Approach to collaborative fuzzy clustering in large data analysis

Dang Trong Hop, Mai Dinh Sinh, Do Viet Duc, Ngo Thanh Long

***Abstract*— When data sets have one or more similar characteristics, the clustering in each of these data sets will have an effect on the other data sets. However, for various reasons such as data security issues, these data cannot be stored centrally but in different places. Collaborative clustering is a clustering technique that allows to performance of local clustering on each sub-data set and to exchange of information with other data sets. A collaborative process will be performed to adjust the clustering results on each subset to achieve better clustering results on the subsets of data. This paper presents a collaborative fuzzy clustering approach in big data analysis based on a high-performance computational model to improve the computation speed. Experiments on the Kitsune network attack dataset show that the proposed algorithm significantly improves the calculation speed compared to the previous method.**

***Tóm tắt*— Khi các tập dữ liệu có một hoặc nhiều đặc điểm tương đồng với nhau, việc phân cụm trong mỗi tập dữ liệu này sẽ có ảnh hưởng đến các tập dữ liệu khác. Tuy nhiên, vì nhiều lý do khác nhau như vấn đề bảo mật dữ liệu, những dữ liệu này không thể được lưu trữ tập trung mà lưu trữ ở những nơi khác nhau. Mô hình phân cụm cộng tác là kỹ thuật phân cụm cho phép thực hiện tại mỗi tập dữ liệu con, và thực hiện trao đổi thông tin với các tập dữ liệu khác. Quá trình cộng tác sẽ được thực hiện để điều chỉnh kết quả phân nhóm trên mỗi tập con để đạt được kết quả phân nhóm tốt hơn trên các tập dữ liệu con. Bài báo này trình bày một cách tiếp cận phân cụm mờ cộng tác trong phân tích dữ liệu lớn dựa trên mô hình tính toán hiệu năng cao để cải thiện tốc độ tính toán của thuật toán được đề xuất. Thử nghiệm trên tập dữ liệu tấn công mạng Kitsune cho thấy thuật toán đề xuất cải thiện đáng kể tốc độ tính toán so với phương pháp trước đó.**

This manuscript is received on December 01, 2022. It is commented on December 03, 2022 and is accepted on December 05, 2022 by the first reviewer. It is commented on December 15, 2022 and is accepted on December 20, 2022 by the second reviewer.

***Keywords*— GPUs; collaborative clustering; fuzzy clustering; high-performance computing.**

***Từ khóa*— GPUs; phân cụm cộng tác; phân cụm mờ; tính toán hiệu năng cao.**

I. Introduction

Today, we are faced with a huge amount of information and data generated every day from many different sources by humans. Data analysis yields a wealth of useful information from sources such as Facebook, Twitter, Google, Wikipedia, and so on, as well as the sensor system found in the majority of electronic devices. However, large data sources will also face demanding requirements in terms of storage memory as well as computing resources. Many techniques and algorithms have been developed to deal with big data problems, but it is difficult to get a universal solution for every problem.

Collaborative fuzzy clustering (CFC) is a tool to find structural similarities and similarities between data patterns located in many distinct regions, that is based on the objective function expansion and fuzzy clustering approach [1]. The idea of this algorithm comes from the fact that the data sets may not be concentrated on one computer but affect each other during the clustering process [2]. At the same time, the fuzzy clustering in this data set has an impact on the clustering in other data sets [3]. There are two features of collaborative fuzzy clustering, one is that detailed information in data sets cannot be exchanged, but only structural information can be exchanged [7]. The second is to consider whether fuzzy clustering in this data set has an impact on clustering in other data sets [8]. Is the information about cluster structure in each dataset useful in clustering the remaining datasets?

