IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 July 2024, accepted 5 August 2024, date of publication 8 August 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440518

RESEARCH ARTICLE

DGBPSO-DBSCAN: An Optimized Clustering Technique Based on Supervised/Unsupervised Text Representation

ASMA KHAZAAL ABDULSAHIB[1], M. A. BALAFAR[1], AND ARYAZ BARADARANI[2], (Senior Member, IEEE)

[1]Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, East Azerbaijan 51666-16471, Iran
[2]Center for Diagnostic Imaging Research, Tessonics Inc., Windsor, ON N9A, Canada

Corresponding author: M. A. Balafar (balafarila@tabrizu.ac.ir)

ABSTRACT Density-based spatial clustering of noisy applications (DBSCAN), a widely used density-based clustering technique, faces challenges in determining its key parameter, Eps, leading to manual specification and suboptimal clustering outcomes. Additionally, its time complexity poses limitations. This study introduces a novel approach to enhance the algorithm's performance. We combined the DBSCAN algorithm with another approach, the particle swarm optimization, based on a novel way to represent text, termed a dependency graph; this method is recognized as DGBPSO-DBSCAN (Dependency Graph Based Particle swarm algorithm for DBSCAN). This paper focuses on employing a novel approach for PSO variable upgrading to explore the Eps range more rapidly and effectively. This method offers a selection of informative elements from the text, where the initial groups are derived from the graph-based degree centrality, the PSO method is used to choose the most compelling features, and the DBSCAN algorithm is used for text clustering. The experimental findings indicate that the modified PSO is used to improve DBSCAN. We compared the outcomes of the suggested technique to those of the standard clustering algorithm. According to the assessment criteria MSE, accuracy, precision, recall, and F-measure, our technique has been demonstrated to be superior to the conventional method.

INDEX TERMS DBSCAN algorithm, decision tree, degree centrality, dependency graph.

I. INTRODUCTION

A. BACKGROUND

Clustering algorithms are indispensable tools for partitioning extensive collections of text documents into coherent groups. These algorithms can be categorized as either vector space or graph-based methods. Vector space algorithms operate directly within a multi-dimensional feature space, where each data object corresponds to a point in the space [1]. Ideally, clusters are composed of closely located points, distinctly separate from other clusters. Conversely, graph-based approaches rely on pairwise associations among data objects, often depicted as a graph with nodes and edges representing data objects and similarities, respectively [2].

Evaluating the quality of a clustering solution poses a challenge, particularly when the data does not exhibit clear clustering patterns aligned with specific user objectives. Moreover, different datasets may necessitate distinct clustering criteria or methodologies, and various solutions may yield comparable results for the same dataset. Consequently, numerous clustering techniques have been developed, each grounded on specific assumptions regarding data cluster properties.

Text clustering, a fundamental task in text mining, involves categorizing documents based on their thematic content. This process finds application in diverse domains such as data management, indexing, and web content mining [3]. Despite its utility, text clustering encounters challenges stemming from high dimensionality, large volumes, and intricate semantics.

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

In this study, we leverage DBSCAN, a widely adopted clustering technique renowned for its effectiveness in network security and data mining. DBSCAN excels in detecting clusters of varying shapes and sizes, yet it is limited when applied to datasets with high dimensionality and extensive data volumes. These drawbacks increase time complexity and suboptimal clustering results [4].

To address these challenges, we propose a novel approach termed DGBPSO-DBSCAN. Our method employs an effective text representation strategy by selecting pertinent features or words that encapsulate document essence, thereby reducing data dimensionality. Subsequently, we apply the DBSCAN algorithm to cluster the text data, facilitating the extraction of informative features. Finally, each cluster is equipped with a decision tree classifier for refined classification.

Another challenge in DBSCAN lies in manually selecting the radius parameter (Eps). To address this, an adapted version of DBSCAN, named DGBPSO-DBSCAN (Density-Grid-Based Parallel Shared Nearest Neighbor DBSCAN), is proposed for efficient density-based clustering on large datasets. This algorithm tackles issues related to memory usage, parallel computing, and scalability.

This paper introduces a novel adaptive approach called DGBPSO-DBSCAN, which comprises six phases. In the first phase, datasets are collected from three standard sources. Subsequently, the datasets undergo preprocessing, and the text data are transformed into a dependency graph in three steps. The fourth phase employs DGBPSO algorithms to identify significant document characteristics. Finally, the DBSCAN algorithm is utilized to cluster the documents.

II. NOVELTY OF DGBPSO-DBSCAN

The DGBPSO-DBSCAN algorithm introduces a novel integration of graph-based centrality metrics and Particle Swarm Optimization (PSO) to enhance the DBSCAN clustering technique. Unlike traditional DBSCAN, which relies solely on density-based metrics, our method uses graph centrality to identify core points more accurately. The PSO component optimizes the clustering parameters dynamically, improving both clustering accuracy and efficiency. This unique combination of techniques allows DGBPSO-DBSCAN to effectively handle high-dimensional data and complex clustering structures, which is a significant advancement over existing methods.

Our research presents a unique solution to address the limitations of clustering algorithms, including the enhancement of algorithm parameters and performance. Clustering aims to categorize texts into groups based on their subjects, ensuring that each class represents a distinct topic. Figure 1 represents the general text clustering structure and its related operations.

III. LITERATURE REVIEW

A. GRAPH-BASED CLUSTERING ALGORITHMS

Clustering stands out as one of the most effective solutions in intelligent engineering. Its primary aim is to group similar entities based on shared characteristics. The choice of clustering method depends on various factors such as the nature of the data, the specific objectives, and the requirements of the task at hand.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) emerges as a frequently utilized density-based clustering technique in this study. Specifically designed to detect and differentiate noise from clusters of arbitrary shapes, DBSCAN offers versatility in handling complex datasets.

Recent advancements in clustering algorithms have leaned towards leveraging graph representations for text data. In this study, we adopt the dependency graph model. A dependency graph, being a directed graph, illustrates relationships among multiple entities. It facilitates the establishment of either an ordered hierarchy reflecting the graph's dependencies or the absence of such dependencies [5]. This graphical representation accurately captures the interdependencies, as illustrated in the accompanying figure.

A dependency graph comprises a collection of nodes representing hypotheses, associated with confidence values, and interconnected by dependency edges, which govern the assignment of confidences. These confidence values can be fully determined, partially determined, or undetermined. Figure 2 clarifies the dependency graph of transferring a sentence.

A dependency graph is defined as a directed graph $G = (V, E)$, where V denotes a collection of nodes (binaries) and E denotes a set of edges (dependencies). The dependency graph was extracted by Colvett et al. using the object-based dependability exploitation paradigm (ODEM) [6]. ODEM is a system design methodology that improves reliability by simplifying design, testing, and maintenance through the grouping of components into objects. Critical applications such as autonomous systems, aircraft, and healthcare find it helpful. Nodes that represent classes appear in the ODEM dependency network. The classification of these nodes (class, interface, annotation, etc.) and their vision, abstraction, and finality are defined by their annotations. Along with the whole class name (package Name. class Name), a list of one-way relations (dependencies) and a dependency categorization annotation (uses, extends, or implements) are all provided in each node. This method improves visualization, making the graph appear less complicated.

The researchers suggest a graph-based textual content representation integrating varying degrees of formal natural language representation [7]. This schema considers various linguistic levels, including lexical, morphological, syntactical, and semantic. The suggested representation architecture is supplemented with a strategy for extracting valuable text patterns based on the concept of minimum pathways in the graph.

According to the results, the suggested graph-based multilevel linguistic representation architecture may be effectively utilized in the broader context of document interpretation. A graph-based clustering algorithm based on a unique

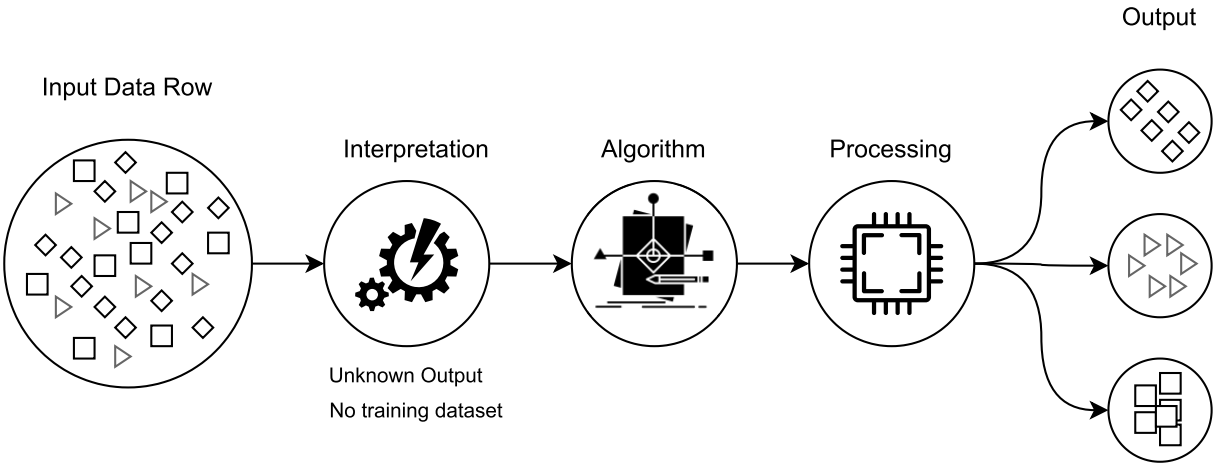

FIGURE 1. The general clustering diagram of text representation.

