Received May 26, 2021, accepted June 6, 2021, date of publication June 10, 2021, date of current version June 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3088295

Adaptive Landmark-Based Spectral Clustering for Big Datasets

AHMED R. ABAS[1], MOHAMED G. MAHDY[1], AND TAREK M. MAHMOUD[2]
[1]Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt
[2]Faculty of Computer and Artificial Intelligence, University of Sadat City, Sadat 32897, Egypt

Corresponding author: Mohamed G. Mahdy (mohamedgresha@hotmail.com)

ABSTRACT Clustering has emerged as an effective tool for the processing and assessment of the vast data generated by modern applications; its primary aim is to classify data into clusters in which the items are grouped into a given category. However, various challenges, such as volume, velocity, and variety, occur during the clustering of big data. Different algorithms have been proposed to enhance the performance of clustering. The landmark-based spectral clustering (LSC) technique has been proven to be efficient in clustering big datasets. In this study, an algorithm called adaptive landmark-based spectral clustering (ALSC) is proposed for clustering big datasets. The proposed algorithm comprises the adaptive competitive learning neural network (ACLNN) algorithm, which can be efficiently used to determine the number of clusters and the LSC technique. The ACLNN algorithm can also be used with small datasets. Thus, in our implementation, the original big dataset is split into N small sub datasets, which run in parallel by N copies of the ACLNN algorithm. To evaluate the performance of the proposed algorithm, two distinctive datasets, namely, Fashion-MNIST and United States Postal Service are used. The experiments show that the proposed ALSC algorithm produces high clustering accuracy with the identification of the number of clusters. Results reveal that the normalized mutual information and adjusted Rand index of the proposed algorithm outperform state-of-the-art models.

INDEX TERMS Adaptive competitive learning neural network, big datasets, clustering, landmark-based spectral clustering, parallel processing.

I. INTRODUCTION

The massive volume of data produced every day in recent years mostly comprises the data generated from satellites, social media, smart devices, sensors, business transactions, and computer simulations. These data generate invaluable information and insight into decision support, forecasts, intensive data research, and business intelligence. Walmart stores approximately 2.5 petabytes of data, whereas Facebook stores approximately 30 petabytes. Such vast data are referred to as big data; mining is needed to retrieve the required information [1], [2]. Structured, semistructured, and unstructured data are the three general categories of data [3]. Majority of data is unstructured and cannot be processed using conventional approaches. Volume, velocity, and variety are three distinct parameters that characterize big data [4]. The amount of data in a file or database is simply referred to as volume. In most networks, the volume of data is stored at an exponential scale. Data extraction becomes more complex as the volume of data increases, and data backup exacerbates these issues. The speed at which data is exchanged, captured, and produced is referred to as velocity. Another clustering challenge faced by data scientists is the pace at which data is produced. This issue is not just regarding the amount of data in a network; if networks produce new data at exponential rates, extracting it in real-time becomes more challenging. The term ''variety of data'' refers to the different types of data. Clustered data can be processed in various formats, making precise comparisons impossible. Most data are contained in structured formats, whereas others are unstructured wholly. Healthcare, biology, bioinformatics, insurance, banking, marketing, telecommunications, earthquake studies, city planning, online document classification, and transportation services are some of the industries that use big data.

Clustering is an example of an unstructured machine learning mechanism used for data analysis. It divides data into

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

subsets that are identical in content [5]. Traditional clustering algorithms classify data structures based on their resemblance to a set of threshold parameters. However, the outcomes of these algorithms are based on a nondeterministic predefined threshold parameter [6]. Market research, pattern recognition, data analysis, and image processing are some examples of the applications where clustering analysis is used.

The competitive learning neural network (CLNN) is used for clustering analysis [1], [7]–[9]. It comprises an input layer and a single competitive learning output layer. Examples of the CLNN are self-organizing map [10]–[12] and learning vector quantization [10], [11]. In the CLNN, the output neurons are completely bound to each input node [13]–[15]. Several algorithms have been proposed to determine the number of neurons in the output layer of the CLNN; this number is considered to be the optimal number of clusters in the input dataset.

The adaptive competitive learning neural network (ACLNN) algorithm was proposed to decide the optimum number of clusters and clustering of a small input dataset. The ACLNN algorithm determines the optimal number of output neurons using the adaptive competitive learning (ACL) criteria [16], [17]. The ACL criteria assume that the best cluster configuration comprises balanced, dense, and well-separated clusters with the fewest parameters to calculate. The ACLNN algorithm fails to maximize the hardware resources offered by modern multicore processors. Parallel processing utilizes more hardware resources, thus eliminating the time taken by it to complete a job [18], [19].

In general, parallel processing uses graphic processing unit (GPU) cores instead of central processing unit (CPU) cores because of GPU Cores large number [19]. The feedforward neural network is the only neural network whose training function is parallelized to be applied on GPU cores [20], [21]. By contrast, the training function of the CLNN is sequential and cannot be applied to GPU cores [20], [21].

The landmark-based spectral clustering (LSC) technique selects a subset of data points (p) as landmarks and represents the remaining data points as linear combinations of these landmarks [22]. The Landmark-based representation is used to compute the spectral embedding of data.

In this study, an algorithm for determining the optimal number of clusters (k_{opt}) is proposed. This number (k_{opt}) is used in the clustering of an input big dataset. This algorithm is referred to as the ALSC algorithm. The proposed algorithm comprises three phases and uses parallel processing to speed up its performance. The main advantages of the proposed algorithm are determining the optimal number of clusters and producing high clustering accuracy with the identification of the number of clusters.

The main contributions of this work are as follows:

- proposing the ALSC algorithm to determine the optimal number of clusters (K_{opt}),
- using this number (K_{opt}) for the clustering input big dataset,
- using multiple samples of the input dataset with parallel processing to reduce the running time of the ALSC algorithm and determine the optimal number of clusters, and
- comparing the proposed algorithm with state-of-the-art models [23]–[29].

This paper is structured as follows: Related work is presented in Section 2; the proposed ALSC algorithm is presented in Section 3; performance evaluation of the ALSC algorithm is presented in Section 4; the results are discussed in Section 5; lastly, the study is concluded in Section 6.

