Multilayer Clustering Based on Adaptive Resonance Theory for Noisy Environments

Narito Amako
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
narito.amako@ci.cs.osakafu-u.ac.jp

Naoki Masuyama
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
masuyama@cs.osakafu-u.ac.jp

Chu Kiong Loo
Faculty of Computer Science and Information Technology
University of Malaya
Kuala Lumpur, Malaysia
ckloo.um@um.edu.my

Yusuke Nojima
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
nojima@cs.osakafu-u.ac.jp

Yiping Liu
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
yiping0liu@gmail.com

Hisao Ishibuchi
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
hisao@sustech.edu.cn

Abstract—Clustering based on Adaptive Resonance Theory (ART) has been actively studied. In previous studies, ART-based clustering algorithms with a topological structure have been proposed and showed their superior self-organizing ability. However, this method deteriorates the clustering performance at high noise ratios. In this paper, we propose a multilayer clustering algorithm based on a topological ART-based clustering for improving a noise reduction ability. Simulation experiments show that the proposed algorithm achieves excellent clustering performance on a 2D synthetic dataset in high noise environments.

Keywords—Adaptive Resonance Theory (ART), Multilayer Clustering, Correntropy, Noisy Environment

I. Introduction

With recent technological advances, robots can obtain a massive amount of information through various types of sensors for environmental perception. These robots require intelligent technology to extract useful information from data continually and efficiently.

Clustering is a simple and promising method that can continually extract knowledge hiding in data. Therefore, clustering has been applied to several studies in the robotics field, such as an extraction algorithm of geometric information of an environmental map generated by Simultaneous Localization and Mapping (SLAM) [1], [2] and a knowledge extraction algorithm for humanoid robots [3]. In general, these studies use Growing Neural Gas (GNG)-based clustering algorithms (e.g., Adjusted Self-Organizing Incremental Neural Networks (ASOINN) [4]) thanks to their applicability and adaptability. However, GNG-based clustering algorithms suffer from the plasticity-stability dilemma [5], i.e., the trade-off between catastrophic forgetting and continually learning new knowledge.

In the research field of clustering, Adaptive Resonance Theory (ART) [6] is a successful approach for handling the plasticity-stability dilemma. Several types of ART-based clustering algorithms have already been proposed. Fuzzy ART (FA) [7] and Bayesian ART (BA) [8] are considered as fundamental algorithms. Kernel BA (KBA) [9] is an extension algorithm of BA that uses kernel Bayes' rule [10] and Correntropy Induced Metric (CIM) [11] to realize a faster and more stable self-organizing ability than FA and BA. Local distribution-based clustering with BA is developed not only to improve its self-organizing ability but also to handle highly imbalanced data distributions [12]. TopoART [13] is developed by applying a hierarchical structure to FA for implementing a noise reduction ability and an implicit topological structure on FA. The hierarchical structure of TopoART effectively removes noise information without complicated functions. Recent studies have integrated a topological structure (i.e., edges between nodes) into the ART-based clustering to explicitly represent the relationship between nodes [14], [15], [16]. Specifically, Fast Topological CIM-based ART (FTCA) [16] is considered to be state-of-the-art ART-based topological clustering that shows a fast and stable self-organizing ability. However, FTCA cannot maintain its superior performance in a high noise environment.

As mentioned earlier, TopoART can be effectively reduced noise information through its simple hierarchical approach. In this paper, we propose Multilayer FTCA (MFTCA) by

This research was supported in part by the Ministry of Education, Culture, Sports, Science and Technology - JAPAN (MEXT) Leading Initiative for Excellent Young Researchers (LEADER), in part by the Frontier Research Grant (Project No. FG003-17AFR) from University of Malaya, ONRG grant (Project No.: ONRG-NICOP-N62909-18-1-2086) from Office of Naval Research Global, UK, in part by the International Collaboration Fund (Project No: IF0318M1006) from MESTECC, Malaysia, in part by the National Natural Science Foundation of China (Grant No. 61876075), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

Corresponding author: Hisao Ishibuchi (email: hisao@sustech.edu.cn).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

introducing a multilayer structure that is inspired by TopoART for improving noise reduction ability. Although the multilayer approach in this paper is inspired by TopoART, several advantages can be expected thanks to the superior self-organizing ability of FTCA. The contributions of this paper are summarized as follows:

(i) MFTCA demonstrates a strong noise reduction ability.

(ii) MFTCA automatically generates the necessary and sufficient number of nodes for self-organization.

(iii) The number of layers can be specified arbitrarily.

This paper is organized as follows. Section II describes the details of the proposed algorithm. Section III presents simulation experiments to examine the self-organizing ability of the proposed algorithm under high noise environments. Section IV concludes this paper.

II. Proposed Algorithm

In this section, first, we briefly describe CIM and FTCA. Then, MFTCA is presented in detail.

A. Correntropy Induced Metric

Correntropy [11] is a nonlinear measure of similarity between two vectors $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_d)$ which is defined as follows:

$$C(\mathbf{x}, \mathbf{y}) = \mathbf{E}[\kappa(\mathbf{x} - \mathbf{y})], \tag{1}$$

where κ is a kernel that satisfies the Mercer's Theorem and induces a Reproducing Kernel Hilbert Space (RKHS), and $\mathbf{E}[\cdot]$ represents the expectation operation. In practice, only a finite number of data points are available. In this case, the correntropy is defined as follows [11]:

$$\hat{C}(\mathbf{x}, \mathbf{y}) = \frac{1}{d} \sum_{i=1}^{d} \kappa(x_i - y_i). \tag{2}$$

Here we use the following Gaussian kernel because it is most frequently used in the correntropy in the literature.

$$\kappa_\sigma(x_i - y_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - y_i)^2}{2\sigma^2}\right), \tag{3}$$

where σ denotes a kernel bandwidth.