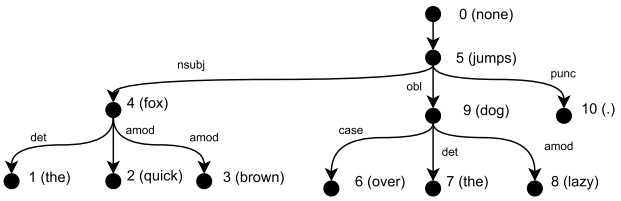

FIGURE 2. The dependency graph. A dependency graph is created by transforming the sentence "The quick brown fox jumps over the lazy dog." as an example.

density-of-graph structure was presented in this work [8]. Semantic enrichment, natural language processing (NLP), and contextual data are all integrated into the graph-based multi-level linguistic representation architecture to improve document understanding. Information retrieval and recommendation systems can benefit from its support for text summarization, clustering, cross-document analysis, machine learning integration, and interactive exploration. The suggested method for classifying dense and sparse nodes uses each node's defined density coefficients. Dense and sparse nodes assigned to various clusters are utilized to determine the clusters' primary structures. Experiments were done on several simulated data sets and benchmark datasets to investigate the characteristics of the proposed DGBPSO-DBSCAN and compare its performance to that of existing spectral clustering and modularity-based approaches. The experimental findings demonstrated that the proposed clustering method outperformed its rivals, even when the data cluster structures were inherently noisy and nonlinearly distributed.

In contrast, [9] provides a text clustering method that does not rely on user-defined parameters. Since text documents and their relationships are represented as graph nodes and edges, the graph community identification method is employed to solve the text clustering problem. Researching graph community identification methods in text clustering problems is imperative to enhance algorithms, find hidden structures, optimize parameter settings, handle scalability issues, and connect text and network analysis. The researchers in [10] propose a graph-based clustering method for detecting crime report labels among massive untagged crime corpora. The successful technique in [11] provided a general Graph-Based System (GBS) for multi-view clustering and examined how different graph metrics affected the effectiveness of multi-view clustering. The suggested method may automatically weigh the graph created by each view to develop a single, unified graph, producing final clusters without the need for other clustering techniques.

The researchers in [12] propose a natural neighbor graph-based cut-point clustering technique (CutPC). When a cut-point value exceeds the critical value, the CutPC technique executes noise cutting. Without prior information or parameter settings, this approach can automatically identify clusters of any shape and find outliers.

The authors in [13] main aim is to develop a unique graph-based semantic representation model for Arabic text that will enhance specific Arabic NLP applications, such as textual entailment. The proposed graph-based model can increase textual entailment detection performance and clustering approach precision. In these tests [14], [15] and [25], a unique type of graph known as Knowledge Graph Embedding was used to enhance the prediction performance of graph models. Where [25] presents Text-enhanced Knowledge Graph Embedding, a novel integrated model for inference over entities, relations, and texts (TKGE). The authors of the second research suggest the Knowledge Graph Embedding with Concepts (KEC) model for acquiring knowledge graphical representations with improved concept graph data. This proposed DGBPSO-DBSCAN resulted in statistically significant performance increases relative to a range of strong baselines.

B. DBSCAN-BASED MODEL

DBSCAN, an unsupervised learning technique employing density clustering, is capable of identifying noise samples

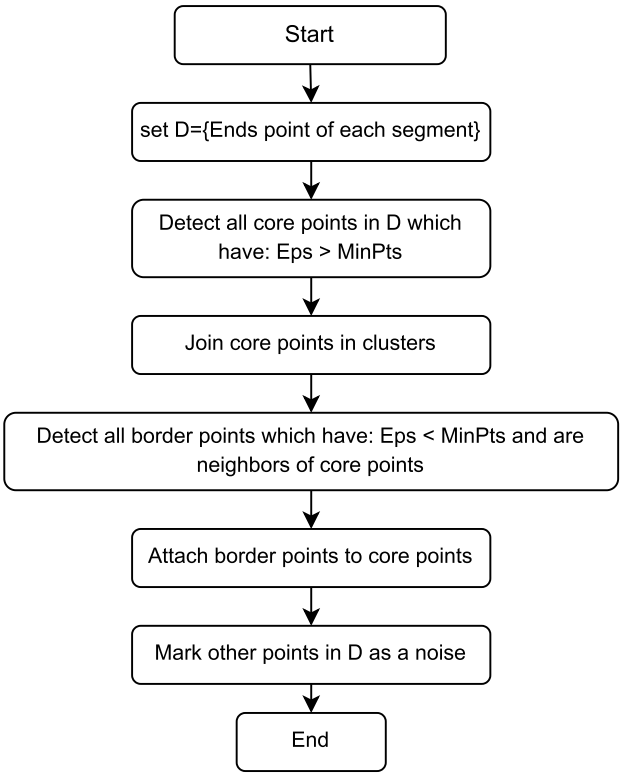

FIGURE 3. The general flowchart of DBSCAN algorithm.

within a dataset and clustering data of any format. However, its clustering efficiency heavily relies on the choice of MinPts and Eps parameters, necessitating theoretical guidance for parameter selection. This section discusses recent research aimed at enhancing the DBSCAN algorithm or combining it with other techniques. Figure 3 clarifies the principle of the DBSCAN algorithm.

DBCLUM, introduced in [16], extends DBSCAN for density-based cluster detection. Unlike DBSCAN, DBCLUM can identify clusters with varying densities and close clusters. Experiments have shown that DBCLUM outperforms DBSCAN in terms of speed by 11% to 52%. In [17], the Particle Swarm Optimization Density-Based Spatial Clustering of Applications with Noise (PSODBSCAN) approach is proposed, automatically improving the critical parameters of the DBSCAN algorithm. Experimental results demonstrate the effectiveness of PSODBSCAN in recognizing background noise and preserving signal photons in raw data.

A novel parameter optimization strategy for DBSCAN is proposed in [18], utilizing the Multi-Verse Optimizer algorithm. Enhanced MVO quickly identifies the optimal clustering accuracy and appropriate MinPts and Eps parameters for DBSCAN. To address the subpar clustering results and low efficiency of DBSCAN, an improved adaptive density-based spatial clustering of applications with noise approach is presented in [19], based on evolutionary algorithms and MapReduce.

A hybrid algorithm combining PSO and DBSCAN is introduced in [20] for document clustering, aiming to increase cluster accuracy based on content. Reference [9] presents a DBSCAN based on Particle Swarm Optimization (PSO) that automatically computes MinPoints and Epsilon values for given input data and identifies spatial hotspots. In [10], popular variations of particle swarm optimization methods and differential evolution methods are employed to optimize DBSCAN clustering parameters. Composite DE (CoDE) stands out as the superior method for parameter optimization.

The I-DBSCAN algorithm is enhanced using PSO in [11] to improve clustering accuracy. In [13], a novel Particle Swarm Optimized Density-based Clustering and Classification (PODCC) approach is proposed to overcome DBSCAN drawbacks. PODCC utilizes SPSO-2011 to search for optimal parameters for density-based clustering and categorization. Experimental results across various datasets demonstrate the effectiveness of the DGBPSO-DBSCAN in improving clustering accuracy and efficiency.

IV. PROPOSED DGBPSO-DBSCAN

The five stages of the approach utilized to achieve the goals of this study's research are briefly outlined in this section. A novel method named Dependency Graph-based Particle Swarm Optimization for the DBSCAN algorithm (DGBPSO-DBSCAN) is employed to address the time complexity and parameter definition challenges inherent in density clustering algorithms.

During the preprocessing stage, which involves tasks such as sentence splitting, tokenization, word removal, and stemming, widely cited standard comparison datasets in the field are utilized to evaluate the outcomes of various methodologies. In the subsequent phase, features or words are transformed into a dependency graph, a graph type specifically chosen for its suitability in representing linguistic dependencies.

The degree centrality algorithm is then applied to the set of extracted features from the dependency graph to assess their importance. Next, the PSO algorithm is employed for text resulting from the preceding phase (DGBDC). Instead of generating the initial population randomly, the DGBDC method is utilized to select the most informative features as the initial population, thereby enhancing the PSO algorithm's efficiency in finding optimal parameters quickly.

This unique strategy effectively addresses challenges inherent in clustering algorithms, such as parameter optimization and performance enhancement, particularly in reducing time complexity for datasets with significant dimensionality. Figure 4 illustrates the research design, depicting the application of PSO and DBSCAN clustering algorithms on three distinct text document datasets to partition documents into predefined groups based on content similarity, representing the five phases of the process.

A. DOCUMENT COLLECTION AND PREPROCESSING

The conventional text document clustering techniques and a well-liked subset of the 20 Newsgroup dataset were utilized to create the data set for this study. Approximately

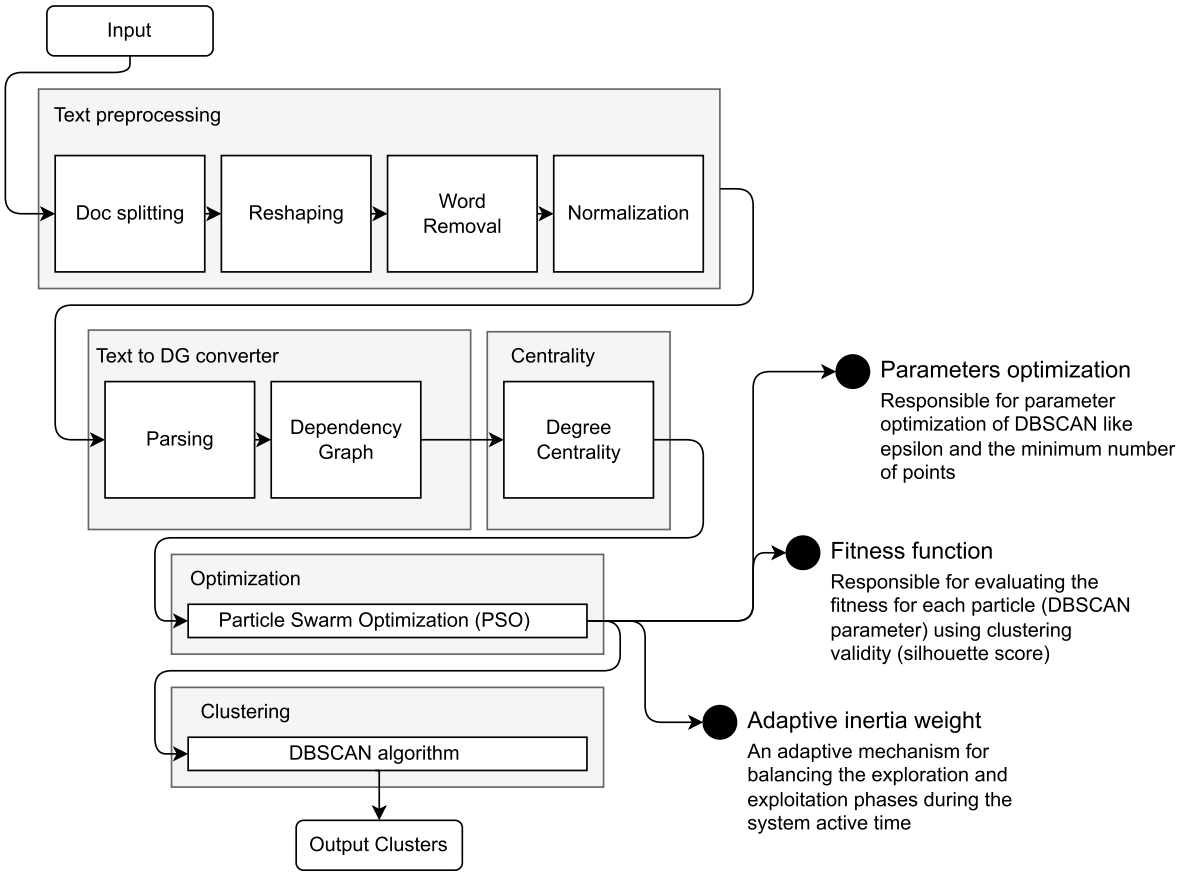

FIGURE 4. The general structure of the proposed DGBDC-DTDBSCAN.