Although fuzzy clustering models have many advantages in handling uncertain data, they face difficulties when working on large, multidimensional data sets [10]. Many authors have developed techniques to improve the efficiency of the original model [9]. Li et al. proposed a Fuzzy C-Means (FCM) algorithm based on GPU [4]. The empirical results obtained by Li et al. showed that the proposed parallel FCM on GPU is more efficient than the sequential FCM. Instead of efficiency, they claimed that the proposed method exhibits improvement in the quality of the GPU segmented image [12]. Mahmoud et al. presented a GPU-based FCM for medical images segmentation [5]. Shalom et al. proposed a scalable FCM based on graphic hardware [6]. They modified the sequential FCM algorithm, such that the calculations of the membership and cluster center matrices are not comparable to the sequential one.

However, to the knowledge of the author, there have not been studies to propose collaborative fuzzy clustering models and algorithms to speed up computation on large data sets and overcome the problem of noise and uncertainty in input data. Therefore, in this study, we propose to use a parallel computing model based on high-performance computing units (GPUs) to improve the computational speed of the collaborative fuzzy clustering algorithm. In fact, it is difficult to give an exact concept of large data, in the experimental part, we test the Kitsune Dataset from the UCI machine learning library with more than 27 million data samples. The results show that the proposed method gives much better results than the pre-improvement method.

The paper is organized into 5 sections: Section 1 is the introduction; Section 2 introduces some related knowledge; Section 3 is the proposed method; Section 4 presents some experiments and Section 5 gives the conclusions.

II. BACKGROUND

A. Fuzzy clustering

One of the widely used fuzzy set applications is the FCM algorithm [1]. This algorithm allows each data element to belong to many different clusters according to different membership grades. This algorithm considers MF values based on the distance from each data pattern to cluster centroids [1]. FCM algorithm model is to optimize the objective function:

$$\min\{J_m(U,V,X)=\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^{m}d_{ik}^{2}\} \tag{1}$$

where $U=[\mu_{ik}]_{cxn}$ is a fuzzy MF, $\mathrm{V}=(\mathrm{v}_1,\mathrm{v}_2,...,\mathrm{v}_c)$ is a vector of (unknown) cluster centers, $X=\{\mathrm{x}_k,\mathrm{x}_k\in\mathrm{R}^M,\mathrm{k}=1,...,\mathrm{n}\}$, $d_{ik}=\|v_i-x_k\|$. With the following constraints:

$$m>1; 0\le\mu_{ik}\le 1;$$
$$\sum_{i=1}^{c}\mu_{ik}=1; 1\le i\le c; 1\le k\le n \tag{2}$$

The objective function J_m=(U,V,X) reaches the smallest value when and only if:

$$v_i=\sum_{k=1}^{n}\mu_{ik}^{m}x_k\,/\sum_{k=1}^{n}\mu_{ik}^{m} \tag{3}$$

$$\mu_{ik}=1/\sum_{j=1}^{c}\left(d_{ik}/d_{jk}\right)^{2/(m-1)} \tag{4}$$

Equations (3), (4) can be obtained based on the Lagrange multiplier theorem with the constraints by Equation (2). FCM algorithm will perform iterations according to Equations (3), (4) until the objective function J_m=(U,V,X) reaches the minimum value.

B. Collaborative fuzzy clustering

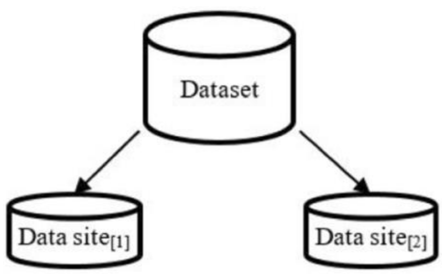

Figure 1. The model of collaboration between data sets

The model of structural information exchange or collaboration between data sets is depicted in Figure 1. Given the data set $X=\{x_1, x_2, ..., x_N\}$ there are P sub-datasets including *D[1], D[2],...,D[P]*, where each sub-dataset containing *N[1], N[2],...,N[P]* data samples in the same attribute space X. In each data set *D[ii]*, we divide into *c* clusters. The clustering results in each data set affect the clustering in the remaining regions, we call this process collaborative fuzzy clustering.