II. RELATED WORK

For clustering, fuzzy c-means (FCM) [23] uses a membership matrix and an updated rule. Each data object is assigned to one cluster using K-means [24]. FCM achieved a performance of 36.44% and 51.59% in terms of adjusted Rand index (ARI) and normalized mutual information (NMI), respectively, in the Fashion-MNIST data set and achieved a performance of 53.93% and 62% in terms of ARI and NMI, respectively, in the United States Postal Service (USPS) dataset. The cornerstone of spectral embedded clustering (SEC) is a foundation of multiple learning [25]. SEC achieved a performance of 36.39% and 51.64% in terms of ARI and NMI, respectively, in the Fashion-MNIST data set and achieved a performance of 54.5% and 62.56% in terms of ARI and NMI, respectively, in the USPS data set. A more sophisticated variant of the K-means algorithm, i.e., minibatch K-means (MBKM) [24], reduces computational complexity using a minibatch. MBKM achieved a performance of 34.5% and 50% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and 51% and 59.93% in terms of ARI and NMI, respectively, in the USPS dataset. Deep embedding clustering (DEC) [26] is a deep learning-based algorithm that forgoes the decoder and uses a specially built distribution. DEC achieved a performance of 45.71% and 62.83% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and achieved a performance of 66.22% and 73.52% in terms of ARI and NMI, respectively, in the USPS dataset. Improved deep embedded clustering (IDEC) [30] is built on the foundations of deep clustering and a carefully designed distribution. IDEC achieved a performance of 44% and 60.13% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and achieved a performance of 67.91% and 75.95% in terms of ARI and NMI, respectively, in the USPS dataset. A reconstruction mechanism is used to regularize the autoencoder. Deep fuzzy c-means (DFCM) [27] is a deep learning-based algorithm that combines deep learning and Fuzzy C-Mean. DFCM achieved a performance of 48.65% and 64.54% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and 68.15% and 76.36% in terms of ARI and NMI, respectively, in the USPS dataset. To improve the clustering process, the encoder–decoder convolutional neural network (CNN) model and the FCM technique are combined in

the improvised fuzzy c-means (IFCM) [28]. IFCM achieved a performance of 54.19% and 67.35% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and 85% and 89% in terms of ARI and NMI, respectively, in the USPS dataset. The fuzzy compactness and separation (FCS) clustering algorithm is an efficient method that estimates the fuzzy memberships of data using within-and between-cluster distances. The deep normalized fuzzy compactness and separation (DNFCS) clustering method was established in response to the superiority of the FCS algorithm. The graph regularized deep normalized fuzzy compactness and separation fuzzy clustering (GrDNFCS) is focused on autoencoder-based data reconstruction, considering between-cluster separation, and affinity regularization using pseudolabels [29]. GrDNFCS achieved a performance of 50.28% and 66% in terms of ARI and NMI, respectively, in the Fashion-MNIST dataset and 69% and 77.61% in terms of ARI and NMI, respectively, in the USPS dataset. The performance of the previous algorithms is poorer than that of the proposed algorithm. The NMI and ARI performance of each dataset and the technique used in each algorithm are summarized in Table 1.

The Visual Assessment of (Clustering) Tendency (VAT) clusters data use a dissimilarity matrix. However, this algorithm is biased toward producing large clusters. Moshtaghi *et al.* [31] clustered data using anomaly detection and used dendrograms for visual representation. This algorithm has been applied to several taxonomy applications. Wilbik *et al.* [32] proposed a single linkage-based clustering for segmenting time series-based data. K-means clusters data efficiently. The advantage of using K-means is its applicability and simplicity in several fields. As a batch-based algorithm, K-means has several limitations, such as poor initialization. Recently, deep learning has produced satisfactory results in big data clustering [5], [33]. Several rough or fuzzy-based approaches have been proposed for handling the uncertainty in clustering data. Deng *et al.* [5] proposed a hierarchical approach integrating neural networks and fuzzy logic for robust clustering. Rajesh and Malar [34] proposed an approach based on neural networks and rough set theory for clustering data. However, this approach requires substantial amount of data for training, thus requiring long running time.

Semisupervized clustering algorithms have been introduced to handle clustering data and are used for online learning and large datasets [33], [35]–[37]. However, these algorithms fail to achieve high accuracy for noisy and uncertain datasets.

The ACLNN algorithm was proposed for determining the optimal number of clusters and clustering of input small datasets. The ACLNN algorithm uses the ACL criterion for determining the optimal number of output neurons [16], [17]. The ACL criterion is based on the theory stating that the best cluster structure is composed of balanced, dense, and well-separated clusters that have the least number of parameters to be calculated. The ACLNN algorithm has been proven to be efficient in identifying the optimal number of clusters and clustering small datasets [16], [17]. Because of its sequential running nature, the ACLNN algorithm has high time complexity and thus consumes a large running time, especially when used with big datasets. In addition, the ACLNN algorithm does not appropriately use hardware resources provided by modern multicore processors. Parallel processing allows additional hardware resources to be used and therefore decreases the running time [18], [19]. Parallel processing requires the algorithm to be parallelized and divided into numerous independent tasks to avoid problems such as deadlock, starvation, and race condition [18], [20].

Several clustering algorithms, such as the K-means [24] and generative mixture models [38], produce clusters that have a convex geometric shape. Furthermore, these algorithms use iterative functions to learn their parameters, which seek out local minima. Therefore, multiple restarts are required to find good solutions [39].

By contrast, spectral clustering (SC) algorithms can produce clusters with more complex shapes, such as intertwined spirals or other nonlinear shapes, because SC algorithms do not impose specific shapes on clusters [40].

SC algorithms use the eigenvectors of the adjacency matrix to determine clusters. These algorithms are used in image segmentation [41], text mining, speech processing, and data analysis and clustering [40]. A survey conducted on SC algorithms can be found in [42].

The LSC technique selects some data points (p) as landmarks and represents the rest of the data points as linear combinations of these landmarks [22]. The spectral embedding of the data is computed using landmark-based representation. This algorithm linearly scales with the problem size [22] and is motivated by the recent progress in sparse coding and scalable semisupervized learning. The following paragraphs present some of these efforts, which are summarized in Table 1.