20,000 newsgroup documents from 20 different newsgroups were dispersed equally throughout the data set. IMBD Review movies: The Internet Movie Database (IMDB) Movie evaluations dataset is a binary sentiment analysis dataset comprising 50,000 positive and negative evaluations. The sample has an equal number of good and adverse reviews. Only the most polarising reviews are taken into account. A negative review is given a score of 4, while a positive review is given a score of 7. Up to 30 reviews are given to each movie. The dataset also includes unlabeled data. Dataset and Iris Dataset: The data set comprises 50 samples from the three Iris species (Iris setosa, Iris virginica, and Iris versicolor). The length and width of the sepals and petals in cm were measured for every specimen. By fusing these four features, Fisher developed a linear discriminating model to distinguish the species. The Fisher linear discriminant (FLD) aims to locate projections on a line that are clearly distinguished from the projections of instances from various samples.

Pre-processing is the method of transforming text data from the document into frameworks for text mining. Processing's main goal is to remove key elements or significant words from online news text documents to increase the words' usefulness to the article and the context. A text almost always has many unnecessary words that can affect the document's readability.

Four actions are typically involved in the syntax analysis phase to remove words or phrases from the text: Divide the text into sentences, Tokenization separates a stream of text files into words or terms, Stop words are a group of often-used words with low weighting, high frequency, and short phrases useful in text-based clustering, and Finally, Stemming converts acceptable inflectional versions of some words to the same root by removing the prefixes and suffixes of each word.

B. TEXT-TO-GRAPH CONVERSION

The syntactic structure of sentences is determined by parsing them. As a result, the "Link Grammar Parser" (LGP) was created. This parser provides the syntactical connection among the words in a sentence since it is based on dependency grammar and depends on context-free grammar. LGP is simpler than more complicated parsing approaches while also providing a richer semantic structure than ordinary context-free parsers [21]. A dependency parser is needed to extract word connections from the source sentences. The goal of a parser is to analyze input text and provide the best (and preferred) parse tree as an output.

In this paradigm, each document is represented as a dependency graph, with each node corresponding to a word that serves as the document's meta-description. The

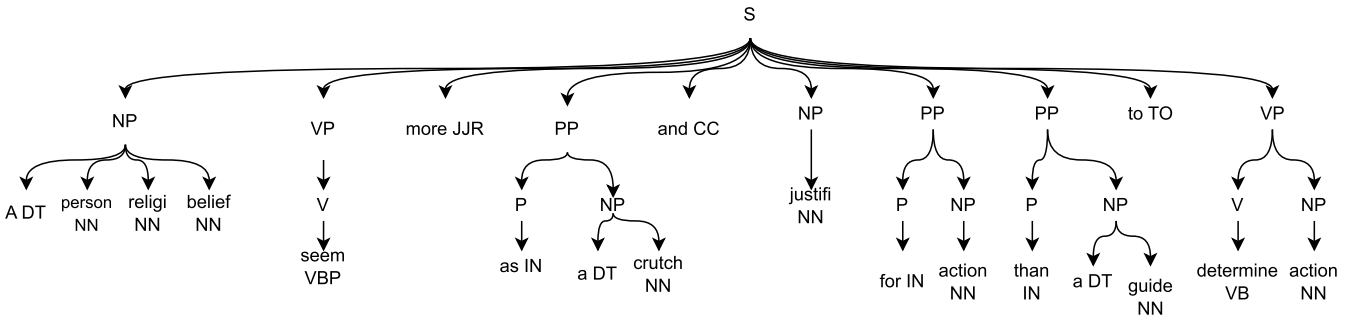

FIGURE 5. Text passing process.

semantic relations between word pairs are captured via the edges between nodes. The dependency graph is projective because all the words are written in a linear sequence. Edges can be removed without crossing over the text, indicating that the word and all of its offspring (dependents and dependents of her dependents, and so on) form a connected succession of words in a phrase. Generate a $(n-1)$ matrix in Matlab by converting the result into a matrix and then building a dependency network. The nodes of a dependency representation, a named directed graph, indicate lexical items and the arcs show dependence relationships from heads to dependencies. The most fundamental measure of centrality is degree centrality; therefore, use it instead. The degree criteria are used to quantify centrality at a very local level. Given a graph G, denote the set of vertices of G as $V(G)$, and then the degree centrality for any $v \in V(G)$ is defined as in eq.1:

$$CD(v) = \frac{\text{degree of } v}{|V(G)|} \tag{1}$$

Only the network's local topology is considered by degree centrality [25] Unlike a network's overall impact, it might be seen as a gauge of instant influence. Centrality indicates a node's importance within the graph.

C. PARSING

Understanding the structure of sentences involves parsing them, a process crucially addressed by the development of the "Link Grammar Parser" (LGP). Unlike complex parsing methods, LGP, grounded in dependency grammar and context-free grammar, offers a simpler yet richer syntactical interpretation of sentences. Dependency parsing is essential for uncovering word relationships within sentences. Ultimately, parsers aim to analyze text inputs, generating optimal parse trees as output, and facilitating comprehension. Figure 5 clarifies the text parsing process in the proposed method.

D. DEGREE CENTRALITY

The most basic metric of centrality is degree centrality. We use degree centrality to assess the quality of the text relationship generated by the dependency graph. The number of edges upon a node measures the degree of centrality. The simplest CM is the degree centrality, which counts the number of edges restricted to nodes. Its foundation is that crucial texts have the most connections to other documents in a dependent network. Applied to a dependency graph, A global centrality measures the distance between nodes in the entire system, whereas a local centrality determines the distance among nodes in a specified radius. In an undirected, unweighted network, a node's degree is just a few edges that connect it to other nodes. A node with a high degree relevance value merely has greater connections than is typical for that graph. The degree of a node V represents the number of words that co-occur with the word corresponding to V. Let $d(V)$ be the set of nodes connected to V; the degree centrality of a node V is given by eq.1. Where $|(V)|$ is the degree of V and $|V(G)|$ is the number of vertices in G. Only the network's local topology is considered by the degree of centrality [25]. It may be seen as a gauge of local influence inside the network instead of network-wide impact. Centrality is a criterion that quantifies the role of a node in the graph.

The weakest criterion of resemblance is when a document has a very low degree of centrality (G4). Here, the resulting object becomes the initial population for PSO. To run the PSO and DBSCAN algorithms, we calculate the cosine distance based on centrality metrics. When texts are represented as word vectors, their similarity is proportional to their correlation. The cosine of the angle between vectors, also known as cosine similarity, is one of the most often used similarity measures for text documents in various information retrieval applications and clustering. In this research, we have a collection of significant words that we transform into a numerical array A of dimension $(n \times n)$, where n is the number of words, and we compute degree centrality on the A matrix. This is done after creating the dependency network. By doing this, we can create a C centrality matrix with dimensions $(n \times c)$, where n represents the number of documents and c represents the number of centralities. Given that we utilize 1 centrality metric in this instance, $c = 1$. On matrix C, we calculate cosine distance. Two document vectors describing the profiles A and B are used to calculate the cosine of the angle produced. Formally, the cosine distance is:

$$\text{Cos}(\alpha) = \frac{A \cdot B}{\|A\| \|B\|} \tag{2}$$

There are several uses of cosine similarity in supervised, unsupervised, and reinforcement algorithms in diverse machine learning domains [22], [23], and [24]. To clarify the cosine similarity work on the dataset used in this study, for example, In the IMDB dataset movies platform, we have movies from four distinct genres (Action, Comedy, Science fiction, and Horror). If we make a feature space with each genre as a dimension, the result is presented in figure 6:

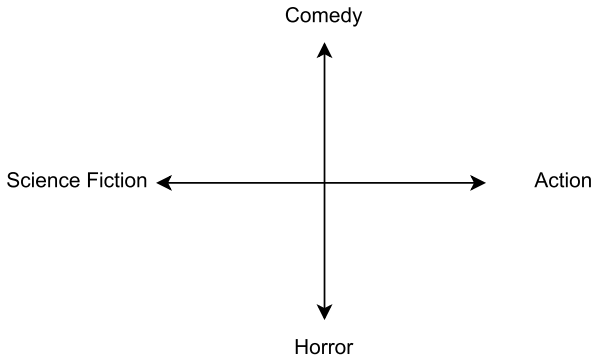

FIGURE 6. Similarity functions.