Correntropy Induced Metric (CIM) is induced by the correntropy and is defined as follows:

$$\mathrm{CIM}(\mathbf{x}, \mathbf{y}, \sigma) = (\kappa_\sigma(0) - \hat{C}(\mathbf{x}, \mathbf{y}))^{1/2}, \tag{4}$$

where $\kappa_\sigma(0) = 1/\sigma\sqrt{2\pi}$.

B. Learning in Fast Topological CIM-based ART

The learning procedure of FTCA is roughly divided into five parts as follows: 1) Initialization Process, 2) Winner Node Selection, 3) Vigilance Test, 4) Node Learning, and 5) Topology Adjustment.

Let us assume that we have a set of data points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ $(\mathbf{x}_l \in \mathbb{R}^d)$ where L denotes the number of data points. We denote a set of generated nodes in an FTCA network by $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$ $(\mathbf{y}_k \in \mathbb{R}^d)$ where K denotes the number of nodes. In addition, in FTCA, all the nodes $\mathbf{Y}$ have the same value of the kernel bandwidth σ.

1) Initialization Process: By using N data points out of a set of the data points $\mathbf{X}$, the kernel bandwidth σ is specified in kernel density estimation with the Gaussian kernel [17]. When there is no node in the FTCA network, the first and second data points become nodes (i.e., $\mathbf{y}_1 = \mathbf{x}_1$ and $\mathbf{y}_2 = \mathbf{x}_2$) with the kernel bandwidth σ. The kernel density estimation with the Gaussian kernel is defined as follows:

$$s_i = \left(\frac{4}{2+d}\right)^{\frac{1}{4+d}} \Gamma_i N^{-\frac{1}{4+d}}, \tag{5}$$

where Γ_i represents the standard deviation for the i-th dimension of the N data points, and s_i is the bandwidth for the i-th dimension. In CIM, the kernel bandwidth σ is a scalar value. The median of $\mathbf{S} = \{s_1, s_2, \ldots, s_d\}$ is used as a value of σ.

2) Winner Node Selection: Once data point $\mathbf{x}_l$ is presented to the network, the 1st and 2nd winner nodes are determined based on CIM as follows:

$$k_1 = \arg\min_{k \in K} [\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_k, \sigma)], \tag{6}$$

$$k_2 = \arg\min_{k \in K/\{k_1\}} [\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_k, \sigma)], \tag{7}$$

where k_1 and k_2 denote the indexes of the 1st and 2nd winner nodes, respectively.

3) Vigilance Test: Vigilance Test classifies the positional relationship between $\mathbf{x}_l$ and the winner nodes into three cases.

- Case I

$$\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_{k_1}, \sigma) > V, \tag{8}$$

where V denotes a predefined similarity threshold. The condition (8) means that $\mathbf{x}_l$ does not belong to either $\mathbf{y}_{k_1}$ or $\mathbf{y}_{k_2}$.

- Case II

$$\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_{k_1}, \sigma) \le V, \tag{9}$$

$$\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_{k_2}, \sigma) > V. \tag{10}$$

The conditions (9) and (10) mean that $\mathbf{x}_l$ belongs to $\mathbf{y}_{k_1}$ but not to $\mathbf{y}_{k_2}$.

- Case III

$$\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_{k_1}, \sigma) \le V, \tag{11}$$

$$\mathrm{CIM}(\mathbf{x}_l, \mathbf{y}_{k_2}, \sigma) \le V. \tag{12}$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

The conditions (11) and (12) mean that $\mathbf{x}_l$ belongs to both $\mathbf{y}_{k_1}$ and $\mathbf{y}_{k_2}$.

4) Node Learning: If $\mathbf{x}_l$ is classified as Case I by the vigilance test (i.e., (8) is satisfied), a new node is defined as $\mathbf{y}_{K+1} = \mathbf{x}_l$.

If $\mathbf{x}_l$ is classified as Case II by the vigilance test (i.e., (9) and (10) are satisfied), the location of the winner node $\mathbf{y}_{k_1}$ is updated as follows:

$$\mathbf{y}_{k_1} = \mathbf{y}_{k_1} + \frac{1}{M_{k_1}}(\mathbf{x}_l - \mathbf{y}_{k_1}), \tag{13}$$

where M_{k_1} is the number of data points that belong to $\mathbf{y}_{k_1}$. Furthermore, M_{k_1} is updated as follows:

$$M_{k_1} = M_{k_1} + 1. \tag{14}$$

If $\mathbf{x}_l$ is classified as Case III by the vigilance test (i.e., (11) and (12) are satisfied), $\mathbf{y}_{k_1}$ and M_{k_1} are updated by (13) and (14), respectively. Moreover, all neighbor nodes $\mathbf{y}_{k_1}^{+}$, which are connected to $\mathbf{y}_{k_1}$ by a single edge, are updated as follows:

$$\mathbf{y}_{k_1}^{+} = \mathbf{y}_{k_1}^{+} + \frac{1}{10M_{k_1}^{+}}(\mathbf{x}_l - \mathbf{y}_{k_1}^{+}), \tag{15}$$

where $M_{k_1}^{+}$ denotes the number of data points that belong to $\mathbf{y}_{k_1}^{+}$. Furthermore, in Case III, a new edge between $\mathbf{y}_{k_1}$ and $\mathbf{y}_{k_2}$ is created if there is no edge between them.

5) Topology Adjustment: The topology adjustment consists of Node Deletion and Edge Deletion. These operations are performed every λ iteration (i.e., after λ data points are presented).

a) Node Deletion: Nodes that do not have any edges are removed.

b) Edge Deletion: Edge intersections are checked by the cross-product-based detection algorithm [18]. If there is an edge intersection, the edge with the largest value of CIM is removed.

C. Proposed Multilayer FTCA: MFTCA

The proposed MFTCA algorithm has a multilayer structure of layers. Each layer has a self-organizing network. The learning procedure of a network in each layer is the same as FTCA. Let us denote the number of layers in MFTCA by H. In the first layer, all the data points are used for the learning. In the learning at the 2nd layer, only the data points that satisfy (9) at the 1st layer become inputs. The same operation is performed in the learning from the 3rd layer to Hth layer. That is, the data points that satisfy (9) at the hth layer become inputs at (h+1)th layer. Fig. 1 shows the transition mechanism for a data point from the hth layer to the (h+1)th layer.