In which the data sets do not directly exchange detailed data, but only share structural information, there is the cluster center vector *v[ii]* (with *ii=1,..,P*). Naturally, when collaboratively clustering results, looking at the overall level of data sets will be better than clustering results based only on local data at each dataset.

The collaborative fuzzy clustering problem whose objective function needs to be optimized is:

$$Q_{[ii]} = \sum_{k=1}^{N[ii]} \sum_{i=1}^{C} u_{ik}^2[ii] d_{ik}^2 + \beta \sum_{jj=1}^{P} \sum_{k=1}^{N[ii]} \sum_{i=1}^{C} (u_{ik} - \tilde{u}_{ik}[ii \mid jj])^2 d_{ik}^2 \quad (5)$$

The first part of the objective function is similar to the FCM algorithmic objective function. The second part of the objective function shows that the optimization in the collaborative process makes the difference between the partitioning matrices decrease.

In the above objective function, u_{ik}[ii] is the matrix that partitions the object k into cluster i in dataset ii. $\tilde{u}_{ik}[ii \mid jj]$ is called the collaborative partitioning matrix of dataset jj onto dataset ii and is calculated by the formula [5]:

$$\tilde{u}_{ik}[ii \mid jj] = \frac{1}{\sum_{j=1}^{c} \left(\frac{| x_k[ii] - v_i[jj] |}{| x_k[ii] - v_j[jj] |} \right)^2} \quad (6)$$

Parameter β represents the degree of cooperation between data sets, the larger the value, the higher the degree of cooperation, and the value $\beta = 0$ represents between data sets that have no cooperation. d_{ik} is the distance from the k^{th} object to the i^{th} cluster center in the same data set.

Using the Lagrange method to optimize the above objective function, the formula for calculating the partition matrix and cluster center is as follows:

$$u_{rs}[ii] = \frac{1}{\sum_{j=1}^{c} \frac{d_{rs}^2}{d_{js}^2}} \left[1 - \sum_{j=1}^{c} \frac{\beta \sum_{jj=1, jj \neq ii}^{P} \tilde{u}_{js}[ii \mid jj]}{(1 + \beta(P-1))} \right] + \frac{\beta \sum_{jj=1, jj \neq ii}^{P} \tilde{u}_{rs}[ii \mid jj]}{(1 + \beta(P-1))} \quad (7)$$

$$v_{rt}[ii] = \frac{\sum_{k=1}^{N[ii]} u_{rk}^2[ii] x_{kt} + \beta \sum_{jj=1, jj \neq ii}^{P} \sum_{k=1}^{N[ii]} (u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^2 x_{kt}}{\sum_{k=1}^{N[ii]} u_{rk}^2[ii] + \beta \sum_{jj=1, jj \neq ii}^{P} \sum_{k=1}^{N[ii]} (u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^2} \quad (8)$$

The partition matrix of the objective function must satisfy the constraint that the total membership of an element in the clusters in the same data set is equal to 1, as follows:

$$U = \{ u_{ik} \in [0,1] \mid \sum_{i=1}^{c} u_{ik}[ii] = 1, \forall k \quad 0 < \sum_{k=1}^{N[ii]} u_{ik}[ii] < N[ii], \forall i \} \quad (9)$$

C. High-performance compute architecture

GPU (Graphic Processing Unit): A microprocessor that specializes in analyzing blocks of image data. The CPU handles graphics and video-related tasks differently than the delicate GPU, which specializes in visual tasks.

The GPU also processes information via multi-threading, parallelism, and memory at high speeds. GPU technology is gradually becoming easier to program, providing more potential for processing acceleration for many programs for a variety of purposes, than conventional processing chips (CPUs).

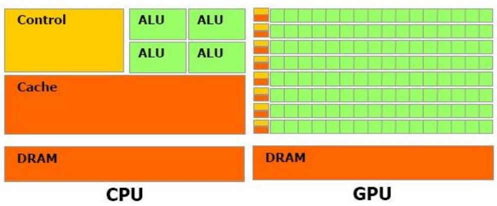

Figure 2. GPU and CPU Architecture

The difference between the data processing capabilities of these two types of chips is this:

GPUs are designed specifically for parallel computing and information processing. Up to 80% of the chip's transistors are used exclusively for data computation, not for receiving data or controlling information flow.