The most basic metric of centrality is degree centrality. We use degree centrality to assess the quality of the text relationship generated by the dependency graph. The number of edges upon a node measures the degree of centrality. The simplest CM is the degree centrality, which counts the number of edges restricted to nodes. Its foundation is that crucial texts have the most connections to other documents in a dependent network. Applied to a dependency graph, A global centrality measures the distance between nodes in the entire system, whereas a local centrality determines the distance among nodes in a specified radius. In an undirected, unweighted network, a node's degree is just a few edges that connect it to other nodes. A node with a high degree relevance value merely has greater connections than is typical for that graph. The degree of a node V represents the number of words that co-occur with the word corresponding to V. Let $d(V)$ be the set of nodes connected to V; the degree centrality of a node V is given by eq.1. Where (V) is the degree of V and $|V(G)|$ is the number of vertices in G. Only the network's local topology is considered by the degree of centrality [25]. It may be seen as a gauge of local influence inside the network instead of network-wide impact. Centrality is a criterion that quantifies the role of a node in the graph.

The weakest criterion of resemblance is when a document has a very low degree of centrality (G4). Here, the resulting object becomes the initial population for PSO. To run the PSO and DBSCAN algorithms, we calculate the cosine distance based on centrality metrics. When texts are represented as word vectors, their similarity is proportional to their correlation. The cosine of the angle between vectors, also known as cosine similarity, is one of the most often used similarity measures for text documents in various information retrieval applications and clustering. In this research, we have a collection of significant words that we transform into a numerical array A of dimension $(n \times n)$, where n is the number of words, and we compute degree centrality on the A matrix. This is done after creating the dependency network. By doing this, we can create a centrality matrix C with dimensions $(n \times c)$, where n represents the number of documents and c represents the number of centralities. Given that we utilize 1 centrality metric in this instance, $c = 1$. On matrix C, we calculate cosine distance. Two document vectors describing the profiles A and B are used to calculate the cosine of the angle produced.

There are several uses of cosine similarity in supervised, unsupervised, and reinforcement algorithms in diverse machine learning domains. To clarify the cosine similarity work on the dataset used in this study, for example, In the IMDB dataset movies platform, we have movies from four distinct genres (Action, Comedy, Science fiction, and Horror). If we make a feature space with each genre as a dimension, the results are presented in figure 7:

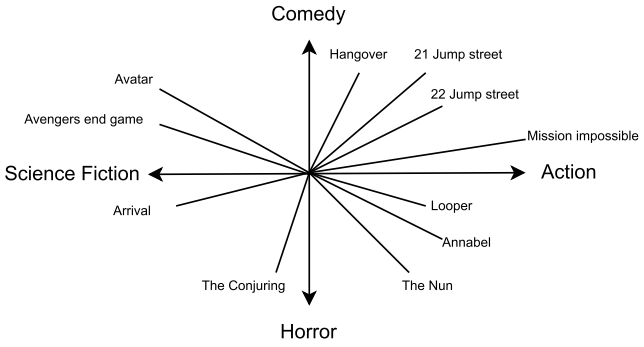

FIGURE 7. Cosine functions.

E. SWARM INTEGRATION

Swarm Intelligence (SI) is an artificial intelligence built on the collective behavior of decentralized and self-organized systems. These systems generally comprise a group of primary players interacting with one another and their surroundings [14]. Many SI-based algorithms have been used to highlight sections in recent decades, including Genetic Algorithm (GA), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). PSO has been successfully used to redact high-dimensional datasets among the SI-based algorithms in the feature selection issue. The primary particle swarm optimization technique comprises "n" particles, where every particle's location in D-dimensional space represents a prospective resolution. The particles change their state based on the three principles below:

1) to maintain inertia,
2) to alter the situation by its most optimistic state,
3) to alter the situation based on the swarm's most favorable position.

The inertia weight is crucial in balancing the global and local search trade-off. A considerable inertia weight (w) encourages particles to explore a vast region (global search).

In contrast, a little inertia weight (w) encourages particles to seek a smaller area (local search). In the following, a considerable inertia value is imported at the start of the search ($w = w_{\max}$), and it reduces until it reaches ($w = w_{\min}$) (the lowest value). Figure 8 shows a schematic updating a particle's location in two iterations. Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique inspired by the social behavior of birds flocking or fish schooling. Each particle in the swarm represents a potential solution in the search space. The particles adjust their positions based on their own experience and the experience of their neighbors.

INITIALIZATION

Each particle i has the following properties:

- Position: $\mathbf{x}_i \in \mathbb{R}^n$
- Velocity: $\mathbf{v}_i \in \mathbb{R}^n$
- Personal best position: $\mathbf{p}_i \in \mathbb{R}^n$

The global best position found by any particle is $\mathbf{g} \in \mathbb{R}^n$.

The velocity and position of each particle are updated using the following equations:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{g}(t) - \mathbf{x}_i(t)) \tag{3}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{4}$$

where:

- w is the inertia weight
- c_1 and c_2 are the cognitive and social coefficients, respectively
- r_1 and r_2 are random numbers uniformly distributed in [0, 1]

Each particle updates its personal best position if the new position is better:

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{p}_i(t)) \\ \mathbf{p}_i(t) & \text{otherwise} \end{cases} \tag{5}$$

The global best position is updated if any particle achieves a better position:

$$\mathbf{g}(t+1) = \arg\min_{\mathbf{p}_i(t+1)} f(\mathbf{p}_i(t+1)) \tag{6}$$

In the context of optimizing the DBSCAN clustering algorithm, each particle represents a set of DBSCAN parameters, typically the epsilon (ϵ) and the minimum number of points ($MinPts$).

$$\mathbf{x}_i = (\epsilon_i, MinPts_i) \tag{7}$$

The fitness function $f(\mathbf{x}_i)$ evaluates the quality of clustering produced by the DBSCAN algorithm with the given parameters. This could be based on metrics such as silhouette score, Davies-Bouldin index, or any other clustering validity index.

PSO ALGORITHM STEPS

1) Initialize the swarm with random positions and velocities.
2) Evaluate the fitness of each particle.
3) Update personal and global best positions.
4) Update velocities and positions of particles.
5) Repeat steps 2-4 until convergence or maximum iterations are reached.

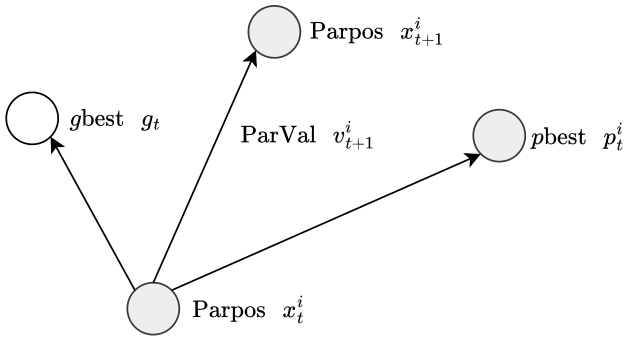

FIGURE 8. Schematic updating a particle's location in two iterations.

The basic particle swarm optimization approach has the following advantages: The core of PSO is intelligence. It applies to both scientific and engineering investigations. In PSO, there will be no overlapping and no mutation computation. The speed of the particle can be utilized to search. Only the most hopeful particle can transmit knowledge to others over generations, and the research rate is exceedingly rapid. Following that, the computation in PSO is relatively simple. It has a higher optimization capacity and can be completed faster than other development computations. The final alternative is PSO, which uses a genuine numeric code generated directly by the solution. The constant of the solution is equal to the number of dimensions.

One of the drawbacks of the particle swarm algorithm is that it does not take the relationship between words, which leads to increased noise, so the DGBDC technique was used to reduce and select only the most useful features and find the relationship between words. Applied PSO algorithm for text resulting from the first phase (DGBDC), where the PSO algorithm is applied to find a new subset of text features, and the initial population of the PSO is generated randomly but using the method DGBDC instead of generating the initial population randomly, utilizes the most informative features as the initial population means features or phrases with high centrality will be considered as a primary solution, this leads to improving the PSO algorithm and thus finding the optimal parameters quickly, where we find a unique strategy for solving the challenges of the clustering algorithm, such as boosting the algorithm parameters and improving the algorithm's performance by reducing the time complexity for the dataset's huge dimensionality.

This study discusses the many forms of fitness functions utilized by PSO. Researchers [15] propose three heuristics:

1) Branching length-based wellness,

2) Assumption grade-based well-being, and
3) Combinations of Fitness to generate test cases for covering the code.

F. BOOSTING THE DBSCAN ALGORITHM PARAMETERS

The optimizations offered by the PSO to the DBSCAN can be summarized as follows:

1) By first representing texts using dependency graphs and then utilizing degree centrality and PSO optimization, the method enhances the quality of features used in clustering. This could potentially lead to better parameter tuning for DBSCAN, as the input data is more informative and structured.
2) The utilization of PSO optimization allows for fine-tuning of parameters such as epsilon and minPts in DBSCAN, potentially leading to improved clustering results.
3) Dependency graphs provide a structured representation of texts, which can help reduce the dimensionality of the dataset by capturing the essential syntactic relationships between words or phrases.
4) By employing PSO optimization, the method aims to optimize the clustering process, potentially reducing the computational burden associated with clustering high-dimensional datasets.
5) DBSCAN itself is known for its efficiency in handling high-dimensional data and its ability to automatically determine the number of clusters without requiring a pre-specified parameter. By integrating DBSCAN into the proposed method, it further contributes to managing the complexity of high-dimensional datasets.

Branch distance is calculated for dependent vertices employing test data. It determines how close the test data needs to be to the true/false criterion to fulfill the demand [28]. Korel's branch distance function analyses branching circumstances [16], [17]. The target path's branch distance is a total of three branch lengths. Using a control framework, the approximation level may be used to gauge how close a person is to reaching a goal [26]. It is determined by counting how many branching nodes a test case does not pass through while taking the desired course. As a result, the amount of approximation should be kept to a minimum. Wegener et al. [27] presented a hybrid strategy that incorporates the abovementioned approaches. The combined fitness function adds a small fixed value to the branch distance.

$$\begin{aligned} \text{Fitness}(t) &= \text{Normalized Branch Distance (NBD)} \\ &\quad + \text{Approximation Level (AL)} \\ \text{NBD} &= 1 - (1.001)^{-\text{distance}} \end{aligned} \tag{8}$$

In this study, we use the second type, which depends on how close the individual is to the target within the features selected as a distinct group using our DGBPSO-DBSCAN. After invoking the fitness function at each location, the global best g and individual best positions (p_i) are determined at each iteration. In general, a higher fitness rating indicates a better position. The particle's best position changes only if the present value of position is greater than the prior best value. Amongst all of the individual best position standards, the one with the greatest level of Fitness is recognized as the overall best. It may be mathematically represented as in equations 9 and 10 respectively.

$$p_i = \begin{cases} p & \text{if } f(p) > f(p_i) \\ p_i & \text{otherwise} \end{cases} \tag{9}$$

$$g = \arg\max f(p_i) \tag{10}$$

The procedure will continue until the termination requirements are satisfied (typically a maximum number of iterations). The location of every particle in the swarm (near experience) is influenced by the position of the most optimistic particle throughout the movement (individual experience) and the positioning of the most optimistic particle in its surroundings. If the complete particle swarm surrounds a particle, the most optimistic location of the surrounding space is comparable to the most optimistic particle in its entirety; this technique is referred to as the full PSO. The partial PSO algorithm is named after the restricted surroundings used in the process [5]. The ideal location for each particle and the position of their surroundings may all be shown, together with their current speed and position. Throughout the optimization process, the PSO helps prevent premature convergence by balancing exploration and exploitation.

V. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity of the DGBPSO-DBSCAN algorithm is a crucial aspect of evaluating its scalability. The complexity of the DBSCAN algorithm is $O(n \log n)$ due to the nearest neighbor search. The PSO component adds complexity of $O(P \times G \times n \times k)$, where P is the population size, G is the number of generations, n is the number of data points, and k is the number of dimensions. Therefore, the overall complexity of the proposed method is $O(n \log n + P \times G \times n \times k)$. This makes it suitable for handling large datasets as long as P and G are kept within reasonable limits.

A. EVALUATION CRITERIA

This section delves into the pivotal criteria employed for evaluating clustering performance and elucidates their utility in predicting class labels. Key evaluation metrics encompass Mean Square Error (MSE), accuracy, precision, recall, and F1-measure. Prior to delving into the intricacies of these metrics, it is imperative to acquaint oneself with several fundamental terms. The term ''positive example'' denotes class samples of interest to the user, while ''negative tuple'' encompasses the remaining examples. Formulas for evaluation criteria typically denote positive examples as P (positive) and negative examples as N (negative). Table 1 clarifies the details of the performance measures that used in this research.

TABLE 1. Classification metrics and their equations.

Metric	Equation
True Positive (TP)	Number of positive examples correctly labeled by the classifier.
True Negative (TN)	Number of negative examples correctly labeled by the classifier.
False Positive (FP)	Number of negative examples erroneously labeled as positive by the classifier.
False Negative (FN)	Number of positive examples mislabeled as negative by the classifier.
Accuracy	$\text{Accuracy} = \frac{TP + TN}{P + N}$
Recall (Sensitivity)	$\text{Recall} = \frac{TP}{TP + FN}$
Precision	$\text{Precision} = \frac{TP}{TP + FP}$

Accuracy, as a fundamental evaluation criterion, represents the percentage of tuples labeled positive and correctly classified as positive:

The F1 measure, a composite metric, harmonizes accuracy and sensitivity:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

In addition to these criteria, the terms N, P, TP, TN, FP, and FN are encapsulated within a confusion matrix, providing a comprehensive summary of the classification outcomes.

VI. EXPERIMENTS AND RESULTS

The circumstances outlined in Table 2 delineate the settings in which the trials were executed. A diverse array of settings was amalgamated to yield optimal outcomes while evaluating the performance of the DBSCAN algorithm employing Particle Swarm Optimization (PSO). To attain the highest accuracy results, this study manipulated various parameters of the PSO method, including population size ($P_{\max}$), inertia weight (w), and maximum number of generations ($t_{\max}$).

In this section, the PSO parameters were configured, setting the dimensions (Dim) to 2 to accommodate the calculation of two parameters. Subsequently, a fitness function was defined to gauge a particle's adaptability to diverse activities. The PSO-based proposed model exhibited substantially lower fitness convergence values, indicative of the algorithm's enhanced ability to discern between optimal local and global solutions, thereby augmenting adaptability. This improvement can be attributed to the adaptive adjustment of inertia weight.

In contrast to linearly decreasing inertia weights, the proposed approach scrutinizes particle spatial dispersion, automatically adjusting inertia weights based on particle distribution distance and fitness deviation, thereby enhancing accuracy. This analytical approach facilitates a better understanding of algorithm convergence behavior, exploration-exploitation balance, and overall performance, leading to improved algorithm suitability for specific optimization problems.

The fitness function played a pivotal role in determining effective values, considering them as prime numbers for the parameter Eps. Consequently, parameter values were automatically determined without prior user selection. An initial population was then generated, with predefined lower and upper limits.

As mentioned in the preceding section, the findings of the assessments will be evaluated using five criteria. Additionally, three datasets - Iris, IMDB, and 20News - will be employed for the evaluations. The assessment results for each dataset, in terms of Recall, F-measure, MSE, Accuracy, and Precision, will be presented individually.

Moreover, an essential aspect of the evaluations involves considering the impact of dataset size variations. Changes in dataset sizes are commonplace in data analysis and

TABLE 2. Evaluation conditions.

Item	Configuration
Processor	11th Gen Intel® Core™ i3-1115G4 @ 3.00GHz 3.00GHz
RAM	8.0 GB
Hard Disk	260 GB SSD
Operating system	Microsoft Windows 10 Ultimate
Programming language	Matlab R2022a

TABLE 3. Evaluation results for IMDB dataset.

Method	Dataset size %	MSE	Precision	Recall	F-Score	Accuracy	Time (s)
DBSCAN	50	0.4933	0.7466	0.5256	0.6169	0.5067	110
	60	0.5843	0.5523	0.516	0.5335	0.4157	120
	70	0.5769	0.4919	0.4979	0.4948	0.4231	154
	80	0.5294	0.6423	0.5277	0.5793	0.4706	126
	90	0.5672	0.4637	0.4898	0.4763	0.4328	135
	100	0.5503	0.5957	0.5102	0.5496	0.4497	167
DGBPSO-DBSCAN	50	0.1867	0.8137	0.8141	0.8133	0.8133	7
	60	0.1685	0.8462	0.7963	0.8099	0.8315	5
	70	0.2019	0.8046	0.7764	0.7835	0.7981	9
	80	0.2017	0.8063	0.7843	0.789	0.7983	9
	90	0.209	0.7876	0.7913	0.7887	0.791	9
	100	0.2215	0.7748	0.7757	0.7752	0.7785	8

TABLE 4. Evaluation criteria for the 20Newsgroup dataset.

Method	Size Dataset	MSE	Precision	Recall	F-Score	Accuracy	Time (s)
DBSCAN	50	4.5345	0.2941	0.2043	0.2411	0.1552	120
	60	5.2286	0.0338	0.1833	0.0570	0.1571	146
	70	5.1852	0.1097	0.2119	0.1445	0.1605	155
	80	5.4731	0.1717	0.1567	0.1638	0.129	146
	90	5.25	0.2313	0.198	0.2133	0.1538	167
	100	5.4828	0.2469	0.1949	0.2178	0.1638	177
DGBPSO-DBSCAN	50	1.9655	0.487	0.4956	0.4526	0.5692	8
	60	2.0714	0.6445	0.5983	0.5876	0.6122	10
	70	1.3951	0.6632	0.6061	0.6193	0.6173	12
	80	1.957	0.5139	0.4901	0.4918	0.5161	17
	90	1.9038	0.5486	0.5224	0.5181	0.5481	13
	100	1.7931	0.5498	0.537	0.5328	0.5431	10

TABLE 5. Evaluation criteria for the Iris dataset.

Method	Size Dataset	MSE	Precision	Recall	F-Score	Accuracy	Time (s)
DBSCAN	50	1.7333	0.1022	0.3333	0.1565	0.3067	112
	60	1.8667	0.0889	0.3333	0.1404	0.2667	115
	70	1.6381	0.1111	0.3333	0.1667	0.3333	110
	80	1.8583	0.092	0.2963	0.1404	0.2667	120
	90	1.6889	0.1111	0.3333	0.1667	0.3333	98
	100	1.6667	0.1111	0.3333	0.1667	0.3333	132
DGBPSO-DBSCAN	50	0.0267	0.9762	0.9744	0.9743	0.9733	5
	60	0.0444	0.9602	0.9602	0.9596	0.9556	8
	70	0.0476	0.9551	0.9515	0.9521	0.9524	7
	80	0.0417	0.9614	0.9609	0.9603	0.9583	12
	90	0.037	0.9639	0.9634	0.9603	0.963	12
	100	0.04	0.9619	0.96	0.9599	0.96	15