III. PROPOSED ALSC ALGORITHM

The proposed ALSC algorithm comprises three phases. The second phase uses the ACLNN algorithm and parallel processing to determine the optimal number of clusters. The third phase uses LSC for clustering the input big dataset. Algorithm 1 and Figure 1 show the steps and description of the proposed ALSC algorithm.

A. PREPROCESSING PHASE (PHASE 1)

All features of the input dataset are normalized from 0 to 1. Then, the dataset is divided into N subdatasets of equal sizes while preserving the dataset's characteristics.

B. PHASE 2

The created N subdatasets are used as input to N separate copies, which are working in parallel (multiple CPU cores), of the ACLNN algorithm. The number of clusters is determined during this step. In our implementation, the minimum number of clusters (K_{opt}) is selected to be the optimal number of clusters produced as an output of Phase 1 and denoted by K_{opt}.

TABLE 1. Summary of the related work and illustration for each study: algorithm name, dataset name, dataset size, # of classes, clustering algorithms that were used, and performance metrics values.

Authors	Algorithm Name	Dataset	Dataset Size	# of classes	Clustering Algorithms Used	Performance
Venkat et al. [23]	Fuzzy c-mean	Fashion MNIST	60000	10	Combination of fuzzy logic and c-mean	ARI = 36.44 % NMI = 51.59 %
		USPS	9298	10		ARI = 53.93 % NMI = 62 %
Sculley et al. [24]	K-means	Fashion MNIST	60000	10	• K-means	ARI = 36.39 % NMI =51.64 %
		USPS	9298	10		ARI = 54.5 % NMI = 62.56 %
Nie et al. [25]	SEC	Fashion MNIST	60000	10	• Spectral clustering	ARI = 38.44 % NMI = 55.8 %
		USPS	9298	10		ARI = 49.36 % NMI = 64.88 %
Sculley et al. [24]	MBKM	Fashion MNIST	60000	10	Combination of K-means and minibatch	ARI = 34.5 % NMI = 50.03 %
		USPS	9298	10		ARI = 51.05 % NMI = 59.93 %
Girshick et al. [26]	DEC	Fashion MNIST	60000	10	• Deep learning	ARI = 45.71 % NMI = 62.83 %
		USPS	9298	10		ARI = 66.22 % NMI = 73.52 %
Guo et al. [30]	IDEC	Fashion MNIST	60000	10	Combination of deep learning and local structure preservation	ARI = 44.09 % NMI = 60.13 %
		USPS	9298	10		ARI = 67.91 % NMI = 75.95 %
Feng et al. [27]	DFCM	Fashion MNIST	60000	10	Estimates the fuzzy memberships of data using within-and between-cluster distances	ARI = 48.65 % NMI = 64.54 %
		USPS	9298	10		ARI = 68.15 % NMI = 76.36 %
Rayala et al. [28]	Improvised FCM	Fashion MNIST	60000	10	Combination of fuzzy logic, c-means, and encoder decoder CNN	ARI = 54.19 % NMI = 67.35 %
		USPS	9298	10		ARI = 85.01 % NMI = 89.01 %
Feng et al. [29]	GrDNFCS	Fashion MNIST	60000	10	Combination of autoencoder-based data reconstruction, between-cluster separation, and affinity regularization using pseudolabels	ARI = 50.28 % NMI = 66.09 %
		USPS	9298	10		ARI = 69.03 % NMI = 77.61 %

C. PHASE 3

The input of this phase is the original big dataset and the minimum number of clusters produced by Phase 2. These data are used as input to the LSC technique, which produces the clustering results for every data object of the input data, and the corresponding NMI, and ARI.

IV. EXPERIMENTS AND RESULTS

The performance of the proposed algorithm is evaluated and compared with that of state-of-the-art algorithms using real datasets. All experiments are conducted on a DELL G5 15 laptop that has an Intel i7 processor, 2.20 GHz CPU, 6 GB Nvidia graphics, and 16 GB RAM. Windows 10 is the operating system used. The Neural Network Toolbox and Deep Learning Toolbox [10], [11] are used in the MATLAB R2018a platform. In Phase 2, the processor starts a pool on the local machine with 12 workers to run the ACLNN in parallel.

Section A describes the datasets used in experiments. The state-of-the-art algorithms used in the experiments are provided in Section B. Section C describes the measures used to evaluate the clustering performance of all algorithms.

A. DATASETS

Two standard real datasets, namely, the Fashion-MNIST and USPS datasets, are used in the experiments.

The Fashion-MNIST dataset, which comprises a training set of 60,000 images and a test set of 10,000 images, is one of the most common fashion clothing image datasets. Each image is a 28 $\times$ 28 pixel grayscale image with a total resolution of 784 pixels and a label from one of ten clusters.

