Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 11, November 2025, pg.66 – 84

An Adaptive Quantum Bacterial Colony Optimization Framework for an Efficient Text Clustering Approach

[1]S Dhanalakshmi; [2]S Sathiyabama; [3]D Ayyamuthukumar

[1]Department of Computer Science, Thiruvalluvar Government Arts College, Rasipuram, Tamil Nadu, India
[2]Department of Computer Science, Government Arts and Science College, Kangeyam, India
[3]Department of Artificial Intelligence and Data Science, Muthayammal College of Engineering, Rasipuram, India
E-Mail: [1] dhanalakshmisubramaniam79@gmail.com; [2] drsathyaabama@gmail.com; [3] ukdkumar@gmail.com

DOI: https://doi.org/10.47760/ijcsmc.2025.v14i11.005

Abstract: Text document clustering is a task that seeks to cluster together related documents based on content similarity. Clustering helps to improve information diversity retrieval, topic spotting, and sentiment analysis by ordering unstructured text and dividing it into information structures. To address the issue of text-based work and its optimization through more efficient and accurate text classification and clustering, this paper offers a more advanced text classification and clustering framework that incorporates the Adaptive Quantum Bacterial Colony Optimization (AQBCO) algorithm and deep word embedding models, such as Word2Vec, TF-IDF, and FastText. AQBCO presents two innovations: adaptive chemotaxis, which produces the movement of bacteria dynamically controlled by the optimal solution, and quantum-inspired position updating, which increases the ability to search the globe and eliminates premature convergence. The framework was tested using benchmark datasets- BBC, SMS, and DMOZ, and was compared with the available optimization-based learning methods, including Improved BCO (IBCO), standard BCO, GQPSO, IPSO, PSO, GA, and k-means. According to the experimental findings, AQBCO always performs better on Accuracy, Precision, Recall, and F-Measure as compared to other models. The highest accuracy of 97.38% on BBC, 96.79% on SMS, and 93.68% on DMOZ datasets allowed AQBCO to use FastText embeddings. The suggested adaptive and quantum systems preserve exploration-exploitation balance, enhancing the speed of convergence and resiliency. These findings make AQBCO a scalable, stable, and efficient method of text classification as well as document clustering.

Keywords: Natural language processing, Text document clustering, Bacterial colony optimization, quantum computing, adaptive mechanism

I. INTRODUCTION

In this digital age, with the rapid growth of textual information through several online sources (e.g., news portals, social media, academic repositories, and web directories), the classification of texts in natural language processing (NLP) has been a pressing and challenging concern [1]. The process of automatically applying predefined categories to text documents is known as text classification and is used in many applications of the real world, such as spam filtering, sentiment analysis, news categorization, and information retrieval. As the big data wave hits, scalable and efficient text classification algorithms are now essential in processing high-dimensional and semantically rich textual data [2]. The classical machine learning algorithms have been extensively used to classify texts. These techniques, however, are based on the intensive use of handcrafted features and bag-of-words representations, which are unsuitable in reflecting the semantic and contextual relationships of words. Therefore, these types of models are limited to dealing with ambiguity, polysemy, and linguistic differences in textual data[3].

The word embedding systems, in particular, Word2Vec, Term Frequency-Inverse Document Frequency (TF-IDF), and FastText, have become effective feature representation systems to counter these limitations. These embeddings convert textual information into continuous and dense spaces, which encode syntactic and semantic information, enhancing the interpretation of models and understanding their context. Although the feature representation has advanced, the overall performance of classification models remains as subject to the optimization of features, hyperparameter optimization, and training strategy. The above has resulted in the combination of machine learning models with metaheuristic optimization algorithms in order to obtain enhanced optimization and generalization. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Gravitational Search Algorithms (GSA), and Bacterial Colony Optimization (BCO) are some of the metaheuristic algorithms that have proven to have great potential in solving complex optimization problems in a wide range of application fields [4]. The algorithms are based on natural phenomena and biological mechanisms that imitate swarm intelligence, collective behavior, and evolutionary adaptation. Nevertheless, the current optimization-based strategies also have several notable weaknesses, which impede their use in large-scale text classification problems, such as premature convergence. Algorithms like PSO and GA tend to get stuck in local minima during optimization because they do not provide a great balance between exploration (global search) and exploitation (local refinement)[3]. In traditional algorithms, step sizes and search parameters are fixed and restrict flexibility in dynamic and complicated search spaces. Numerous algorithms have sluggish convergence, particularly in the case of large feature vectors based on large text collections. Lack of sufficient diversity in the population results in repeated search and poor solutions.