The overall learning procedure of MFTCA is presented in Algorithm 1.

Algorithm 1: Learning Algorithm of MFTCA

Input:
the data point: $\mathbf{x}_l \in \mathbb{R}^d$,
the current level of layer: h,
the total number of layers: H,
the existing nodes at each layer: $\mathbf{Y}_h = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}(\mathbf{y}_k \in \mathbb{R}^d)$,
the number of data points that belong to each node at each layer: $\mathbf{M}_h = (M_1, M_2, \ldots, M_K)$,
the kernel bandwidth for CIM at each layer: σ_h,
the similarity threshold: V,
and the topology adjustment cycle: λ.

Output:
the updated nodes at each layer: $\mathbf{Y}_h$,
and the number of data points that belong to each node at each layer: $\mathbf{M}_h = (M_1, M_2, \ldots, M_K)$.

```
function LearningMFTCA($\mathbf{x}_l$, $h$, $\mathbf{Y}_h$, $\mathbf{M}_h$, $\sigma_h$, $V$, $\lambda$)
    if $K < 2$ then
        Create the new node as $\mathbf{y}_{K+1} = \mathbf{x}_l$.
        if $h = 1$ then
            Set the kernel bandwidth $\sigma_h$ to a median of
            $\mathbf{S} = \{s_1, s_2, \ldots, s_d\}$ calculated by (5).
        else
            Set the kernel bandwidth $\sigma_h = \sigma_{h-1}$.
    else
        Search the indexes of winner nodes $k_1$ and $k_2$ by (6)
        and (7), respectively.
        if CIM($\mathbf{x}_l$, $\mathbf{y}_{k1}$, $\sigma$) > $V$ then
            Create the new node as $\mathbf{y}_{K+1} = \mathbf{x}_l$.
        else
            Update the state of $\mathbf{y}_{k1}$ by (13).
            Update the state of $M_{k1}$ by (14).
            if $h < H$
                LearningMFTCA($\mathbf{x}_l$, $h$+1, $\mathbf{Y}_{h+1}$, $\mathbf{M}_{h+1}$,
                $\sigma_{h+1}$, $V$, $\lambda$).
            if CIM($\mathbf{x}_l$, $\mathbf{y}_{k2}$, $\sigma$) $\leq$ $V$ then
                Update the state of neighbor nodes $\mathbf{y}_{k1}^{+}$
                by (15).
                Create a new edge between $\mathbf{y}_{k1}$ and $\mathbf{y}_{k2}$.
    if the number of data point inputs $l$ is multiple of a
    topology adjustment cycle $\lambda$ then
        forall $k \in 1, \ldots, K$ do
            if $\mathbf{y}_k$ does not have any edge then
                Remove $\mathbf{y}_k$ from $\mathbf{Y}_h$.
        Remove an edge with an intersection based on a
        state of CIM.
    return $\mathbf{Y}_h$ and $\mathbf{M}_h$.
```

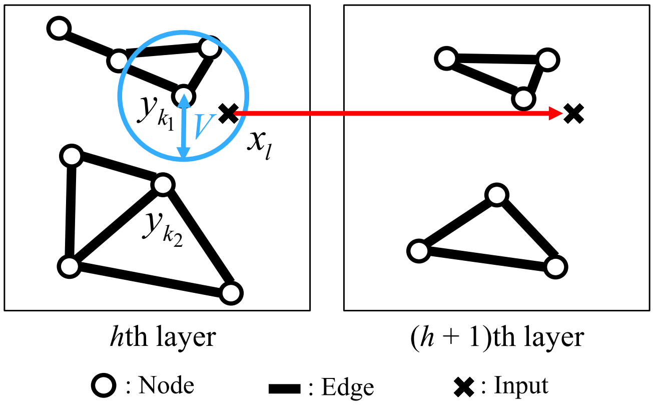


Fig. 1. Transition mechanism for a data point x_l from the hth layer to the (h+1)th layer.

III. Simulation Experiments

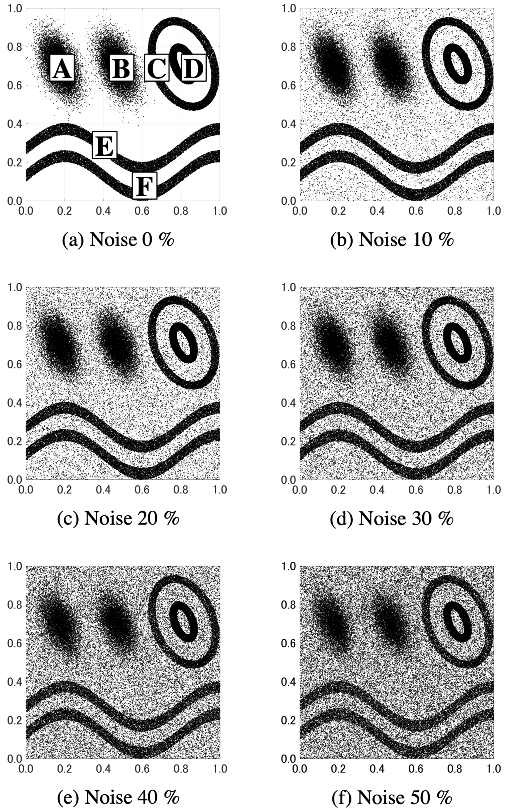
We compare MFTCA with FTCA [16], ASOINN [4], and TopoART [13]. As mentioned in section II-C, the result of FTCA is the same as the result of MFTCA in the 1st layer.