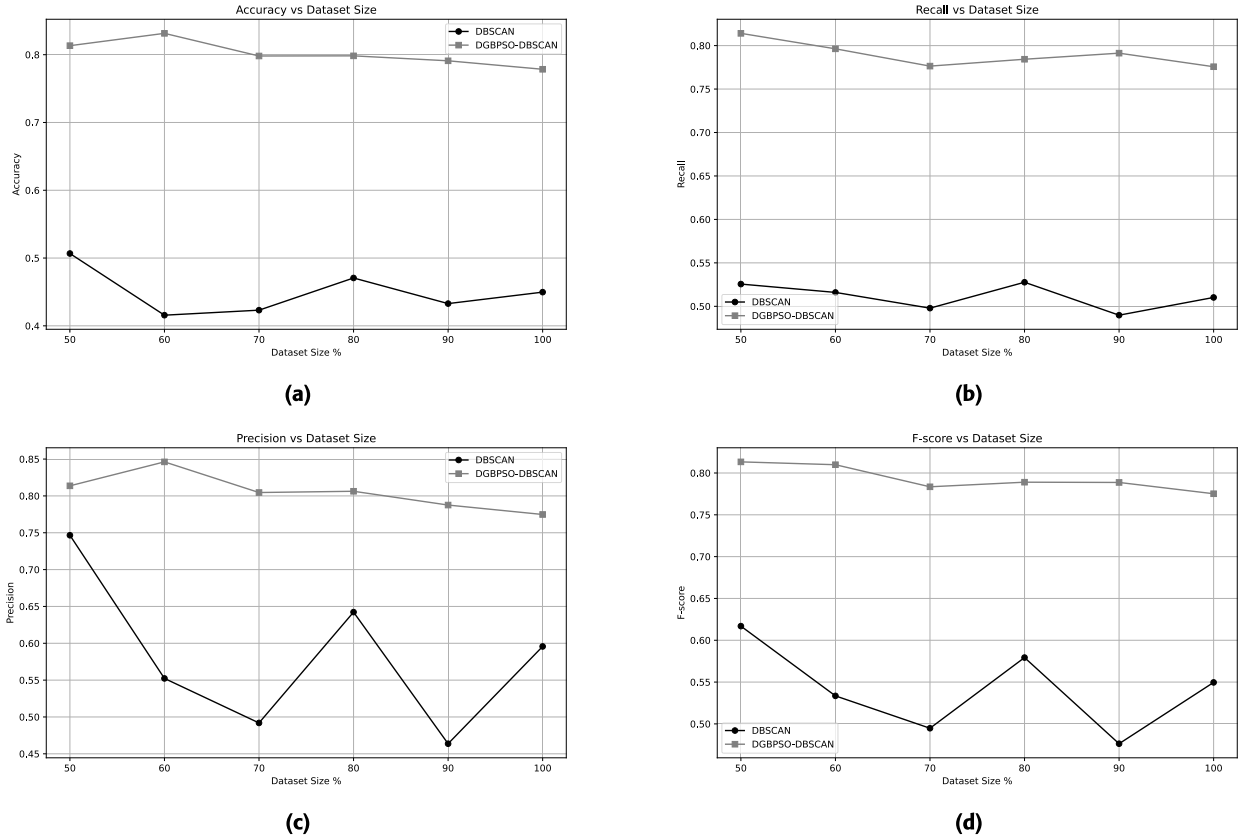

FIGURE 9. The proposed DGBPSO-DBSCAN results in terms of accuracy, recall, precision, and f1-measure compared to traditional DBSCAN algorithm in IMDB dataset.

research for reasons such as resource management, model training speed, data imbalance, and privacy concerns. The performance of each compared method will be assessed in experiments with dataset size variations ranging from 50% to 100%. The evaluation results will be presented for each dataset size change, culminating in a comprehensive comparison between the proposed method and the compared method across varying dataset sizes.

A. IMDB

The initial dataset under scrutiny is IMDB, comprising 50,000 entries. Table 2 presents the evaluation criteria results for both the training and testing portions of this dataset.

The results unequivocally demonstrate the superior performance of the proposed technique across all five assessment criteria compared to the baseline method. Notably, the suggested technique exhibits a lower error rate in the MSE criterion compared to the base method. Moreover, it consistently outperforms in terms of F-Score, Precision, Recall, and Accuracy.

Furthermore, the proposed technique showcases remarkable consistency across varying assessment scenarios, including changes in the size of the training dataset. This underscores the adaptability of the suggested strategy to fluctuations in training dataset size. Conversely, the baseline

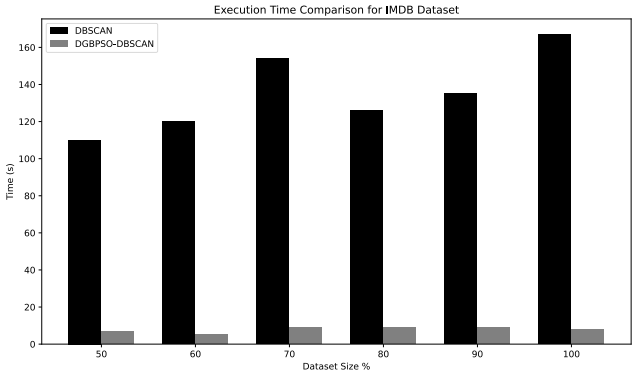

FIGURE 10. Comparison in execution time between the DBSCAN and the proposed DGBPSO-DBSCAN for the IMDB dataset.

method displays fluctuating performance with changes in the training dataset size, particularly evident in the Precision and Recall criteria. The results are presented in figure 9.

The proposed method in this research significantly reduces time complexity, as evidenced by the results presented in the table above. It's noteworthy that the execution time of the traditional algorithm consistently surpasses that of the proposed algorithm. When employing the DBSCAN algorithm, execution times range from 110 to 167 seconds, whereas with our DGBPSO-DBSCAN algorithm, execution

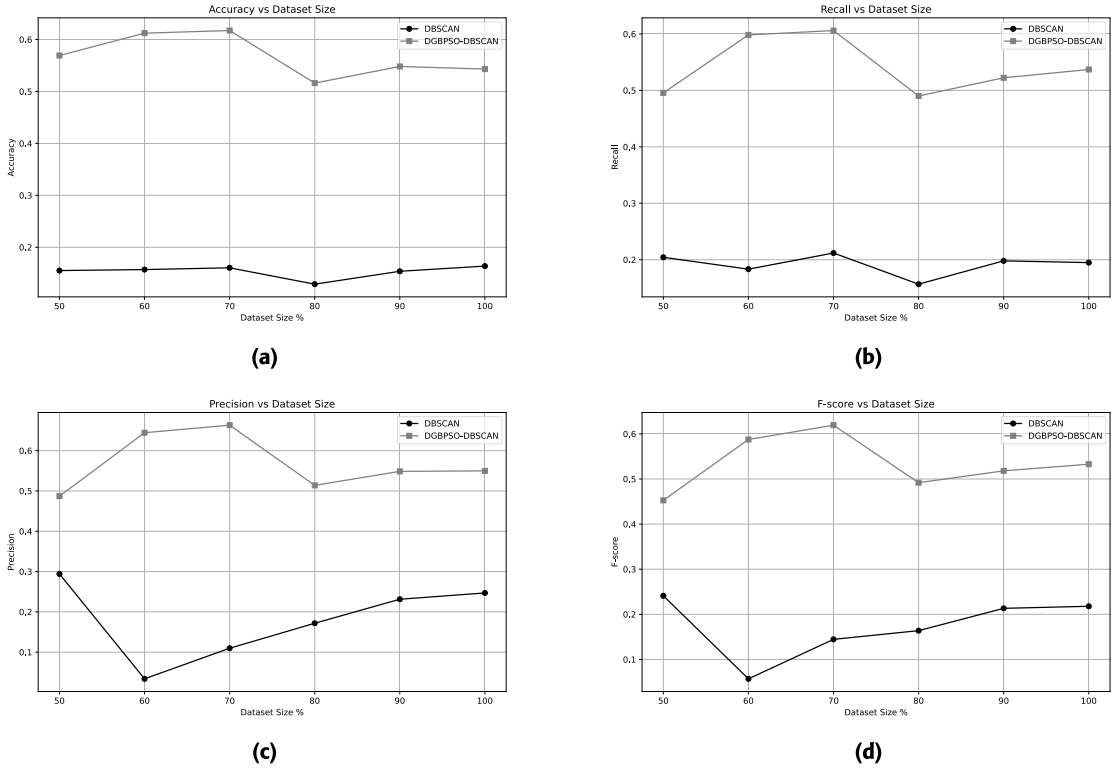

FIGURE 11. The proposed DGBPSO-DBSCAN results in terms of accuracy, recall, precision, and f1-measure compared to traditional DBSCAN algorithm in 20NewsGroup dataset.

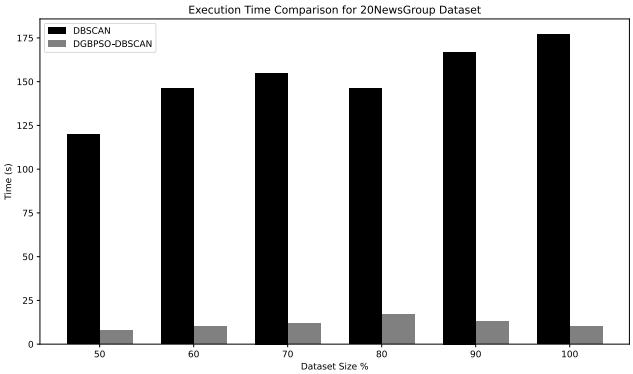

FIGURE 12. Comparison in execution time between the DBSCAN and the proposed DGBPSO-DBSCAN for the 20NewsGroup dataset.

times range from a mere 5 to 9 seconds, as displayed in Figure 10. This achievement aligns precisely with the primary goal of our study: to mitigate the time complexity of the clustering algorithm.

B. 20NewsGroup

The second dataset under consideration is the 20NewsGroup dataset. This dataset, similar to the IMDB dataset, is a text-processing dataset containing 18828 records. The evaluation results of this dataset can be seen in Table 4 based on the evaluation criteria.

The results presented in this dataset and the evaluation criteria demonstrate significant superiority of the proposed DGBPSO-DBSCAN over the primary method. However, unlike the IMDB dataset, both the proposed and basic methods could not demonstrate the necessary flexibility to adapt to changes in the dataset size. In fact, in both methods, the results fluctuate significantly with changes in the dataset size. This indicates that the compared approaches lack flexibility in handling changes in dataset size. The results are represented in figure 11.

Figure 12 represents the difference in the execution time between the proposed DGBPSO-DBSCAN and traditional DBSCAN algorithm for 20NewsGroup dataset.

C. Iris

The Iris dataset is the final dataset analyzed. Unlike the previous two datasets, the Iris dataset does not contain text. It comprises information on various flowers, and the goal of this data collection is to study the effect of text in the proposed technique. Similar to the previous datasets, the outcomes of this dataset are presented using five assessment criteria. The results are shown in Table 5 below.

The outcomes of this dataset indicate that the proposed strategy significantly outperforms the compared method. However, these results also suggest that both the proposed approach and the compared method are equally adaptable to changes in the size of the dataset. Consequently, as the size of the dataset varies, both techniques yield somewhat different outcomes, as evidenced by numbers 1.17 through 1.21. Moreover, the table above demonstrates a significant

TABLE 6. clustering algorithms, datasets, evaluation measures, and results.