III. PROPOSAL METHOD

A. Collaborative fuzzy clustering based on GPUs

CUDA is the parallel programming model used for NVIDIA GP-GPUs. CUDA can increase performance by harnessing the power of a GPU device. Thousands of threads can be executed concurrently using CUDA on GPUs. The general model of the proposed method is shown in Figure 3. Accordingly, on the CPU, the input data is randomly divided into P data sites. On the GPU, perform local clustering at data sites using the FCM algorithm.

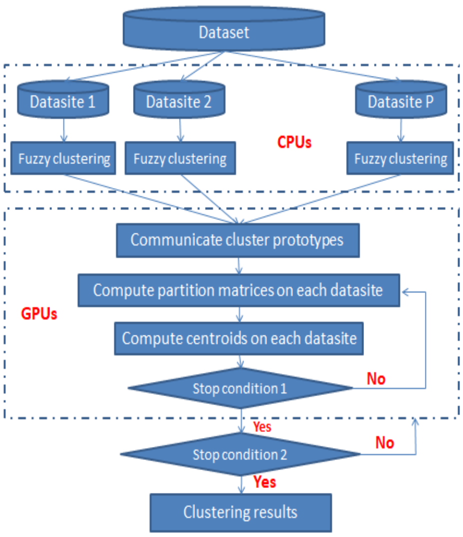

Figure 3. General model of the proposed method

For data set $X = \{x_1, x_2, ..., x_N\}$, N is the number of data samples, n is the number of features of the data. X is divided into data sets D[1], D[2],...,D[P], which contains N[1], N[2],...,N[P] data samples in the same space X contains M attributes. In each data set D[ii] we divide into c clusters.

$$Q_{[ii]} = \sum_{k=1}^{N[ii]} \sum_{i=1}^{c} u_{ik}^m[ii] d_{ik}^2 + \sum_{k=1}^{N[ii]} \sum_{jj=1, jj\neq ii}^{P} \sum_{i=1}^{c} \beta[ii \mid jj] u_{ik}^m[ii] d_{ik}^2[jj] \quad (10)$$

In the objective function (10), ii and jj are the index of the ii data set and jj, $u_{ik}[ii]$ are the partition matrix of the ii dataset. The parameter represents the degree of cooperation of the set jj with the set ii and has the value domain [0, 1], the value $\beta = 0$ represents the data sets without the cooperation and is calculated according to the following formula:

$$\beta[ii \mid jj] = min[1, \frac{J[ii]}{\tilde{J}[ii \mid jj]}] \quad (11)$$

with $\tilde{J}[ii \mid jj] = \sum_{k=1}^{N[ii]} \sum_{j=1}^{C} \tilde{u}_{ik}^2[ii \mid jj] | x_k - v_i[jj]|^2$,

and $\tilde{u}_{ik}[ii \mid jj]$ is the cooperative partitioning matrix of dataset jj on dataset ii and is calculated by the formula:

$$\tilde{u}_{ik}[ii \mid jj] = \frac{1}{\sum_{j=1}^{c} \left(\frac{| x_k[ii] - v_i[jj] |}{| x_k[ii] - v_j[jj] |} \right)^2} \quad (12)$$

where, d_{ik}^2 is the distance between the k^{th} data in the set D[ii] and the center of the i^{th} cluster v_{ij} in this same data set: $d_{ik}^2 = \sum_{j=1}^{M} (x_{kj} - v_{ij})^2$ and $d_{ik}[jj]$ is the distance between the k^{th} data in the set D[ii] and the i^{th} cluster center $v_{ij}[jj]$ in the D[jj]: $d_{ik}^2[jj] = \sum_{j=1}^{M} (x_{kj} - v_{ij}[jj])^2$.