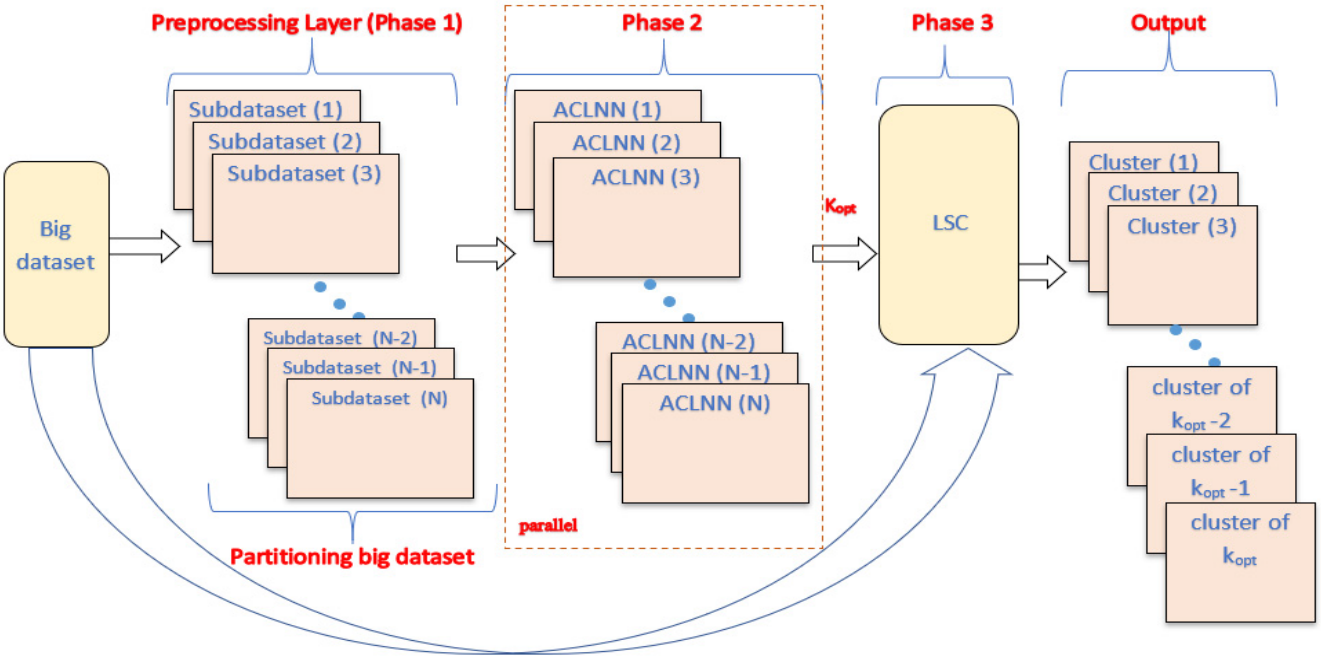


FIGURE 1. Description of the proposed ALSC algorithm.

Algorithm 1 ALSC Algorithm

Input: Big dataset X

1. Normalize the values of each data feature to range from 0 to 1.
2. Divide the input big dataset into N small sub-datasets of equal sizes. (Phase 1)
3. Open parallel pool with 12 workers. (Phase 2)
4. Apply steps 5, 6, and 7 in parallel workers on each subdataset obtained in Step 2.
5. Use one of the subdatasets determined in Step 1.
6. (K) = ACLNN (Kmax =16, subdataset) to obtain the best CLNN model that has the minimum value of the ACL criterion.
7. Save results.
8. Close the parallel pool.
9. From the stored results in Step 7, select the minimum K that is considered the optimal number of clusters (K_{opt}).
10. (clustering id results) = LSC (K_{opt}, big dataset). (Phase3)
11. Calculate the NMI, and ARI according to clustering id results for every data object in the input dataset.

Output: K_{opt}, clustering id results, NMI, and ARI.

Each pixel has a single pixel value that indicates its lightness or darkness, with higher numbers representing darker pixels. An integer between 0 and 255 is used as the pixel number. Assuming x has been decomposed as $x = I * 28 + j$, where I and j are integers between 0 and 27, the aim is to find a pixel in the image. A 28 × 28 matrix includes the pixels on row I and column j. Each row contains a different image. In all training and testing splits, all groups are equally represented [43]. Figure 2 shows samples of the Fashion-MNIST image dataset.

The USPS digit dataset is one of the most commonly used datasets for recognizing handwritten digits. The dataset comprises numeric data extracted from the USPS's scanning of handwritten digits from envelopes. Here, the photos here been disassembled and size have been normalized from the original binary scanned digits, which are varying in size and orientation. It has a total of 9,298 handwritten digit images, divided into 7,291 training and 2,007 test images. Each image is a 16 × 16 grayscale image with a total resolution of 256 pixels and a label from one of ten groups. Each pixel has a single pixel value that indicates its lightness or darkness, with higher numbers representing darker pixels. This pixel value is an integer ranging from 0 to 255. Assuming x has been decomposed x as $x = I * 16 + j$, where I and j are integers between 0 and 15, the aim is to find a pixel in the image. In a 16 × 16 matrix, the pixel is in row I and column j. Each row contains a unique image [44]. Figure 3 shows samples of the USPS image dataset.

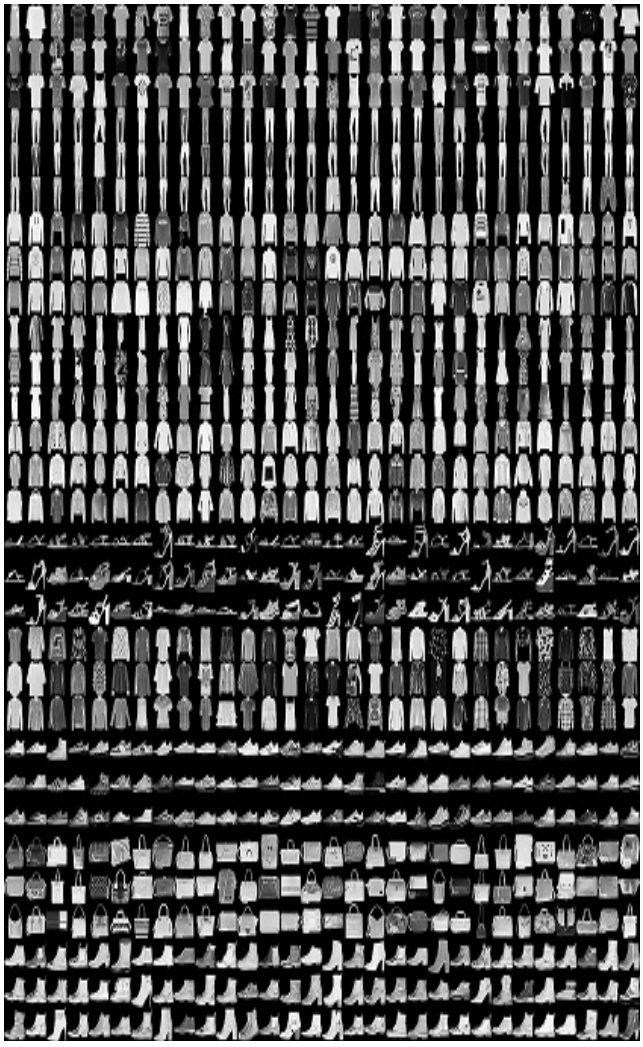


FIGURE 2. Samples from the Fashion-MNIST image dataset.