Clustering algorithm	Dataset	Evaluation measures	RESULTS	Citation
I-DBSCAN	pen digits data from UCI machine learning	Rand-Index	DBSCAN takes 1563.7 seconds whereas I-DBSCAN takes only 65.7 seconds	[29]
ST-DBSCAN	Satellite data	density factor		[30]
Rough-DBSCAN	standard datasets & synthetic datasets	Rand-Index	Rough-DBSCAN's execution time 23 S whereas DBSCAN takes 101.82 S	[31]
MR-DBSCAN	GPS location records	speedup	The execution time is reduced by approximately 20% when using the proposed algorithm	[32]
PDSDBSCAN	IBM synthetic data	speedup	speedups up (maximum 4.82%, minimum 0.21%, and average 1.25%)	[33]
BDE-DBSCAN	2D artificial data sets	Purity	Purity (92-99%)	[34]
DMDBSCAN	peatland hotspots in Sumatera	running time	1.72 seconds to 32.29	[35]
PSODBSCAN	Re0, re1, 20news, tr41 and La1	Purity	DBSCAN=0.42, PSODBSCAN=0.82	[36]
GA-DBSCANMR	Baidu Encyclopedia data	Accuracy	96%	[19]
PSO BDBSCAN	artificial datasets	purity	97%	[9]
IPSO, PCPSO, SPSO	Iris, UCI machine	Accuracy	0.92%	[10]
DBSCAN++	real datasets	RAND index	65%	[37]
PSODBSCAN	MATLAS dataset	Recall, Precision, f-measure	97%	[38]
I-DBSCAN	OLAP, WebKB, SMS	Accuracy	93%	[39]
PODCC	WebKB	CD index value	90%	[13]
IMVO2-DBSCAN	Seed, Iris subset	Accuracy	93%	[18]
GIDBSCAN	Wikipedia, IMDB, 20newsgroups	Silhouette and Metrics Ave.	0.069	[40]
BSA-DBSCAN	Iris dataset	Purity	77%	[41]
DBSCAN based CM and ED	artificial data set	CM & ED	CM=1.00, ED=22%	[42]
PSO based Optimization of DBSCAN Algorithm	official road accident database of Hungary	sliding window method	77%	[43]
PSO and the adaptive DBSCAN	wind farms	Root Mean Squarerror (RMSE)	1.98	[44]
DBScan-based task scheduling algorithm	NASA iPSC workload log file, 4 data centers	scaling task & cloudsim simulation framework	49%	[45]
DGBPSO-DBSCAN	Ours	scaling task & Accuracy	96%	

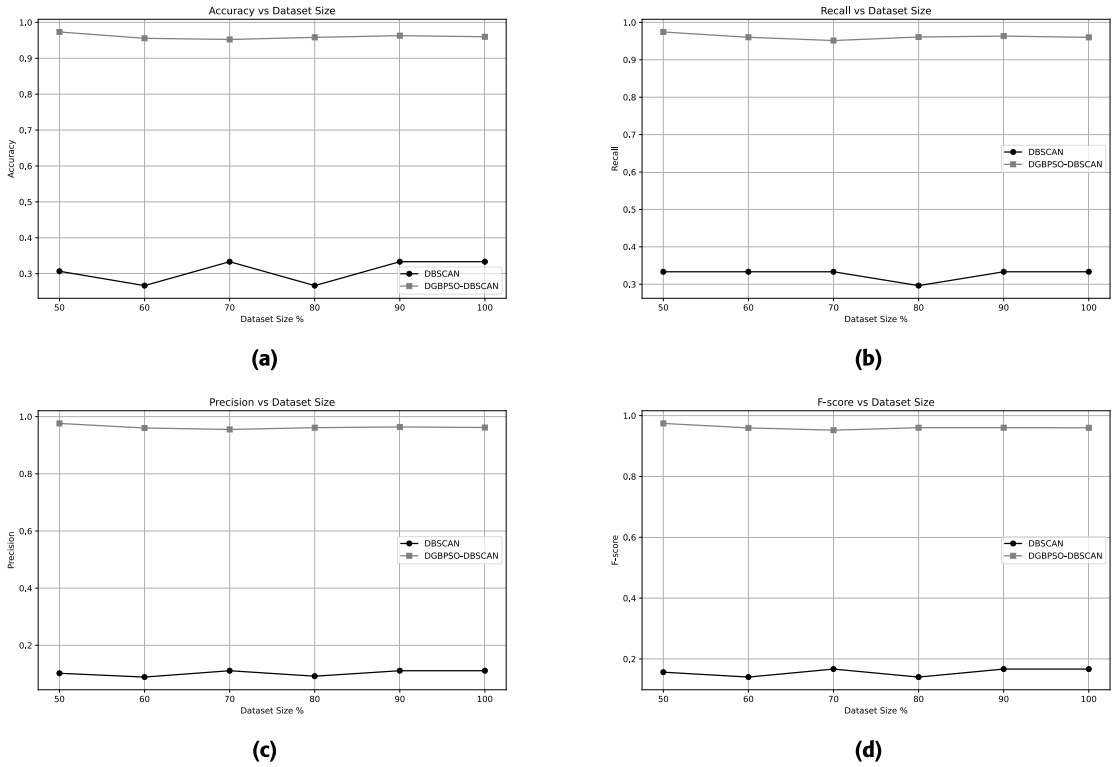

FIGURE 13. The proposed DGBPSO-DBSCAN results in terms of accuracy, recall, precision, and f1-measure compared to traditional DBSCAN algorithm in Iris dataset.

variation and discrepancy in the time required to implement the algorithm, with the algorithm devised for this study outperforming the conventional technique. The results are represented in figure 13.

Using the third type of dataset, Iris, the complexity time of the algorithm was reduced from 132 to only 5 seconds, as seen in the graph 14.

DGBDC significantly reduces document size and eliminates redundant and irrelevant features, effectively reducing data dimensionality and creating a conducive environment for the DBSCAN algorithm. The primary aim of this study is to enhance algorithm efficiency by alleviating high-dimensional data challenges. Our results demonstrate that the text representation method utilized in this study (DGBDC) effectively addresses high dimensionality by filtering out irrelevant document features. Consequently, when applying the DBSCAN algorithm to features derived from DGBDC, algorithm efficiency improves due to reduced computation on only the most relevant features. Additionally, DGBDC eliminates redundant and irrelevant text features, thereby mitigating the time complexity associated with the DBSCAN algorithm—a critical issue that it commonly faces.

D. QUALITATIVE COMPARISON

Regarding the achieved results, table 6 clarifies the qualitative description of the results compared to our proposed method.

E. CONCLUSION AND FUTURE CHALLENGES

DGBDC significantly reduces document size and eliminates redundant and irrelevant features, effectively reducing data

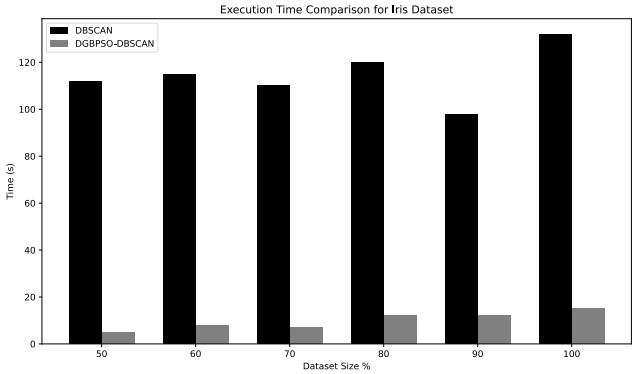

FIGURE 14. Comparison in execution time between the DBSCAN and the proposed DGBPSO-DBSCAN for the Iris dataset.

dimensionality and creating a conducive environment for the DBSCAN algorithm. The primary aim of this study is to enhance algorithm efficiency by alleviating high-dimensional data challenges. Our results demonstrate that the text representation method utilized in this study (DGBDC) effectively addresses high dimensionality by filtering out irrelevant document features. Consequently, when applying the DBSCAN algorithm to features derived from DGBDC, algorithm efficiency improves due to reduced computation on only the most relevant features. Additionally, DGBDC eliminates redundant and irrelevant text features, thereby mitigating the time complexity associated with the DBSCAN algorithm—a critical issue it commonly faces.

REFERENCES

[1] E. Ash and S. Hansen, "Text algorithms in economics," *Annu. Rev. Econ.*, vol. 15, no. 1, pp. 659–688, Sep. 2023.

[2] L. Cai, T. Lai, L. Wang, Y. Zhou, and Y. Xiong, "Graph convolutional network combining node similarity association and layer attention for personalized recommendation," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105981.

[3] S. Aminizadeh, A. Heidari, S. Toumaj, M. Darbandi, N. J. Navimipour, M. Rezaei, S. Talebi, P. Azad, and M. Unal, "The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things," *Comput. Methods Programs Biomed.*, vol. 241, Nov. 2023, Art. no. 107745.

[4] M. S. Ali and G. V. Dahake, "Unsupervised learning-based news aggregation: A comparative study of different embedding and clustering techniques," *J. Tech. Educ.*, vol. 46, no. 3, p. 18, 2023.

[5] H. Wu, C. Huang, and S. Deng, "Improving aspect-based sentiment analysis with knowledge-aware dependency graph network," *Inf. Fusion*, vol. 92, pp. 289–299, Apr. 2023.

[6] J. S. Colvett, B. J. Weidler, and J. M. Bugg, "Revealing object-based cognitive control in a moving object paradigm," *J. Experim. Psychol., Human Perception Perform.*, vol. 49, no. 11, pp. 1467–1484, Nov. 2023.

[7] D. Pinto, H. Gómez-Adorno, D. Vilariño, and V. K. Singh, "A graph-based multi-level linguistic representation for document understanding," *Pattern Recognit. Lett.*, vol. 41, pp. 93–102, May 2014.

[8] T. Bezdan, C. Stoean, A. A. Naamany, N. Bacanin, T. A. Rashid, M. Zivkovic, and K. Venkatachalam, "Hybrid fruit-fly optimization algorithm with K-means for text document clustering," *Mathematics*, vol. 9, no. 16, p. 1929, Aug. 2021.

[9] A. Wadhwa and M. K. Thakur, "Modified DBSCAN using particle swarm optimization for spatial hotspot identification," in *Proc. 11th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2018, pp. 1–3.