A. Experimental Conditions

We examine the performance of the self-organizing ability from qualitative and quantitative perspectives. For evaluating the self-organizing ability, we use a 2D synthetic dataset as shown in Fig. 2. The dataset is divided into six distributions as A, B, C, D, E, and F as shown in Fig. 2(a), which consists of 90k data points in total (15k data points in each distribution). Here, A and B are 2D Gaussian distributions. C and D are concentric-ring distributions. E and F are sinusoidal distributions. The experiments are performed on the 2D synthetic dataset with the noise ratios of 0%, 10%, 20%, 30%, 40%, and 50% (i.e., Figs. 2(a)-(f)). Data points are inputted randomly. Throughout Section III, parameters for each algorithm are set as listed in Table I. The parameters of FTCA are the same as in [16]. The total number of layers of MFTCA is set as $H = 5$. We remark that the total number of layers H can be set as any positive integer thanks to the multilayer approach of MFTCA. The rest of the parameters in MFTCA (i.e., a similarity threshold V and a topology construction cycle λ) are set to the same value as FTCA for providing a fair comparison. The parameters of ASOINN and TopoART are set to the same value as in [15].

TABLE I. Parameter Settings for MFTCA, FTCA, ASOINN, and TopoART

Algorithm	Parameter	Value
MFTCA	the total number of layers H a similarity threshold V a topology construction cycle λ	5 0.2 50
FTCA	a similarity threshold V a topology construction cycle λ	0.2 50
ASOINN	a node insert cycle λ a maximum age of edge age_{MAX} a parameter for node deletion c	400 25 0.5
TopoART	a node insert cycle τ a learning rate β a parameter for node deletion ϕ a vigilance parameter ρ	(300,300) (1.00, 0.65) (2, 2) (0.92, 0.96)



(a) Noise 0 %

(b) Noise 10 %

(c) Noise 20 %

(d) Noise 30 %

(e) Noise 40 %

(f) Noise 50 %

Fig. 2. 2D synthetic dataset.

B. Qualitative Evaluation

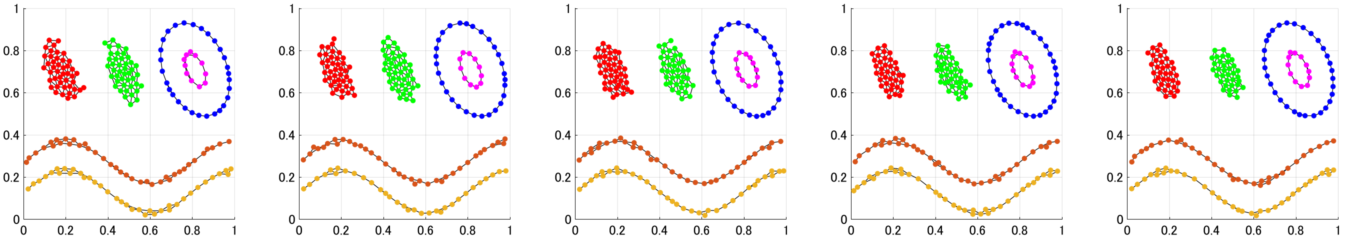
We show the generated topological networks and visually evaluate whether the distributions of the data can properly be represented.

The generated topological networks of MFTCA are presented in Figs. 3-8. Each figure shows the results for noise ratios of 0%, 10% 20%, 30%, 40%, and 50%. In the case that the noise ratio is 50%, we show the 6th to 10th layers to observe the effect of multilayer structures. The results of FTCA are the same as the result of the 1st layer of MFTCA. Figs. 9 and 10 show the generated topological networks of ASOINN and TopoART, respectively.

Focusing on the self-organizing results with 0% noise ratio, each algorithm successfully finds six distributions of data. However, as the noise ratio increases, the topological networks of FTCA, ASOINN, and TopoART tend to connect between different distributions. As a result, these algorithms generate over-connected networks. TopoART tends to make a lot of nodes (i.e., hyper rectangles) because TopoART does not have a mechanism to remove the isolated node like FTCA. On the other hand, MFTCA can generate well-organized networks even in high noise environments by removing noise information through multilayer structures. It means that MFTCA succeeded in acquiring a strong noise reduction ability. In particular, in the dataset with a noise ratio of 40% (i.e., Fig. 7, Fig. 9(e), and Fig. 10(e)), FTCA, ASOINN, and TopoART cannot reduce noise information at all. In contrast, MFTCA successfully organizes

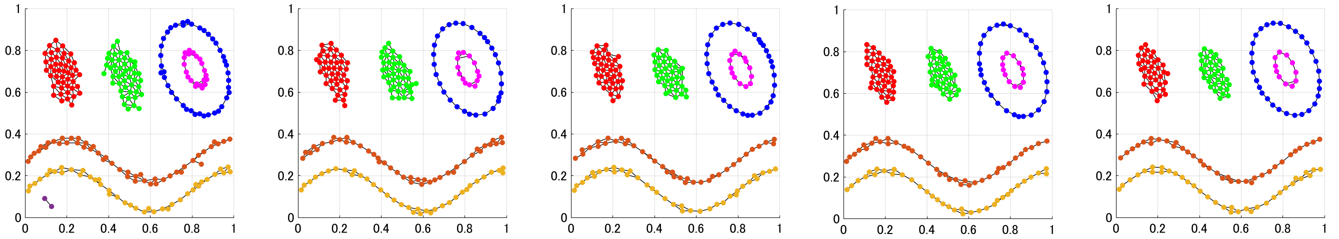
978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the network according to the distributions of data through multilayer structures. However, one drawback of MFTCA is that when the level of the layer is too large (i.e., Fig. 8(i)-(j)), the number of the nodes decreases and the connections are excessively cut. The cause of this drawback is that due to the transition mechanism for data points (i.e., Fig. 9), the larger the number of layers is, the less the number of input data points is. Therefore when the number of layers is too large, the number of input data points highly decreases. As a result, the nodes and the edges are not generated well.