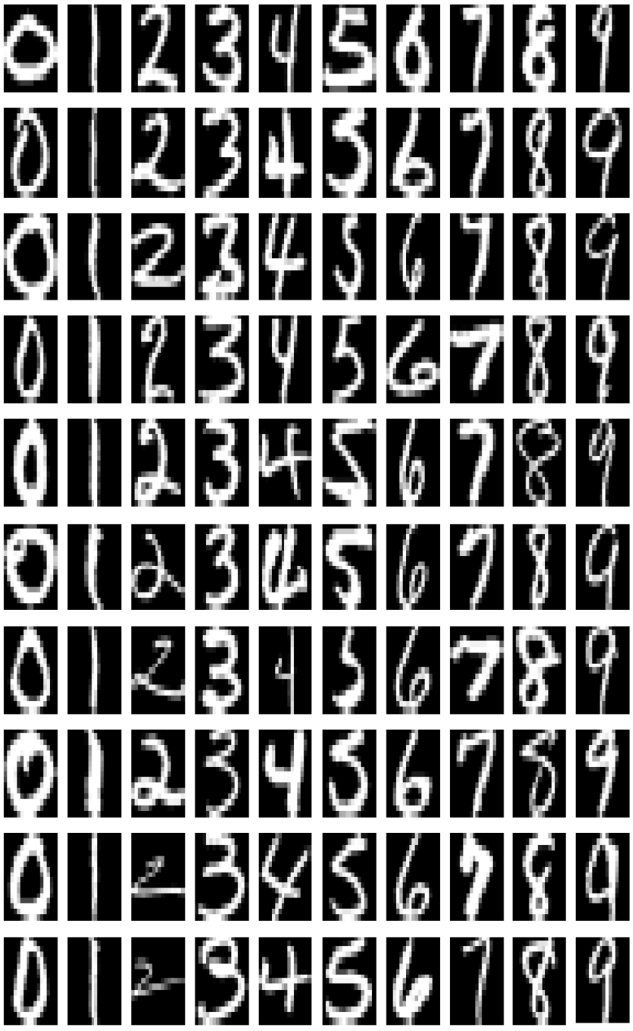


FIGURE 3. Samples from the USPS image dataset.

B. STATE-OF-THE-ART ALGORITHMS

This section describes some state-of-the-art algorithms. FCM [23] uses the matrix of membership and updates rules for clustering. K-means [24] allocates every data object into one cluster. SEC [25] is based on manifold learning. MBKM [24] is a more advanced version of the K-means algorithm and uses a minibatch to minimize computational complexity. DEC [26] is based on deep learning, abandons the decoder part, and has a particularly designed distribution. Deep clustering and a precisely crafted distribution are the pillars of IDEC [30]. The autoencoder is regularized via the reconstruction mechanism. To improve the clustering process, improvised FCM [28] uses the encoder-decoder CNN model and FCM technique. The FCS method is an efficient fuzzy clustering method that estimates the fuzzy memberships of data using within and between-cluster distances. DNFCS is established in response to the superiority of the FCS algorithm, whereas GrDNFCS is focused on autoencoder-based data reconstruction, consideration of between-cluster separation, and affinity regularization using pseudolabels [29].

C. PERFORMANCE MATRICES

1) NORMALIZED MUTUAL INFORMATION

The measure of mutual dependency between two variables is known as mutual information. NMI is a metric for assessing clustering algorithm efficiency [45]. The NMI value ranges from 0 (perfect mismatch) to 1 (perfect match). It indicates the amount of information the clusters have produced.

$$I\,(X;Y) = \sum_{i=1}^{m} \sum_{j=1}^{k} P_{ij} \log_2 (P_{ij}/P_i P_j), \tag{1}$$

where P_{ij} is the probability that a member of cluster j belongs to class i, P_i is the probability of class i, and P_j is the probability of cluster j.

$$NMI\,(X;Y) = \frac{I(X;Y)}{\sqrt{H\,(X)\,H(Y)}}, \tag{2}$$

where H(X) and H(Y) denote the entropy of X and Y.

2) ADJUSTED RAND INDEX

The Rand index is a calculation of the closeness of two separate data clusters, and it ranges from 0 to 1. The higher the