In an attempt to resolve such difficulties, this study introduces a progressive optimization model called AQBCO. The AQBCO algorithm is based on the foraging mechanism of bacteria and flows with the principles of quantum computing. It provides a mechanism of adaptive chemotaxis step size to dynamically change the movement of bacteria in the search process. This change allows greater responsiveness to the fitness landscape, which is why the algorithm is able to explore new areas without stagnating too quickly. Moreover, a quantum-inspired search strategy increases the variety of candidate solutions by allowing probabilistic jumps in the solution space, giving it a better balance between exploration and exploitation. The suggested classification system with AQBCO is a combination of the algorithm and the potent word embedding methods (Word2Vec, TF-IDF, and FastText) to guarantee the best feature selection and sound performance on various datasets. Benchmark datasets used in the extensive experiments include BBC, WebKB, SMS, and DMOZ, which are distinguished by the difference in document length, domains, and the disparity of classes. The relative analysis of the AQBCO with the existing optimization techniques that are Improved BCO (IBCO), Quantum-Behaved PSO (GQPSO), Improved PSO (IPSO), GA, and K-Means shows that AQBCO performs better in terms of accuracy, precision, recall, and F-measure. The objective of the study is to develop a very flexible, efficient, and scalable optimizer algorithm that improves the accuracy of the text classification by using intelligent parameter optimization and hybrid feature optimization. It is not only meant to achieve better performance in classification but also to guarantee stability, quicker convergence, and flexibility in heterogeneous text datasets. The major contributions of this study can be outlined as follows:

- An innovative optimization algorithm that combines adaptive chemotaxis with quantum behavior to balance exploration and exploitation effectively.
- A dynamic process leading to the adjustment of bacterial movement depending on fitness enhancement, which allows flexible and efficient convergence.
- Three well-known text representation models have been used to analyze how AQBCO can be applied to different feature spaces, including Word2vec, TF-IDF, and FastText. Large-scale testing on four benchmark datasets in terms of BBC, WebKB, SMS, and DMOZ to assess the strength of the suggested method and its generalization.
- Compared to the existing metaheuristic algorithms like IBCO, BCO, GQPSO, IPSO, PSO, GA, and K-Means, it is clear that AQBCO outperforms its rivals.

- Evaluation of the algorithmic performance on the different dataset sizes, dimensions, and feature extraction methods, demonstrating the ability of AQBCO to be used in massive NLP applications.
- According to the experimental findings, it was found that AQBCO has the best average performance in classification, with the highest recorded accuracy of up to 2-5% and F-measure of 3-4% improvement over other methods.

The rest of this paper will be structured in the following way: In Section 2, the corresponding literature and reviews of the existing text classification and optimization methods are provided. In Section 3, the research methods that are used in this paper, such as BCO, Quantum computing, adaptive mechanism, and the proposed AQBCO model, are explained. Section 4 remarks on the experimental design, data, and performance measures with results and discussion. Section 5 gives a conclusion to the paper with a summary of the findings, the benefits of the proposed method, and the recommendation of future improvement, like integration of hybrid deep learning and evaluation with huge text datasets.