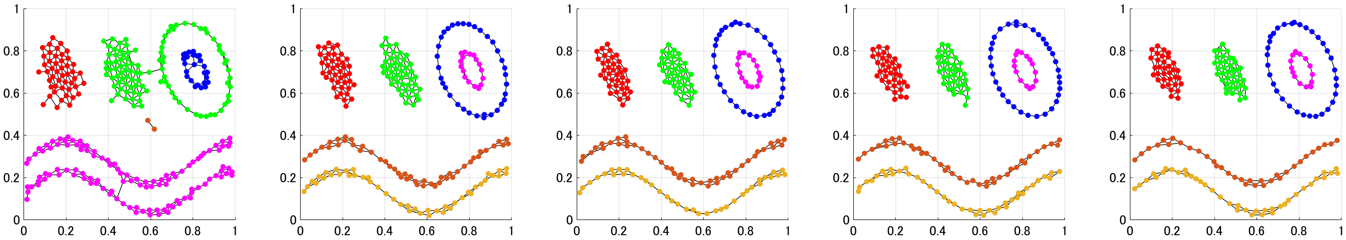
(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

Fig. 3. Generated networks of MFTCA (noise ratio: 0%).



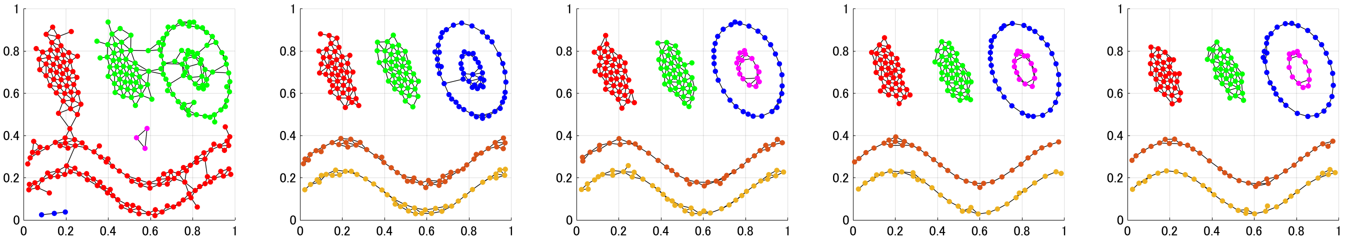
(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

Fig. 4. Generated networks of MFTCA (noise ratio: 10%).



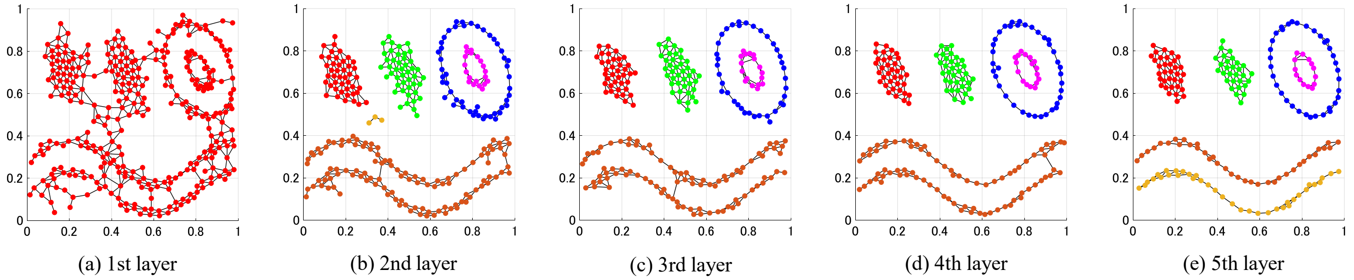
(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

Fig. 5. Generated networks of MFTCA (noise ratio: 20%).



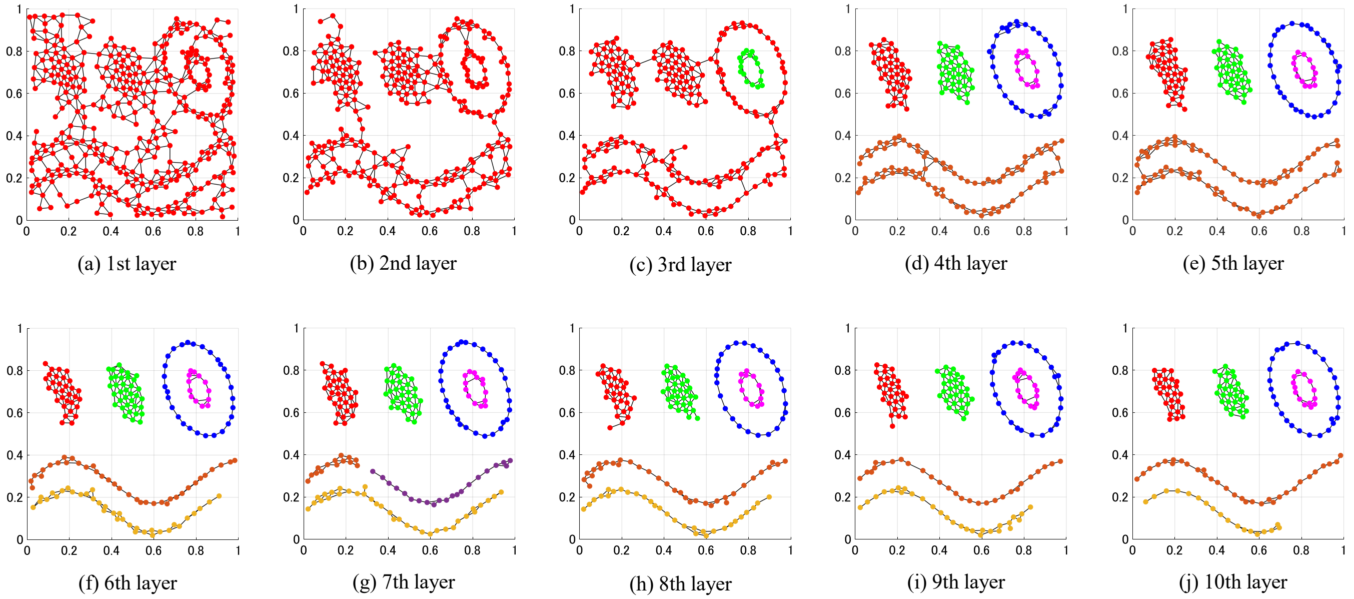
(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

Fig. 6. Generated networks of MFTCA (noise ratio: 30%).