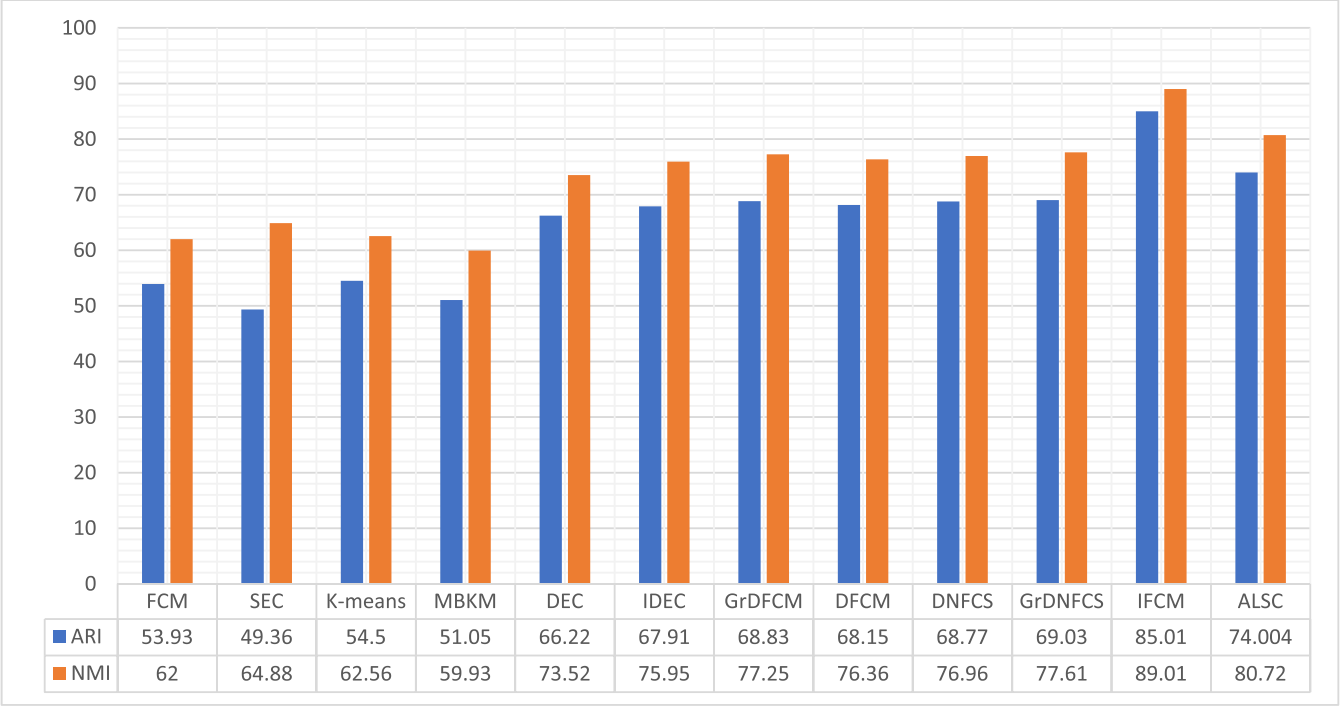


FIGURE 4. Performance evaluation of the compared algorithms using the USPS dataset.

TABLE 2. Number of clusters according to the number of subdatasets using the Fashion-MNIST dataset.

# of subdataset	K	Time (seconds)
1	15	3.118e+03
2	15	1325.64
3	14	867.459
4	15	713.511
5	14	625.343
6	15	591.892
7	16	574.169
8	14	571.567
9	15	476.236
10	15	446.932
11	**11**	**388.742**
12	15	289.812
13	14	190.979
14	15	177.409

ARI value, the more accurate is that the received clustering model [28].

$$\begin{aligned} &\textit{Average Rand index} \\ &\quad = (\textit{Rand\_Index} - \textit{true negative}) \\ &\qquad /(\max(\textit{Rand\_Index}) - E(\textit{Rand\_Index})) - 1 \end{aligned} \quad (3)$$

In addition to Tables 2 and 3 show the different sizes of subdatasets created from a big dataset using the USPS and

TABLE 3. Number of clusters according to the number of subdatasets using the USPS dataset.

# of subdataset	K	Time (seconds)
1	12	3.9042e + 02
2	12	143.23
3	12	100.017
4	11	79.88
5	11	87.287
6	**10**	**69.996**
7	12	67.628
8	12	65.978

Fashion-MNIST datasets, as well as the effect of subdataset size on the determination of the number of clusters (K). Figures 4 and 5 show the performance evaluation of the compared algorithms using the USPS and Fashion-MNIST datasets, respectively. Table 4 shows the performance evaluation of the proposed ALSC algorithm in determining the minimum number and the standard deviation of clusters, the maximum NMI and ARI, and the minimum time in seconds for both datasets.

V. DISCUSSION OF RESULTS

Table 2 shows the different sizes of subdatasets created from the Fashion-MNIST dataset and the number of clusters (K) determined by Phase 2 in the proposed ALSC algorithm. We observe that the best number of subdatasets created from

TABLE 4. Performance evaluation of the proposed ALSC algorithm using both datasets.

Data	k_{opt}		NMI		ARI		Time (seconds)	
	MIN	STD	MAX	STD	MAX	STD	MIN	STD
USPS dataset	10	2.044	80.72	0.0059	74.004	0.2196	42.985	0.1497
Fashion-MNIST dataset	11	1.5239	74.35	0.0068	65.9554	1.3605	64.1175	0.5949

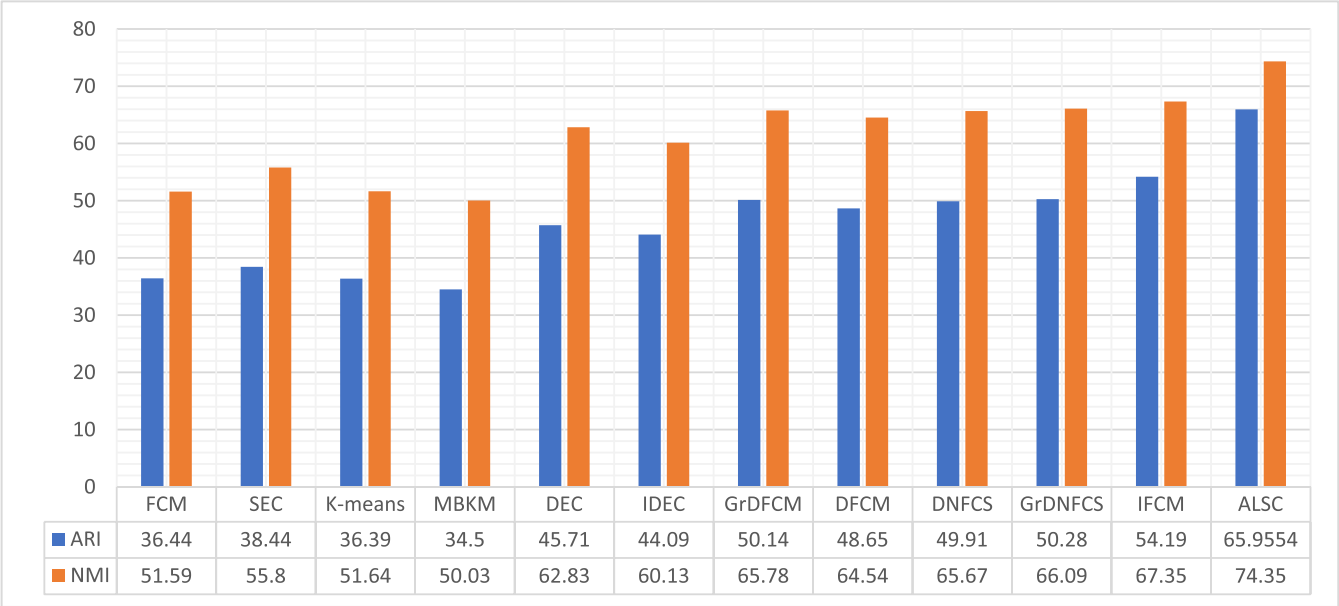


FIGURE 5. Performance evaluation of the compared algorithms using the Fashion-MNIST dataset.

this dataset is 11, which are used as input data to 11 separate copies of the ACLNN algorithm. The minimum number of clusters (K_{opt}) produced using the 11 ACLNN algorithms is 11 clusters, which is approximately equal to the exact number of clusters.