For the convenience of implementation, we build calculation functions on GPU including: distance function d_{ik}, $d_{ik}[jj]$, membership value function u_{rs}, cluster center function v_{rt}, objective function $Q_{[ii]}$, and some other functions $\beta[ii \mid jj]; \tilde{J}[ii \mid jj]$. Input data on data sites is divided into blocks of 16 samples (array). All the arrays are defined in a 1-D pattern. After defining device's memories, all data is transferred from the host to the device, and then the main program loop is started.

The host calls four CUDA kernels one after another to calculate the cluster centers from memberships. The host will determine if the new membership function satisfies the condition as shown in Figure [3]. If the condition is satisfied, finally the cluster center arrays will be transferred back to the host. Defuzzification is performed and the final clustering results are obtained.

The steps to implement the CFC-GPU algorithm are shown in Algorithm 1.

Algorithm 1: The CFC-GPU algorithm

Input: Dataset X, ε, and initialize the parameters m, the maximum number of iterations $T_{\max}$, the number of data site P, the number of items in each data site ii is N[ii], the number of cluster in each data site ii is c[ii], the number of attribute of the data item is n, the data item in each data site X[ii].

Output: Clustering results.

Begin
Phase 1:
1.1 Locally clustering.
1.2 Run FCM algorithm for each data site
Phase 2: Collaboration
2.1 **REPEAT**
2.1.1 t++.
2.1.2 Communicate cluster prototypes from each data site to all others
2.1.3 **For** each data site D[ii]
Compute induced partition matrices
Repeat
+ Compute local partition matrices $u^{(t)}$ by Equation (7).
+ Compute local cluster prototypes $v^{(t)}$ by Equation (8).
Until the objective function is minimized
End for
2.2 **UNTIL** $max(|v^{(t)} - v^{(t-1)}| < \varepsilon$ **OR** $t = T_{max}$
End.

For each phase in the algorithm flowchart, the parameters for each algorithm are selected based on the parameter suggestions in the original paper.

B. Cluster quality evaluation index

Bezdek studied and presented two coefficients, that are Partition Coefficient (PC) [13] and Partition Entropy (PE) [14] to measure the clarity of the fuzzy membership function values.

$$PC = \frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N}u_{rs}^{2}$$
$$PE = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N}u_{rs}\ln u_{rs} \qquad (13)$$

The resulting classification performance is evaluated by determining True Positive Rate (TPR) and False Positive Rate (FPR) defined as follows:

$$TPR = \frac{TP}{TP+FN}; FPR = \frac{TN}{TN+FP} \qquad (14)$$

The accuracy of the classification results is calculated by the formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (15)$$

where TP is the number of correctly classified data and FN is the number of incorrectly misclassified data, FP is the number of incorrectly classified data and TN is the number of correctly misclassified data. The better the algorithm is, the higher the TPR value is, and the smaller the FTR value is encountered.

IV. EXPERIMENTS

In the experiment, we select an Intel Core2 2.66 GHz CPU-based desktop computer with a Windows 10 operating system and C++ programming environment as our platform. NVIDIA graphics card with a memory capacity of 2 GB.

To test the effectiveness of the proposed method, we experimented on a dataset downloaded from the UCI machine learning library [15]. The details of the experimental data set: A cybersecurity dataset containing nine different network attacks on a commercial IP-based surveillance system and an IoT network. The dataset includes reconnaissance, MitM, DoS, and botnet attacks. There are 9 network capture datasets in total, listed below. Viol. is a breach of security (of confidentiality, integrity, and authenticity).

For each attack (network capture) above we provide (1) a CSV of the features used in our paper where each row is a network packet, (2) the corresponding labels [benign, malicious], and (3) the original network capture in truncated pcap format. The dataset consists of 27.170.754 observations with 115 features. We randomly divided the Kitsune Network Attack dataset into 3 subsets (3 data sites). The data is clustered into 10 clusters (one benign packet cluster and nine attack packet clusters).