(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

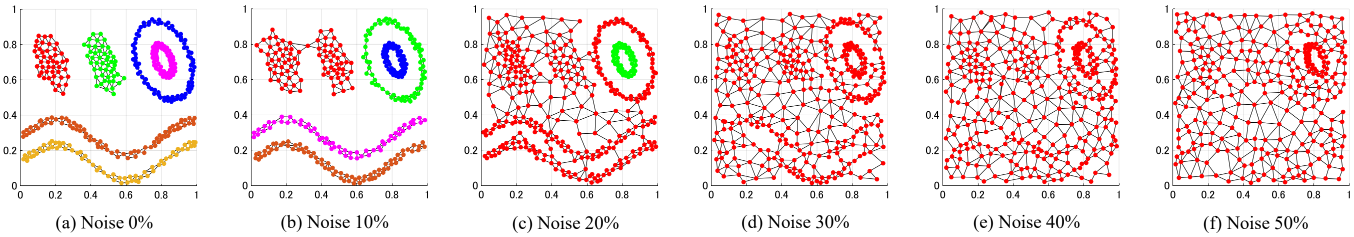
Fig. 7. Generated networks of MFTCA (noise ratio: 40%).



(a) 1st layer (b) 2nd layer (c) 3rd layer (d) 4th layer (e) 5th layer

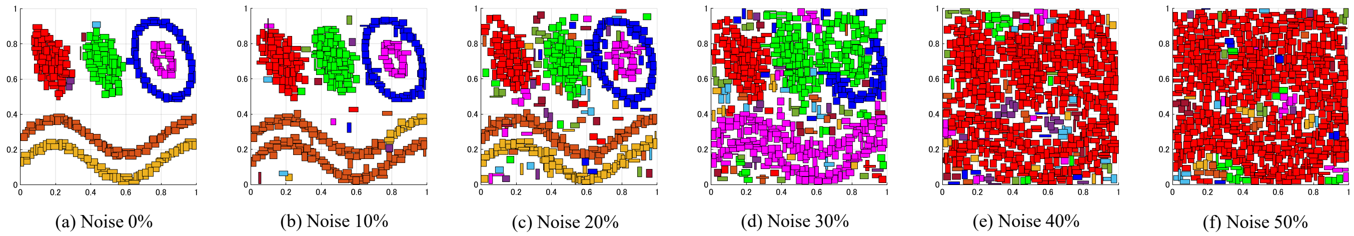
(f) 6th layer (g) 7th layer (h) 8th layer (i) 9th layer (j) 10th layer

Fig. 8. Generated networks of MFTCA (noise ratio: 50%)



(a) Noise 0% (b) Noise 10% (c) Noise 20% (d) Noise 30% (e) Noise 40% (f) Noise 50%

Fig. 9. Generated networks of ASOINN



(a) Noise 0% (b) Noise 10% (c) Noise 20% (d) Noise 30% (e) Noise 40% (f) Noise 50%

Fig. 10. Generated networks of TopoART

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

C. Quantitative Evaluation

We use the generated topological networks as classifiers and perform classification tasks. In this experiment, Normalized Mutual Information (NMI) [19], Micro and Macro F-measures [20], and Adjusted Rand Index (ARI) [21] are used as performance indicators.

Specifically, the label information is assigned to data points of each distribution A to F in Fig. 6 as class 1 to 6, respectively. Then, 90% of the data points in Fig. 6 are randomly selected as training data to generate a topological network. We add noise data by replacing the part of the training data with uniformly distributed random data points. The remaining 10% of data points are used as test data for the classification task. The label information for each cluster is determined by majority vote from labeled data points that are considered to belong to the connected-nodes (i.e., a cluster). To reduce the influence of the random sampling of data points, 10-fold cross-validation is used. Furthermore, all the experiments are conducted five times to obtain consistent averaging results.

Table II shows the average and standard deviation of NMI, Micro and Macro F-measures, and ARI on the classification tasks. The best value for each metric is indicated in bold. A symbol † represents the results that have a statistically significant difference ($p < 0.05$) from the best result by the Wilcoxon signed-rank test. Table III shows that the average number of nodes and clusters with standard deviations in parentheses. MFTCA presented the best results for any noise ratios. TopoART generated a lot of clusters due to the influence of noise data. On the other hand, MFTCA provides excellent evaluation values of all indicators while maintaining the optimal number of clusters (i.e., six clusters in this experiment). Since ART-based clustering makes input data regarded as a new cluster, FTCA and TopoART tend to generate an excessive number of nodes as the noise ratio increases. In contrast, MFTCA successfully controls the number of nodes by removing noise information. However, the performance of MFTCA deteriorates as the number of layers increases in the cases where the noise is low (i.e., 0% and 10%). This is because that when a data point in the hth layer is input to the $(h+1)$th layer, the data point is used to define a new node in the $(h+1)$th layer. As a result, the number of data points decreases as the number of layers increases (i.e., rows of the number of nodes in Table II).

IV. Conclusion

In this paper, we proposed the multilayer clustering algorithm, called MFTCA based on ART-based topological clustering to realize a strong noise reduction ability. Experimental results showed that MFTCA has high self-organizing ability even when the noise ratio is high. However, our experimental results also showed that the performance of MFTCA was deteriorated when the number of layers (H) is unnecessarily large.

As mentioned in section III-A, parameter specification is an important future research issue in MFTCA. We will work on the development of a mechanism to automatically specify the number of layers based on the internal information obtained from the generated topological networks.