II. RELATED WORKS

The rapid growth of unstructured textual information on the internet, particularly in the case of social media and digital publishing, has enhanced the need to have practical text summarization and document clustering methods. Semantic complexity and redundancy are normally not addressed under traditional methods. Therefore, recent studies have been integrating deep learning and meta-heuristic optimization in order to increase the accuracy, consistency, and scalability. Vinitha et al. [5] came up with a machine learning-assisted framework of text summarization incorporating Diverse Beam Search-Based Maximum Mutual Information (DBSMMI) and Ant Colony Optimization (ACO)-optimised Deep Belief Network (DBNTS). Their model, which combines semantic similarity with the k-means clustering, delivered a 92.35% and 90.29% accuracy and precision, respectively, compared to standard summarizers. In the same manner, Dhanalakshmi et al. [6] proposed a BCO-based clustering algorithm, which alleviated premature convergence and local minima issues that are prevalent in Swarm Intelligence (SI) algorithms, leading to faster convergence and better text cluster quality. Dabjan and Kurdy [7], in another approach based on optimization, had made an Artificial Bee Colony Optimization (ABCO) framework for the efficient web page classification of Arabic language. In their research, they focused on minimizing the computation cost and enhancing accuracy in online classification. To better clustering, Periyasamy et al. [8] refined a hybrid Bacterial Foraging Optimization (BMO) based on Rough Set Analysis (RSA) with Rough Set Analysis TF-IDF features were used to reduce uncertainty and enhance classification performance. In the meantime, Widiartha et al. [9] introduced the Multi-Document Summarization model that relies on Tuna Swarm Optimization (TSO) and Markov Clustering in the context of the Indonesian news corpora, which is able to produce high-quality summaries with the help of optimized feature selection. A number of researchers have studied swarm-based optimizers in document clustering. Al-Betar et al. [10] applied Bare-Bones Salp Swarm Algorithm (BBSSA) in clustering, which is shown to be superior in exploring than typical PSO. Gurusamy et al. [11] proposed a hybrid optimized Auto-encoded LSTM model for text summarization, utilizing the Whale Optimization Algorithm (WOA) to optimize hyperparameters, resulting in better semantic coverage and less redundancy in text summaries.

Similarly, Azarshab et al. [12] integrated Teaching-Learning-Based Optimization (TLBO), Gray Wolf Optimization (GWO), as well as Genetic Algorithms (GA) in the selection of features, which leads to improved cluster purity and reduced computational complexity. Innovations were further outlined by Abuain et al. [13] who initiated the link-based Salp Swarm Algorithm (LSSA) in document clustering, which produced better scores on cohesion as well as separation. Dodda et al. [14] used Chaotic Northern Goshawk Optimization (CNGO) as it was combined with k-means to ensure faster convergence in clustering activities. Equally, Radomirović et al.) [15] proposed an Improved Meta-heuristic Tuned k-means (IMT-k-means) model, in which the centroid dynamically changes using an adaptive optimization mechanism to improve clustering strength. Summarization has also been applied with the use of optimization. A Multi-Document summarization framework, Muddada et al. [16] introduced Multi-Objective Ant Colony Optimization (MOACO), which maximized the informativeness and reduction of redundancy. Rautararay et al. [17] added a further step in this direction, providing a hybrid Cuckoo Search Optimization (CSO) and Harris Hawks Optimization (HHO) together with Restricted Boltzmann Machine (RBM) to enhance summary coherence and decrease computation time. Emambocus et al. [18] used Salp Swarm Optimization (SSO) for K-means clustering based on the recommendation of movies and obtained better recommendation diversity and accuracy. A semantic similarity-based feature selection and clustering algorithm for novel text retrieval was suggested by Sun [19] and the hybrid similarity metrics were found to be useful. A hybrid K-Means++/Particle Swarm Optimization (PSO) feature extraction and feature clustering framework was proposed by Hassan et al.) [20] , resulting in a higher level of intra-cluster similarity and scalability to large datasets. Naderi et al. [21] The Multi-Objective Firefly Differentiated Jaya (MFDJ) algorithm provides optimization of clustering measures, including compactness and separation, which was highly improved as compared to traditional algorithms.

Rautaray et al. [3] also developed multi-document data in a Deep Learning-based summarizer system with hybrid CSO-HHO in place of the synergy between optimization and deep architecture. Mousa et al. [22]

created a hierarchical Arabic document classification, a multi-tag-based model that combines text mining and ML with a high level of accuracy in a multi-label setting. Finally, a hybrid NSGA-II and XGBoost model to classify toxic tweets was introduced by Mosa et al. [23] which is a unification of evolutionary optimization with ensemble learning to provide better feature selection and interpretability. Even though there are a lot of optimization-based models, which have increased the performance of text summarization and clustering, the majority of existing literature does not focus on issues such as real-time flexibility and scalability, or even on high-performance computation with high-dimensional text data. Moreover, a great deal of hybrid optimization methods, although useful, continue to have premature convergence and overfitting in complex semantic space. Table 1 shows the comparisons of document clustering methods with their contributions and limitations.