Experimental evaluation of the accuracy of the proposed algorithm CFC-GPU is compared with the algorithms FCM [1], CFC [7, 8], FCM-GPU [11]. The experimental parameters are set together as follows: Fuzzy parameter m=2, maximum number of iterations $T_{max} = 500$, stopping condition $\epsilon = 10^{-6}$. These parameters were selected based on previously published studies.

TABLE 1. ACCURACY OF CLASSIFICATION RESULTS

Alg.	**Indexes**			
	PC	***PE***	***Acc.***	***Time***
FCM	0.8211	0.4871	0.8706	2h36m28s
CFC	0.8974	0.1571	0.9376	1h24m41s
FCM-GPU	0.8209	0.4895	0.8645	29m32s
CFC-GPU	**0.8973**	**0.1482**	**0.9388**	**16m29s**

Table 1 shows the results of the classification efficiency evaluation according to the CFC, FCM-GPU, and CFC-GPU algorithms. The classification results show that the CFC, CFC-GPU algorithms give the best results on most data sets. The running times of the algorithms FCM, CFC, FCM-GPU, CFC-GPU are shown in Table 1 and Figure 4. It can be seen that the FCM and CFC algorithms running on the CPU run much slower than when running on the GPU.

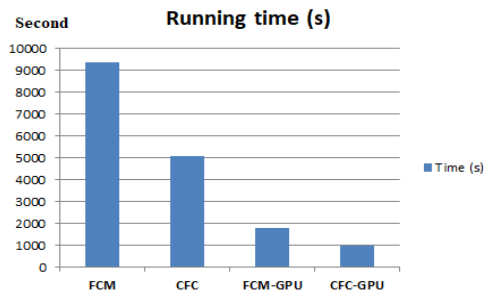

Figure 4. Running times of algorithms

Specifically, the running times on FCM and CFC are *2h36m28s* and *1h24m41s*, respectively. Meanwhile, FCM-GPU and CFC-GPU are *29m32s* and *16m29s*, respectively. As can be seen, when performed on GPU, the CFC and FCM algorithms produce results approximately five times faster than when performed on CPU.

From the experimental results, it can be seen that GPU-based parallel computing on subdatasets will significantly speed up the clustering process according to the traditional sequential method. This is achieved because the parallel computation will perform local clustering (simultaneously) on the subdatasets and there is no communication with each other. After finishing the local clustering process, information will be exchanged between the sub-datasets. Thus, instead of clustering each subset of data in turn before collaborating, parallel clustering will significantly reduce the time to solve the problem.

V. CONCLUSION

The paper proposed to improve the computational speed of the collaborative fuzzy clustering algorithm based on the high-performance computing platform. To evaluate the effectiveness of the proposed algorithm, we tested it on the Kitsune Network Attack large dataset with more than 27 million data samples. Experimental results show that the proposed CFC-GPU algorithm running on the GPU gives computational results many times faster than the CFC algorithm running on the CPU. This result shows the potential of applying for integrated graphics cards on personal computers to increase the computational speed of algorithms on large data sets.

In the future, we will experiment with other types of data and develop semi-supervised and supervised techniques for more powerful algorithms, such as fuzzy deep learning.

REFERENCES

[1] James C.Bezdek, Robert Ehrlich, William Full. FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, Volume 10, Issues 2–3, 1984, Pages 191-203. https://doi.org/10.1016/0098-3004(84)90020-7.

[2] K.R. Zalik, An efficient k-means clustering algorithm, Pattern Recognition Letters Vol. 29,

pp 1385 - 1391, 2008. https://doi.org/10.1016/j.patrec.2008.02.014.

[3] L. Zhu, F.L. Chung, S. Wang, Generalized Fuzzy C-Means Clustering Algorithm With Improved Fuzzy Partitions, IEEE Trans. On Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 39(3), pp 578-591, 2009. 10.1109/TSMCB.2008.2004818.