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[2] D. Viejo, J. Garcia-Rodriguez, and M. Cazorla, "Combining visual features and growing neural gas networks for robotic 3D SLAM," *Information Sciences*, vol. 276, pp. 174–185, Aug. 2014.

[3] D. Kimura, R. Nishimura, A. Oguro, and O. Hasegawa, "Ultra-fast multimodal and online transfer learning on humanoid robots," *in ACM/IEEE International Conference on Human-Robot Interaction*, 2013, pp. 165–166.

[4] F. Shen and O. Hasegawa, "A fast nearest neighbor classifier based on self organizing incremental neural network," *Neural Networks*, vol. 21, no. 10, pp. 1537–1547, 2008.

[5] G. A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, no. 3, pp. 77–88, 1988.

[6] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, no. 1, pp. 23–63, 1987.

[7] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.

[8] B. Vigdor and B. Lerner, "The Bayesian ARTMAP," *IEEE Transactions on Neural Networks,* vol. 18, no. 6, pp. 1628–1644, 2007.

[9] N. Masuyama, C. K. Loo, and F. Dawood, "Kernel Bayesian ART and ARTMAP," *Neural Networks*, vol. 98, pp. 76–86, Feb. 2018.

[10] K. Fukumizu, L. Song, and A. Gretton, "Kernel Bayes' rule: Bayesian inference with positive definite kernels," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3753–3783, 2013.

[11] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[12] L. Wang, H. Zhu, J. Meng, and W. He, "Incremental local distribution-based clustering using bayesian adaptive resonance theory," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3496–3504, 2019.

[13] M. Tscherepanow, "TopoART: A topology learning hierarchical ART network," *in Proc. International Conference on Artificial Neural Networks*, Berlin, Germany, Springer, 2010, pp. 157–167.

[14] N. Masuyama, C. K. Loo, and S. Wermter, "A kernel Bayesian adaptive resonance theory with a topological structure," *International Journal of Neural Systems*, vol. 29, no. 5, p. 1850052 (20 pages), 2019.

[15] N. Masuyama, C. K. Loo, H. Ishibuchi, N. Kubota, Y. Nojima, and Y. Liu, "Topological clustering via adaptive resonance theory with information theoretic learning," *IEEE Access*, vol. 7, pp. 76920–76936, 2019.

[16] N. Masuyama, N. Amako, Y. Nojima, Y. Liu, C. K. Loo, and H. Ishibuchi, "Fast topological adaptive resonance theory based on correntropy induced metric," *Proc. of 2019 IEEE Symposium Series on Computational Intelligence*, Xiamen, China, 2019, pp. 2215–2221.

[17] D. J. Henderson and C. F. Parmeter, "Normal reference bandwidths for the general order, multivariate kernel density derivative estimator," *Statistics & Probability Letters*, vol. 82, no. 12, pp. 2198–2205, 2012.

[18] T. H. Cormen, *Introduction to Algorithms.* MIT press, 2009.

[19] A. Strehl and J. Ghosh, "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, Dec. 2002.

[20] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks,'" *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[21] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193-218, 1985.

TABLE II. COMPARISONS OF PERFORMANCE ON CLASSIFICATION TASKS.

Noise Ratio	Measurement			MFTCA			FTCA	ASOINN	TopoART
		1st layer	2nd layer	3rd layer	4th layer	5th layer			
0 [%]	NMI	**0.998** (0.001)	0.997†(0.001)	0.997†(0.001)	0.995†(0.005)	0.989†(0.014)	**0.998** (0.001)	0.993 (0.014)	0.996 (0.006)
	Micro F-measure	**1.000** (0.000)	0.999†(0.000)	0.999†(0.000)	0.998†(0.002)	0.995†(0.008)	**1.000** (0.000)	0.986 (0.035)	0.997 (0.008)
	Macro F-measure	**1.000** (0.000)	0.999†(0.000)	0.999†(0.000)	0.998†(0.002)	0.995†(0.009)	**1.000** (0.000)	0.982 (0.047)	0.997 (0.009)
	ARI	**0.999** (0.001)	0.998†(0.001)	0.998†(0.001)	0.996†(0.006)	0.989†(0.018)	**0.999** (0.001)	0.985 (0.037)	0.995 (0.012)
10 [%]	NMI	0.983 (0.031)	**0.998** (0.001)	0.997†(0.001)	0.996†(0.004)	0.992†(0.010)	0.983 (0.031)	0.771†(0.143)	0.980†(0.023)
	Micro F-measure	0.963 (0.077)	**0.999** (0.000)	0.999†(0.000)	0.999†(0.002)	0.997†(0.005)	0.963 (0.077)	0.605†(0.161)	0.980†(0.039)
	Macro F-measure	0.951 (0.104)	**0.999** (0.000)	0.999†(0.000)	0.999†(0.002)	0.997†(0.005)	0.951 (0.104)	0.516†(0.177)	0.975†(0.050)
	ARI	0.961 (0.079)	**0.999** (0.001)	0.998†(0.001)	0.997†(0.005)	0.993†(0.012)	0.961 (0.079)	0.511†(0.227)	0.972†(0.047)
20 [%]	NMI	0.917†(0.072)	0.996†(0.005)	**0.997** (0.002)	0.994†(0.006)	0.989†(0.018)	0.917†(0.072)	0.411†(0.192)	0.893†(0.068)
	Micro F-measure	0.813†(0.155)	0.996†(0.010)	**0.999** (0.001)	0.998†(0.004)	0.995†(0.011)	0.813†(0.155)	0.303†(0.062)	0.843†(0.129)
	Macro F-measure	0.753†(0.203)	0.995†(0.014)	**0.999** (0.001)	0.998†(0.004)	0.995†(0.011)	0.753†(0.203)	0.191†(0.066)	0.806†(0.166)
	ARI	0.810†(0.155)	0.995†(0.012)	**0.998** (0.002)	0.995†(0.008)	0.989†(0.023)	0.810†(0.155)	0.117†(0.055)	0.809†(0.134)
30 [%]	NMI	0.811†(0.136)	0.978 (0.032)	**0.995** (0.007)	0.995†(0.005)	0.990†(0.012)	0.811†(0.136)	0.060†(0.127)	0.757†(0.103)
	Micro F-measure	0.643†(0.195)	0.949 (0.080)	0.996†(0.012)	**0.998** (0.003)	0.996†(0.006)	0.643†(0.195)	0.185†(0.044)	0.649†(0.160)
	Macro F-measure	0.546†(0.233)	0.933 (0.106)	0.995†(0.015)	**0.998** (0.003)	0.996†(0.006)	0.546†(0.233)	0.068†(0.044)	0.572†(0.195)
	ARI	0.614†(0.228)	0.946 (0.083)	0.994†(0.014)	**0.996** (0.006)	0.991†(0.014)	0.614†(0.228)	0.017†(0.036)	0.575†(0.181)
40 [%]	NMI	0.441†(0.136)	0.903†(0.032)	0.980 (0.007)	**0.991** (0.005)	0.986†(0.012)	0.441†(0.136)	0.000†(0.000)	0.555†(0.204)
	Micro F-measure	0.363†(0.195)	0.793†(0.080)	0.959 (0.012)	0.991†(0.003)	**0.992** (0.006)	0.363†(0.195)	0.166†(0.003)	0.430†(0.141)
	Macro F-measure	0.236†(0.233)	0.729†(0.106)	0.946 (0.015)	0.988†(0.003)	**0.992** (0.006)	0.236†(0.233)	0.047†(0.001)	0.315†(0.146)
	ARI	0.278†(0.228)	0.786†(0.083)	0.956 (0.014)	**0.987** (0.006)	0.984†(0.014)	0.278†(0.228)	0.000†(0.000)	0.340†(0.201)
50 [%]	NMI	0.149†(0.136)	0.773†(0.032)	0.925†(0.007)	0.969 (0.005)	**0.973** (0.012)	0.149†(0.136)	0.000†(0.000)	0.166†(0.203)
	Micro F-measure	0.222†(0.195)	0.602†(0.080)	0.838†(0.012)	0.943 (0.003)	**0.969** (0.006)	0.222†(0.195)	0.165†(0.004)	0.220†(0.074)
	Macro F-measure	0.100†(0.233)	0.494†(0.106)	0.787†(0.015)	0.925 (0.003)	**0.962** (0.006)	0.100†(0.233)	0.047†(0.001)	0.108†(0.067)
	ARI	0.075†(0.228)	0.582†(0.083)	0.832†(0.014)	0.936 (0.006)	**0.957** (0.014)	0.075†(0.228)	0.000†(0.000)	0.062†(0.118)