TABLE I
COMPARISONS OF DOCUMENT CLUSTERING METHODS

Author(s) & Year	Method	Focus Area	Dataset / Domain	Contributions	Limitations
Vinitha et al. (2024) [5]	DBSMMI + ACO-optimized DBNTS	Text summarization & clustering	Web / Social media text	Improved semantic grouping using ACO and DBN	High computational complexity
Dhanalakshmi et al. (2023) [6]	BCO	Text document clustering	Benchmark text corpora	Overcomes local optima issues	Requires parameter tuning
Dabjan & Kurdy (2024) [7]	ABCO	Arabic web page classification	Arabic text dataset	Reduces computational load	Limited to a single language
Periyasamy et al. (2024) [8]	Hybrid BFO + Rough Set Analysis (RSA)	Document clustering	Text corpus	Reduces uncertainty and improves stability	Slow convergence on large data
Widiartha et al. (2025) [9]	TSO + Markov Clustering	Multi-document summarization	Indonesian news	Enhances summary quality	Domain-specific approach
Al-Betar et al. (2023) [10]	BBSSA	Document clustering	Text datasets	Better exploration–exploitation	May converge slowly
Gurusamy et al. (2024) [11]	Optimized Auto-encoded LSTM + WOA	Text summarization	English corpora	Semantic coherence and less redundancy	High training cost
Azarshab et al. (2024) [12]	TLBO-GWO-GA Hybrid	Feature selection for clustering	Text datasets	Strong feature optimization	Computationally intensive
Abuain et al. (2024) [13]	LSSA	Document clustering	English text	Improved cluster separation	Sensitive to initialization
Dodda et al. (2024) [14]	CNGO + K-means	Text clustering	Multi-domain	Faster convergence	Risk of instability
Radomirović et al. (2023) [15]	Improved Metaheuristic Tuned K-means	Text clustering	Benchmark data	Adaptive centroid optimization	Limited semantic integration
Muddada et al. (2024) [16]	MOACO	Multi-document summarization	News & reports	Balanced informativeness and redundancy	Parameter sensitive
Rautaray et al. (2023) [17]	Hybrid CSO + HHO + RBM	Text summarization	News data	Enhanced semantic summarization	High computational overhead
Emambocus et al. (2024) [18]	SSO-Optimized K-means	Clustering for movie recommendation	Movie datasets	Better recommendation diversity	Limited scalability
Sun (2024) [19]	Semantic Similarity + Feature Selection	Text retrieval	Multi-domain	Improved retrieval and clustering	High dimensionality
Hassan et al. (2025) [20]	Hybrid K-Means++ + PSO	Document clustering	Text corpora	Better intra-cluster similarity	Convergence issues on large data
Naderi et al. (2023) [21]	MFDJ	Text document clustering	Large text data	Optimizes multiple cluster metrics	Complex parameter control
Rautaray et al. (2025) [3]	Deep Learning + Hybrid CSO-HHO	Multi-document summarization	News dataset	Improved informativeness	Overfitting risk

Mousa et al. (2025) [22]	Multi-tag Hierarchical ML Model	Arabic document classification	Arabic datasets	Supports multi-label tags	Domain-specific
Mosa et al. (2025) [23]	Hybrid NSGA-II + XGBoost	Toxic tweet classification	Social media (Arabic)	Enhanced feature selection and interpretability	High model complexity

III. RESEARCH METHODS

A) *Preprocessing methods*

The text demands model pre-processing activities to form clusters during this epoch. Text clustering pre-processing includes tokenizing, stop word removal, stemming, item weighting, and document representation. The brief discussions of the pre-processing procedures are in the following subsection.

1) Tokenizing: The word tokenization is the procedure of dividing words into fragments or tokens that most likely would lose individual letters at the same time (punctuation). These tokens are normally linked to words or objects; however, one should differentiate between types/tokens. A token in a text is a pattern of characters that is used as an effective semantic unit. A sort consists of all tokens with the same letter chain.

2) Stop Words Removal: Stop words include words such as which, the, our, is, an, me, that, and important phrases that are used repeatedly in the text, and low-help words. Being often repeated, they should be better eliminated from the text, which would enhance the effectiveness of clustering methods. The stop-word list contains more than 500 words.