Table 3 shows the different sizes of subdatasets created from the USPS dataset and the number of clusters (K) determined by Phase 1 in the proposed ALSC algorithm. We observe that the best number of subdatasets created from this dataset is six, which are used as input data to six separate copies of the ACLNN algorithm. The minimum number of clusters (K_{opt}) produced using the six ACLNN algorithms is 10, which is equal to the exact number of clusters.

Figure 4 shows that the proposed ALSC algorithm achieved 74.004% for the ARI and 80.72% for the NMI using the USPS dataset. In addition, Figure 5 shows that the proposed ALSC algorithm achieved an ARI value of 65.96% and an NMI value of 74.35% using the Fashion-MNIST dataset. These results show that in terms of clustering performance, the proposed ALSC algorithm outperforms all the state-of-the-art algorithms used in the comparison study with the Fashion-MNIST database. In the USPS dataset, the proposed ALSC algorithm outperforms all the state-of-the-art algorithms used in the comparison study, except the improvised FCM.

Table 4 shows that the proposed ALSC algorithm can determine the number of clusters that is approximately equal to the exact number of clusters in both datasets used in the comparison study; by contrast, the other algorithms require this number to be given as an input parameter.

The reason of the superiority of the proposed ALSC algorithm in clustering performance is its ability to perform SC to deal with clusters having complex shapes, such as intertwined spirals or other nonlinear shapes. In addition, this algorithm does not impose any specific shape on the data clusters. Lastly, the proposed ALSC algorithm efficiently determines the number of clusters based on the ACL criterion. This criterion is efficient in selecting the number of balanced, dense, and well-separated clusters and has the least number of parameters to be calculated.

VI. CONCLUSION

In this study, the ALSC algorithm is proposed for clustering big datasets while determining the optimal number of clusters. In the ALSC algorithm, the original big dataset is split into N small subdatasets, which run in parallel by N copies of the ACLNN algorithm, to determine the optimal number of clusters of this dataset. Then, it uses this number to determine the clustering results for every data object in the input dataset using the LSC technique. A performance

evaluation study is conducted, and results show that the proposed ALSC algorithm produces high clustering accuracy with the identification of the number of clusters.

In the future, the proposed ALSC algorithm for high-dimensional data could be improved using more priors. To discover a better latent representation of data, novel deep autoencoder variants, such as denoising autoencoders (DAE), contractive autoencoders (CAE), and Relation autoencoders (RAE), must be considered. We plan investigating the online training of an ALSC algorithm for big data by combining incremental and reinforcement learning, as inspired by online clustering.

REFERENCES

[1] W. Li, Y. Gu, D. Yin, T. Xia, and J. Wang, "Research on the community number evolution model of public opinion based on stochastic competitive learning," *IEEE Access*, vol. 8, pp. 46267–46277, 2020, doi: 10.1109/ACCESS.2020.2978522.

[2] M. Elkano, J. A. Sanz, E. Barrenechea, H. Bustince, and M. Galar, "CFM-BD: A distributed rule induction algorithm for building compact fuzzy models in big data classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 1, pp. 163–177, Jan. 2020, doi: 10.1109/TFUZZ.2019.2900856.

[3] D. Dietrich, B. Heller, and B. Yang, *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. Hoboken, NJ, USA: Wiley, 2015.

[4] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, "A hybrid approach to clustering in big data," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2372–2385, Oct. 2016, doi: 10.1109/TCYB.2015.2477416.

[5] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 1006–1012, Aug. 2017, doi: 10.1109/TFUZZ.2016.2574915.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.

[7] T. C. Silva and L. Zhao, "Detecting and preventing error propagation via competitive learning," *Neural Netw.*, vol. 41, pp. 70–84, May 2013, doi: 10.1016/j.neunet.2012.11.001.

[8] L. Qu, Z. Zhao, L. Wang, and Y. Wang, "Efficient and hardware-friendly methods to implement competitive learning for spiking neural networks," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13479–13490, Sep. 2020, doi: 10.1007/s00521-020-04755-4.

[9] T. Li, G. Kou, Y. Peng, and Y. Shi, "Classifying with adaptive hyperspheres: An incremental classifier based on competitive learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 4, pp. 1218–1229, Apr. 2020, doi: 10.1109/TSMC.2017.2761360.

[10] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox TM User's Guide R2017b*. Natick, MA, USA: Mathworks, 2017.

[11] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep Learning Toolbox User's Guide*. Herborn, MA, USA: Mathworks, 2018.

[12] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Deep self-organizing maps for unsupervised image classification," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5837–5845, Nov. 2019, doi: 10.1109/TII.2019.2906083.

[13] T. Kohonen, *Self-Organizing and Associative Memory*, vol. 8. Berlin, Germany: Springer-Verlag, 1989.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Amsterdam, The Netherlands: Elsevier, 2013.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 8, 1st ed. New York, NY, USA: Springer-Verlag, 2006.

[16] A. R. Abas, "Adaptive competitive learning neural networks," *Egyptian Informat. J.*, vol. 14, no. 3, pp. 183–194, Nov. 2013, doi: 10.1016/j.eij.2013.08.001.

[17] A. R. Abas, "On determining efficient finite mixture models with compact and essential components for clustering data," *Egyptian Informat. J.*, vol. 14, no. 1, pp. 79–88, Mar. 2013, doi: 10.1016/j.eij.2013.02.002.

[18] C. Mathworks, *Parallel Computing Toolbox TM User's Guide R 2018 A, 6.12*. Portola Valley, CA, USA: MathWorks, 2018.

[19] H. Ü. Dinkelbach, J. Vitay, F. Beuth, and F. H. Hamker, "Comparison of GPU- and CPU-implementations of mean-firing rate neural networks on parallel hardware," *Netw., Comput. Neural Syst.*, vol. 23, no. 4, pp. 212–236, Dec. 2012, doi: 10.3109/0954898X.2012.739292.

[20] N. Ploskas and N. Samaras, *GPU Programming in MATLAB*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[21] N. Ploskas and N. Samaras, "Parallel computing toolbox," in *GPU Programming in MATLAB*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 37–70.