The best value for each metric is indicated by bold. A symbol † represents the result which has a statistically significant difference (p<0.05) from the best result by the Wilcoxon signed-rank test.

TABLE III. RESULTS OF NUMBER OF NODES AND CLUSTERS ON CLASSIFICATION TASKS.

Noise Ratio	Measurement			MFTCA			FTCA	ASOINN	TopoART
		1st layer	2nd layer	3rd layer	4th layer	5th layer			
0 [%]	Number of Nodes	216.40 (21.37)	206.00 (20.35)	197.10 (19.76)	189.00 (15.12)	183.00 (11.38)	216.40 (21.37)	363.70 (12.05)	362.30 (7.32)
	Number of Clusters	6.00 (0.00)	6.00 (0.00)	6.10 (0.27)	6.40 (0.78)	6.70 (1.24)	6.00 (0.00)	5.90 (0.21)	17.00 (5.17)
10 [%]	Number of Nodes	243.10 (16.67)	212.90 (18.93)	199.00 (19.15)	190.00 (15.34)	183.00 (12.12)	243.10 (16.67)	357.50 (12.48)	412.80 (9.91)
	Number of Clusters	5.90 (0.55)	6.00 (0.06)	6.00 (0.13)	6.20 (0.43)	6.60 (1.02)	5.90 (0.55)	3.60 (0.96)	37.30 (7.61)
20 [%]	Number of Nodes	262.00 (12.32)	221.20 (17.69)	205.40 (18.90)	192.20 (14.66)	182.70 (11.46)	262.00 (12.32)	365.90 (14.06)	505.20 (15.07)
	Number of Clusters	5.10 (0.99)	6.00 (0.13)	6.10 (0.32)	6.20 (0.51)	6.50 (0.83)	5.10 (0.99)	1.80 (0.38)	81.70 (11.60)
30 [%]	Number of Nodes	282.30 (8.91)	230.50 (11.40)	208.00 (15.62)	193.00 (11.19)	182.60 (10.02)	282.30 (8.91)	349.40 (18.41)	603.40 (15.08)
	Number of Clusters	4.60 (1.33)	5.80 (0.50)	6.10 (0.29)	6.30 (0.51)	6.60 (0.79)	4.60 (1.33)	1.10 (0.25)	105.20 (12.70)
40 [%]	Number of Nodes	311.30 (11.69)	253.70 (17.55)	217.70 (18.13)	197.10 (14.39)	181.40 (12.52)	311.30 (11.69)	316.20 (18.41)	683.40 (15.08)
	Number of Clusters	3.10 (1.59)	5.00 (1.28)	6.10 (1.12)	6.40 (0.99)	6.70 (1.13)	3.10 (1.59)	1.00 (0.25)	85.10 (12.70)
50 [%]	Number of Nodes	345.40 (14.16)	271.60 (17.23)	229.90 (14.11)	202.20 (11.11)	183.10 (10.98)	345.40 (14.16)	291.00 (16.08)	739.70 (13.36)
	Number of Clusters	2.00 (1.34)	4.00 (1.40)	5.30 (1.28)	6.20 (1.35)	6.70 (1.14)	2.00 (1.34)	1.00 (0.00)	61.20 (11.77)

978-1-7281-6926-2/20/$31.00 ©2020 IEEE