[10] A. Banerjee and I. Abu-Mahfouz, "Evolutionary clustering algorithms for relational data," *Proc. Comput. Sci.*, vol. 140, pp. 276–283, Jan. 2018.

[11] T. Wang, S. Guo, M. He, and Y. Li, "Research on community discovery method of complex network based on density peak clustering and its application," in *Proc. 3rd Int. Conf. Digit. Soc. Intell. Syst. (DSInS)*, Nov. 2023, pp. 265–269.

[12] L.-T. Li, Z.-Y. Xiong, Q.-Z. Dai, Y.-F. Zha, Y.-F. Zhang, and J.-P. Dan, "A novel graph-based clustering method using noise cutting," *Inf. Syst.*, vol. 91, Jul. 2020, Art. no. 101504.

[13] C. Guan, K. K. F. Yuen, and F. Coenen, "Particle swarm optimized density-based clustering and classification: Supervised and unsupervised learning approaches," *Swarm Evol. Comput.*, vol. 44, pp. 876–896, Feb. 2019.

[14] Q. Bai, "Analysis of particle swarm optimization algorithm," *Comput. Inf. Sci.*, vol. 3, no. 1, p. 180, Jan. 2010.

[15] A. Baresel, H. Sthamer, and M. Schmidt, "Fitness function design to improve evolutionary structural testing," in *Proc. 4th Annu. Conf. Genetic Evol. Comput.*, 2002, pp. 1329–1336.

[16] D. Garg and P. Garg, "Basis path testing using SGA & HGA with ExLB fitness function," *Proc. Comput. Sci.*, vol. 70, pp. 593–602, Jan. 2015.

[17] M. Alshraideh and L. Bottaci, "Search-based software test data generation for string data using program-specific search operators," *Softw. Test., Verification Rel.*, vol. 16, no. 3, pp. 175–203, Sep. 2006.

[18] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A new DBSCAN parameters determination method based on improved MVO," *IEEE Access*, vol. 7, pp. 104085–104095, 2019.

[19] X. Hu, L. Liu, N. Qiu, D. Yang, and M. Li, "A MapReduce-based improvement algorithm for DBSCAN," *J. Algorithms Comput. Technol.*, vol. 12, no. 1, pp. 53–61, Mar. 2018.

[20] W. Song, Y. Qiao, S. C. Park, and X. Qian, "A hybrid evolutionary computation approach with its application for optimizing text document clustering," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2517–2524, Apr. 2015.

[21] M. de Jonge, L. C. L. Kats, E. Visser, and E. Söderberg, "Natural and flexible error recovery for generated modular language environments," in *Proc. ACM Trans. Program. Lang. Syst. (TOPLAS)*. New York, NY, USA: ACM, 2012, vol. 34, no. 4, pp. 1–50.

[22] A. Huang, "Similarity measures for text document clustering," in *Proc. 6th New Zealand Comput. Sci. Res. Student Conf.*, vol. 4, Christchurch, New Zealand, 2008, pp. 9–56.

[23] R. L. Kumar, R. Subramanian, and S. Karthik, "A novel approach to improve network validity using various soft computing techniques," *J. Intell. Fuzzy Syst.*, vol. 43, no. 6, pp. 7937–7948, Nov. 2022.

[24] J. Shanthini, P. Punitha, and S. Karthik, "Improvisation of node mobility using cluster routing-based group adaptive in MANET," *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2619–2636, 2023.

[25] A. Luo, S. Gao, and Y. Xu, "Deep semantic match model for entity linking using knowledge graph and text," *Proc. Comput. Sci.*, vol. 129, pp. 110–114, Jan. 2018.

[26] D. N. Thi, V. D. Hieu, and N. V. Ha, "A technique for generating test data using genetic algorithm," in *Proc. Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2016, pp. 67–73.

[27] J. Wegener, A. Baresel, and H. Sthamer, "Evolutionary test environment for automatic structural testing," *Inf. Softw. Technol.*, vol. 43, no. 14, pp. 841–854, Dec. 2001.

[28] Y. Chen, Y. Zhong, T. Shi, and J. Liu, "Comparison of two fitness functions for GA-based path-oriented test data generation," in *Proc. 5th Int. Conf. Natural Comput.*, vol. 4, Aug. 2009, pp. 177–181.

[29] P. Viswanath and R. Pinkesh, "L-DBSCAN: A fast hybrid density based clustering method," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 912–915.

[30] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, Jan. 2007.

[31] P. Viswanath and V. Suresh Babu, "Rough-DBSCAN: A fast hybrid density based clustering method for large data sets," *Pattern Recognit. Lett.*, vol. 30, no. 16, pp. 1477–1488, Dec. 2009.

[32] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, and J. Fan, "MR-DBSCAN: An efficient parallel density-based clustering algorithm using MapReduce," in *Proc. IEEE 17th Int. Conf. Parallel Distrib. Syst.*, Dec. 2011, pp. 473–480.

[33] M. M. A. Patwary, D. Palsetia, A. Agrawal, W.-K. Liao, F. Manne, and A. Choudhary, "A new scalable parallel DBSCAN algorithm using the disjoint-set data structure," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2012, pp. 1–11.

[34] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution," *Int. J. Comput. Appl.*, vol. 91, no. 7, pp. 1–11, Apr. 2014.
[35] N. Rahmah and I. S. Sitanggang, "Determination of optimal epsilon (Eps) value on DBSCAN algorithm to clustering data on peatland hotspots in Sumatra," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 31, Jan. 2016, Art. no. 012012.
[36] R. Janani and S. Vijayarani, "A hybrid approach for improving text document clustering," *Int. J. Eng. Res. Comput. Sci. Eng. (IJERCSE)*, vol. 4, pp. 2320–2394, Apr. 2320.
[37] J. Jang and H. Jiang, "DBSCAN++: Towards fast and scalable density clustering," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3019–3029.
[38] J. Huang, Y. Xing, H. You, L. Qin, J. Tian, and J. Ma, "Particle swarm optimization-based noise filtering algorithm for photon cloud data in forest area," *Remote Sens.*, vol. 11, no. 8, p. 980, Apr. 2019, doi: 10.3390/rs11080980.
[39] Neha and P. Verma, "I-DBSCAN algorithm with PSO for density based clustering," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 6, pp. 627–632, Jun. 2019.
[40] S. M. Mohammed, K. Jacksi, and S. R. M. Zeebaree, "Glove word embedding and DBSCAN algorithms for semantic document clustering," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, Dec. 2020, pp. 1–6.
[41] L. Wang, H. Wang, X. Han, and W. Zhou, "A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm," *Comput. Commun.*, vol. 174, pp. 205–214, Jun. 2021.
[42] K. Giri and T. K. Biswas, "Determining optimal epsilon (eps) on DBSCAN using empty circles," in *Proc. Int. Conf. Artif. Intell. Sustain. Eng.*, vol. 1. Singapore: Springer, Apr. 2022, pp. 265–275.
[43] S. Szénási, M. Sipos, and P. Mogyorósi, "PSO based optimization of DBSCAN algorithm parameters for road accident blackspot localization," in *Proc. IEEE 10th Jubilee Int. Conf. Comput. Cybern. Cyber-Med. Syst. (ICCC)*, Jul. 2022, pp. 000043–000048.
[44] N. Zhou, H. Ma, J. Chen, Q. Fang, Z. Jiang, and C. Li, "Equivalent modeling of LVRT characteristics for centralized DFIG wind farms based on PSO and DBSCAN," *Energies*, vol. 16, no. 6, p. 2551, Mar. 2023.
[45] S. M. F. D. S. Mustapha and P. Gupta, "DBSCAN inspired task scheduling algorithm for cloud infrastructure," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 32–39, Jan. 2024.

ASMA KHAZAAL ABDULSAHIB received the bachelor's degree in software engineering from the Al-Rafidain College, Baghdad, Iraq, in 2000, and the M.Sc. degree in information technology (IT) from Utara University Malaysia (UUM), in 2014. She is currently pursuing the Ph.D. degree in artificial intelligence with the University of Tabriz. She is the Supervisor of the preliminary studies laboratories and responsible for the Systems and Software Division, University of Baghdad. She teaches undergraduate students. She supervises a number of research studies for graduate students, and has evaluated a number of research articles for journals in ISI as Neural Computing. She has several articles published in Scopus journals. Her research interests include data mining, machine learning, clustering algorithms, and IT applications.

M. A. BALAFAR received the Ph.D. degree in IT engineering from Universiti Putra Malaysia, Seri Kembangan, Malaysia, in 2010, following the completion of his Ph.D. studies, he joined the University of Tabriz, Tabriz, Iran.

He is an integral part of the academic community with the University of Tabriz. He is a Distinguished academic. Currently, he is a Full Professor with the Faculty of Electrical and Computer Engineering (ECE), University of Tabriz. He has earned him a reputation as a Leading Expert and a valuable contributor to the advancement of artificial intelligence research. With a vast research experience, he has made significant contributions to the field of computer science and artificial intelligence. He has authored, co-authored, and reviewed numerous articles across prestigious scientific communities, including renowned publishers such as Springer, Elsevier, IET, and IEEE. His publications showcase his expertise in various areas, including machine learning, data mining, deep learning, computer vision, expert systems, and evolutionary computing. Through his influential research, he continues to shape the field of computer science and advance the frontiers of knowledge in machine learning, data mining, computer vision, and related areas.

ARYAZ BARADARANI (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Windsor, ON, Canada, in 2012.

He is currently a Principal Scientist and a Research and Development Project Manager with Tessonics Inc., Windsor. His commitment to advancing the field of electrical engineering and his dedication to professional excellence. He has also served as a top peer reviewer for various esteemed journals, conferences, and symposiums. His primary research interests revolve around feature extraction and diagnostic imaging, where he has made significant contributions to the field.

Prof. Baradarani is a recipient of several prestigious awards, including the Endowment Scholar Award, the Queen Elizabeth II Graduate Scholarship in Science and Technology, the University of Windsor President's Excellence Scholarship, the University of Windsor International Graduate Excellence Scholarship (IGES), and the 3M Company Bursary Award. In addition to his professional roles, he has been an active contributor to technical program committees and international advisory committees.