[4] Haiyang Li, Zhaofeng Yang, Hongzhou He. An Improved Image Segmentation Algorithm Based on GPU Parallel Computing. Journal of Software, Vol 9, No 8 (2014), 1985-1990, 2014. 10.4304/jsw.9.8.1985-1990.

[5] Mahmoud Al-Ayyoub, AnsamM Abu-Dalo, Yaser Jararweh, Moath Jarrah, Mohammad Al Sa'd, A GPU-based implementations of the fuzzy C-means algorithms for medical image segmentation. The Journal of Supercomputing, 1-14, Springer US, 2015. https://doi.org/10.1007/s11227-015-1431-y.

[6] Shalom, S.A.A., Dash, M. and Tue, M. Graphics hardware based efficient and scalable fuzzy c-means clustering. 7th Australasian data mining conference, volume 87, 179-186, 2008. https://dl.acm.org/doi/10.5555/2449288.2449316.

[7] Witold Pedrycz, Collaborative fuzzy clustering, Pattern Recognition Letters 23, pp1675–1686, 2002. https://doi.org/10.1016/S0167-8655(02)00130-7.

[8] Witold Pedrycz, Collaborative clustering with the use of Fuzzy C-Means and its quantification, Fuzzy Sets and Systems, pp 2399 – 2427, 2008. https://doi.org/10.1016/j.fss.2007.12.030.

[9] Trong Hop Dang, Dinh Sinh Mai, Long Thanh Ngo. Multiple kernel collaborative fuzzy clustering algorithm with weighted super-pixels for satellite image land-cover classification, Engineering Applications of Artificial Intelligence, Vol. 85, 2019, 85-98. https://doi.org/10.1016/j.engappai.2019.05.004.

[10] N. Aitali, B. Cherradi, A. El Abbassi, O. Bouattane and M. Youssfi, "GPU based implementation of spatial fuzzy c-means algorithm for image segmentation," 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), 2016, pp. 460-464. 10.1109/CIST.2016.7805092.

[11] Mishal Almazrooie, Mogana Vadiveloo, Rosni Abdullah. GPU-Based Fuzzy C-Means Clustering Algorithm for Image Segmentation, 2016, https://arxiv.org/abs/1601.00072.

[12] L. T. Ngo, D. S. Mai and M. U. Nguyen, "GPU-based acceleration of interval type-2 fuzzy c-means clustering for satellite imagery land-cover classification," 12th International Conference on Intelligent Systems Design and Applications (ISDA), 2012, pp. 992-997, https://doi.org/10.1109/ISDA.2012.6416674.

[13] J. C. Bezdek, Numerical taxonomy with fuzzy sets, Journal of Mathematical Biology, vol. 1, no. 1, pp. 57–71, 1974. https://doi.org/10.1007/BF02339490.

[14] J. C. Bezdek, Cluster validity with fuzzy sets, Journal of Cybernetics, vol. 3, no. 3, pp. 58–73, 1974.https://doi.org/10.1080/01969727308546047.

[15] https://archive.ics.uci.edu/ml/datasets.php.

ABOUT THE AUTHOR

Dang Trong Hop

Workplace: Hanoi University of Industry.

Email: hopdt@haui.edu.vn

Education: PhD in mathematical foundations for informatics.

Recent research direction: Computational intelligence; Fuzzy system; Image processing; Clustering.

Mai Dinh Sinh

Workplace: Le Quy Don Technical University.

Email: maidinhsinh@lqdtu.edu.vn

Education: PhD in mathematical foundations for informatics.

Recent research direction: Fuzzy logic; Soft computing; Pattern recognition; Geo-Informatics.

Do Viet Duc

Workplace: Le Quy Don Technical University.

Email: minhducmta@gmail.com

Education: PhD Student

Recent research direction: Fuzzy logic; Artificial intelligence; Big data.

Ngo Thanh Long

Workplace: Le Quy Don Technical University.

Email: ngotlong@lqdtu.edu.vn

Education: Associate Professor of Information Technology.

Recent research direction: Fuzzy logic; Artificial intelligence; Pattern recognition; Robotics.