[22] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. AAAI Conf. Artif. Intell.*, Aug. 2011, vol. 25, no. 1, pp. 1–6. Accessed: Feb. 24, 2021. [Online]. Available: https://www.aaai.org

[23] R. Venkat and K. S. Reddy, "Clustering of huge data with fuzzy C-means and applying gravitational search algorithm for optimization," *Int. J. Recent Technol. Eng.*, vol. 8, no. 5, pp. 3206–3209, Jan. 2020, doi: 10.35940/ijrte.D9130.018520.

[24] D. Sculley, "Web-scale K-means clustering," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 1177–1178, doi: 10.1145/1772690.1772862.

[25] F. Nie, D. Xu, I. W. Tsang, and C. Zhang, "Spectral embedded clustering," in *Proc. IJCAI*, 2009, pp. 1181–1186.

[26] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[27] A. Arshad, S. Riaz, L. Jiao, and A. Murthy, "A semi-supervised deep fuzzy C-mean clustering for two classes classification," in *Proc. IEEE 3rd Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Oct. 2017, pp. 365–370, doi: 10.1109/ITOEC.2017.8122317.

[28] V. Rayala and S. R. Kalli, "Big data clustering using improvised fuzzy C-means clustering," *Revue d'Intell. Artificielle*, vol. 34, no. 6, pp. 701–708, Dec. 2020, doi: 10.18280/ria.340604.

[29] Q. Feng, L. Chen, C. L. Philip Chen, and L. Guo, "Deep fuzzy clustering—A representation learning approach," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1420–1433, Jul. 2020, doi: 10.1109/TFUZZ.2020.2966173.

[30] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759, doi: 10.24963/ijcai.2017/243.

[31] M. Moshtaghi, T. C. Havens, J. C. Bezdek, L. Park, C. Leckie, S. Rajasegarar, J. M. Keller, and M. Palaniswami, "Clustering ellipses for anomaly detection," *Pattern Recognit.*, vol. 44, no. 1, pp. 55–69, Jan. 2011, doi: 10.1016/j.patcog.2010.07.024.

[32] A. Wilbik, J. M. Keller, and J. C. Bezdek, "Linguistic prototypes for data from eldercare residents," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 110–123, Feb. 2014, doi: 10.1109/TFUZZ.2013.2249517.

[33] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018, doi: 10.1109/TIP.2017.2772836.

[34] T. Rajesh and R. S. M. Malar, "Rough set theory and feed forward neural network based brain tumor detection in magnetic resonance images," in *Proc. Int. Conf. Adv. Nanomater. Emerg. Eng. Technol.*, Jul. 2013, pp. 240–244, doi: 10.1109/ICANMEET.2013.6609287.

[35] S. Zhou, Q. Chen, and X. Wang, "Fuzzy deep belief networks for semi-supervised sentiment classification," *Neurocomputing*, vol. 131, pp. 312–322, May 2014, doi: 10.1016/j.neucom.2013.10.011.

[36] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2017, Mar. 2017, pp. 1196–1205. Accessed: Feb. 22, 2021. [Online]. Available: http://arxiv.org/abs/1703.01780

[37] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," Oct. 2016, *arXiv:1610.02242*. Accessed: Feb. 22, 2021. [Online]. Available: http://arxiv.org/abs/1610.02242

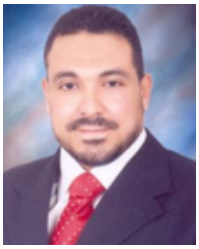
[38] S. P. Chatzis and G. Tsechpenakis, "A possibilistic clustering approach toward generative mixture models," *Pattern Recognit.*, vol. 45, no. 5, pp. 1819–1825, May 2012.

[39] J. Liu and J. Han, "Spectral clustering," in *Data Clustering: Algorithms and Applications*, 1st ed., C. Aggarwal and C. Reddy, Eds. Boca Raton, FL, USA: CRC Press, 2013, pp. 177–200.

[40] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2001, pp. 849–856.

[41] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000, doi: 10.1109/34.868688.

[42] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra Appl.*, vol. 421, nos. 2–3, pp. 284–305, Mar. 2007, doi: 10.1016/j.laa.2006.07.020.
[43] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: https://arxiv.org/abs/1708.07747
[44] *USPS Dataset | Kaggle*. Accessed: Feb. 24, 2021. [Online]. Available: https://www.kaggle.com/bistaumanga/usps-dataset
[45] A. Amelio and C. Pizzuti, "Correction for closeness: Adjusting normalized mutual information measure for clustering comparison," *Comput. Intell.*, vol. 33, no. 3, pp. 579–601, Aug. 2017, doi: 10.1111/coin.12100.



AHMED R. ABAS received the bachelor's degree in electronics and communications engineering from Zagazig University, Egypt, in 1993, the M.Sc. degree in computer engineering from Cairo University, Egypt, in 1999, and the Ph.D. degree in computer science from Exeter University, U.K., in 2005. He is currently an Assistant Professor and a Master's Supervisor with the Computer Science Department, Faculty of Computer and Informatics, Zagazig University. He is also a middle aged and young expert. His research interests include deep learning, neural networks, artificial intelligence, natural language processing, data mining, parallel processing, and pattern recognition.

MOHAMED G. MAHDY received the bachelor's degree in computer science from the Faculty of Computers and Information, Zagazig University, Egypt, in 2016, where he is currently pursuing the master's degree with the Faculty of Computer and Informatics. He is currently a Teaching Assistant with the Computer Science Department, High Institute of Computer Science and Information System, New Cairo, Egypt. He is also a young age and young expert. His research interests include deep learning, neural networks, artificial intelligence, parallel processing, pattern recognition, and big data analytics.

TAREK M. MAHMOUD received the bachelor's degree in mathematics from Minia University, Egypt, in 1984, the M.Sc. degree from Assiut University, Egypt, in 1991, and the Philosophy Doctorate (Dr.Ing.) degree in computer science (computer networks) from Bremen University, Germany, in 1997. He was a Professor of computer science with the Computer Science Department, Faculty of Science, Minia University. He is currently a Professor of computer science with the Faculty of Computers and Artificial Intelligence, University of Sadat City, Egypt. His research interests include pattern recognition, social networks analytics, web and text mining, artificial intelligence, natural language processing, and computer networks.