3) Stemming: A process involving the reduction of existing words to their stem or root is called stemming. The root morphological approach is different from the stem technique because both techniques utilize the same stem to define words, though this is not a root.

B) *Word embedding*

Word embedding is essentially used to convert textual data into numerical vectors that can be understood by machines to facilitate models to learn semantic connections between words and their contextual meanings. The three most popular methods of word representation, Word2Vec, TF-IDF, and FastText, are used in this research and compared to each other in order to determine how they influence the results in terms of classification.

1) Word2Vec: it is an embedding algorithm that is a neural network and is trained to predict word contexts in a given sentence. It is based on either the Continuous Bag-of-Words (CBOW) or Skip-Gram model to produce dense low-dimensional vectors. Word2Vec is a successful model that preserves semantic similarity i.e. words that are grouped in close contexts (e.g., teacher and professor) are represented as close to each other in the vector space. This property allows it to be very effective in deep learning models with downstream applications because it ensures that semantic linkages are maintained with a lower dimensionality of the data.

2) TF-IDF (Term Frequency -Inverse Document Frequency) represents a statistical method that measures the significance of each word in a document as compared to the overall corpus. In contrast to Word2Vec, TF-IDF fails to obtain the contextual semantics but rather emphasizes the relevance of words depending on frequency. Words that are common in one document yet are uncommon in other documents are assigned greater weights. Despite the sparseness of the representations generated by TF-IDF, the dimensionality, it continues to be popular because of its simplicity, interpretability, and its ability to work with traditional machine learning classifiers.

3) FastText was created by Facebook AI Research as an extension of Word2Vec, where subword data is provided as character n-grams. Words are encoded in the form of a collection of their character fragments, and the model can know the internal makeup of words. This method is quite useful in enhancing the performance of morphologically rich languages as well as out-of-vocabulary words. FastText thus captures both the syntactic and semantic attributes, hence it is more robust than Word2Vec in addressing the infrequent and misspelled words. Overall, the three embeddings are complementary to each other because they focus on various facets of the textual representation.

C) *BCO*

A new population-based method proposed by Niu and Wang (2012) was a method known as BCO [24]. Older bacterial methods, such as BFO [25] and BC [26] largely looked at a particular behavior of interest, but not at the shared behaviors of interest that SI examines. None of the colony members shares information, with each one finding food individually based on his/her own experience. As an example, the BFO wastes time on unsystematic chemotaxis and complicates the computing. A recent SI behavior approach, the BCO, was therefore introduced in order to solve the above-named problem with the help of simplifying the optimization process. The stages of making up the BCO include chemotaxis and communication, the elimination and reproduction, and the migration. The entire BCO process is made with the aid of chemotaxis and communication. By learning the population data, the bacteria are in a position to change their swimming and tumult patterns. The positions of the bacteria are updated with the help of a special chemotaxis and communication method. During the lifespan of bacteria, there are two types of chemotaxis: tumbling and swimming. A stochastic direction participates in the process of swimming when tumbling occurs. The combined

effects of all the turbulent directors that influence the search direction and efficient positions of every bacterium can be formulated as follows:

$$Position_i(T) = Position_i(T-1) + C(i) * [f_i.(G_{best} - Position_i(T-1)) + (1-f_i) * (P_{best_i} - Position_i(T-1)) + turb_i] \quad (1)$$

However, the swimming process does not have a turbulence director to orient the bacteria to an optimal state, which can be stated as follows:

$$Position_i(T) = Position_i(T-1) + C(i) * [f_i.(G_{best} - Position_i(T-1)) + (1-f_i) * (P_{best_i} - Position_i(T-1))] \quad (2)$$

$$C(i) = C_{min} + \left(\frac{Iter_{max} - Iter_j}{Iter_{max}}\right)(C_{max} - C_{min}) \quad (3)$$

Where, $turb_i$ is represents the direction of turbulence. f_i is between 0 and 1. $C(i)$ is the value of chemotaxis. P_{best} refers to the personal best. G_{best} refers to the global best. $Iter_{max}$ is the maximum iteration, whereas $Iter_j$ is the current iteration, respectively. Two communication strategies, which include individual contact and group exchange, have also been suggested to increase the effectiveness of the optimization process. One type of exchange model can only be chosen by each bacterium at any given time. Bacteria that outperform replicate to create the latest during the eradication and production stage, and weak bacteria are substituted. The bacterium exhibits high performance in terms of nutrient uptake, which is demonstrated by its high energy. Bacteria are also able to travel within a given range of search space under certain conditions in the migration phase. As a result, bacteria aim to find the latest nutrients with the help of the mentioned probabilities during the migration period.

