. In: International](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb12) [conference on machine learning. PMLR; 2021, p. 5519–29.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb12)
6. [Akpan VA, Agbogun JB. Hybrid adaptive neural-fuzzy algorithms based on](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb13) [adaptive resonant theory with adaptive clustering algorithms for classifica-](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb13) [tion, prediction, tracking and adaptive control applications. Am J Intell Syst](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb13) [2022;12(1):9–33.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb13)
7. [Masuyama N, Amako N, Yamada Y, Nojima Y, Ishibuchi H. Adaptive resonance](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb14) [theory-based topological clustering with a divisive hierarchical structure capable](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb14) [of continual learning. IEEE Access 2022;10:68042–56.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb14)
8. Elnabarawy I, Tauritz DR, Wunsch DC. Evolutionary computation for the automated design of category functions for fuzzy ART. In: Proceedings of the genetic and evolutionary computation conference companion. ACM; 2017, [http:](http://dx.doi.org/10.1145/3067695.3082056)

[//dx.doi.org/10.1145/3067695.3082056](http://dx.doi.org/10.1145/3067695.3082056).

1. [Basu S, Banerjee A, Mooney R. Semi-supervised clustering by seeding. In:](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb16) [Proceedings of 19th international conference on machine learning. Citeseer;](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb16) [2002.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb16)
2. [Taghizabet A, Tanha J, Amini A, Mohammadzadeh J. A semi-supervised](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb17) [clustering approach using labeled data. Sci Iran 2023;30(1):104–15.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb17)
3. Bair E. Semi-supervised clustering methods. Wiley Interdiscip Rev Comput Stat 2013;5(5):349–61. <http://dx.doi.org/10.1002/wics.1270>.
4. Qin Y, Ding S, Wang L, Wang Y. Research progress on semi-supervised cluster- ing. Cogn Comput 2019;11(5):599–612. [http://dx.doi.org/10.1007/s12559-019-](http://dx.doi.org/10.1007/s12559-019-09664-w)

[09664-w](http://dx.doi.org/10.1007/s12559-019-09664-w).

1. [Pei Y, Zaïane O. A synthetic data generator for clustering and outlier analysis.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb20) [Educ Res Arch 2006.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb20)
2. [Dahmen J, Cook D. SynSys: A synthetic data generation system for healthcare](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb21) [applications. Sensors 2019;19(5):1181.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb21)
3. [Lahariya M, Benoit DF, Develder C. Synthetic data generator for electric vehicle](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb22) [charging sessions: modeling and evaluation using real-world data. Energies](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb22) [2020;13(16):4211.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb22)
4. Jiang W, Song S, Hou L, Zhao H. A set of efficient methods to generate high-dimensional binary data with specified correlation structures. Amer Statist 2020;1–13. <http://dx.doi.org/10.1080/00031305.2020.1816213>.
5. Zellinger MJ, Bühlmann P. repliclust: Synthetic data for cluster analysis. 2023, arXiv preprint [arXiv:2303.14301](http://arxiv.org/abs/2303.14301).
6. Lunn A, Davies SJ. A note on generating correlated binary variables. Biometrika 1998;85(2):487–90. <http://dx.doi.org/10.1093/biomet/85.2.487>.
7. Farrell PJ, Rogers-Stewart K. Methods for generating longitudinally correlated binary data. Internat Statist Rev 2008;76(1):28–38. [http://dx.doi.org/10.1111/j.](http://dx.doi.org/10.1111/j.1751-5823.2007.00017.x) [1751-5823.2007.00017.x](http://dx.doi.org/10.1111/j.1751-5823.2007.00017.x).
8. Sengupta S, Ghosh T, Dan PK, Chattopadhyay M. Hybrid fuzzy-ART based K- means clustering methodology to cellular manufacturing using operational time. 2012, arXiv preprint [arXiv:1212.5101](http://arxiv.org/abs/1212.5101).
9. [da Silva LEB, Elnabarawy I, Wunsch II DC. A survey of adaptive reso-](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb28) [nance theory neural network models for engineering applications. Neural Netw](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb28) [2019;120:167–203.](http://refhub.elsevier.com/S2590-0056(23)00044-9/sb28)