D) Quantum techniques

Quantum computing is a branch of the larger field of computation, where computation is performed using postulates of quantum mechanics. Unlike in classical computation, where the information is represented as binary digits, quantum computation involves the use of quantum bits (qubits) in the representation and manipulation of information. The superposition principle allows qubits to exist in more than one state at a time, and therefore provides quantum computers with the ability to solve some computational problems at a speed that is orders of magnitude faster than that of classical machines [27]. A qubit state can be described as a two-dimensional complex Hilbert space that is usually called the Bloch sphere. A typical qubit model consists of two basis states, which are denoted by $|0\rangle$ and $|1\rangle$. As it was mentioned above, the qubit can be in a superposition between the states $|0\rangle$ and $|1\rangle$ and this state can be represented by the state vector, denoted by $|\Psi\rangle$ and it can be expressed as the following,

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (4)$$

The probability amplitudes of the computational basis states 0 and 1 are represented by the complex numbers α and β, respectively. The state of a qubit is ascertained by use of an observation protocol. After opening the observation, the qubit may be measured to be either 0 or to be measured to be 1. The probability of measuring the qubit in state |human| or state |human| is respectively provided by $|\alpha|^2$ and $|\beta|^2$and the coefficients α and β should be normalised.

$$|\alpha|^2 + |\beta|^2 = 1 \quad (5)$$

Therefore, a qubit can be stated as a pair of compound integers using the method $q = \left[\frac{\alpha}{\beta}\right]$. Similarly, a multi-qubit system with a n is the number of qubits that can be exposed as

$$Q = [q_1, q_2, \ldots . q_n] = \left[\frac{\alpha_1}{\beta_1} \middle| \frac{\alpha_2}{\beta_2} \middle| \frac{\ldots}{\ldots} \middle| \frac{\alpha_n}{\beta_n} \middle|\right] \quad (6)$$

The qubit can be transformed by using different quantum gates, including the NOT gate, controlled-NOT (CNOT) gate, Hadamard gate, and other transformations of the same nature. In the case of a qubit represented by q_i, the quantum angle may be represented as follows:

$$\theta_i = arctan\frac{\beta_i}{\alpha_i} \quad (7)$$

E) Adaptive quantum BCO

Any evolutionary and SI algorithms searching in the space of possible solutions of the problem will react differently to the way their parameters are tuned. Parameter configuration has a direct impact on the quality of the solution; thus, it is critical to know what parameter configuration provides the most competent solution. The optimization of such parameters at this stage turns into an optimization problem in the problem under optimization [28, 29]. BCO is a popular SI method of solving any optimization problem due to its global search. But the original BCO algorithm has poor convergence properties, and the search activity reduces as the search space dimensionality and complexity of the problem increase. The step-size parameter is the main parameter that controls the effectiveness of BCO approaches. The following problems are associated with the BCO with fixed step size: (i) in case of a very small step size, many chemotaxis steps have to be performed before the optimal location is reached, which causes the convergence rate to slow down [30-32]. Therefore, during a small number of iterations, the algorithm might fail to reach the optimal position. (ii) Bacteria can

quickly get to the optimum, but an extremely large step size reduces accuracy, as the bacteria will then overshoot in the next chemotaxis step. In population-based optimization schemes, both global and local exploitation need proper control in order to find the optimal solution. In order to balance the local and global searches, an adaptive step size is used in this study. This functionality is as follows [33].

$$A = \frac{f^i(X)}{f^i_{max}}(x_{max} - x_{min})rand \quad (8)$$

Where, $f^i(X)$ is the i^{th} bacterium's fitness function value. x_{max} and x_{min} are the maximum and minimum values of variables. Chemotaxis step length is also adjusted adaptively as the algorithm is being run to provide a balance between the local and global optimization. The adaptive step is changed as follows,

$$C(i)_{new} = C(i)\ \frac{Iter_{max} - Iter_j}{Iter_{max}}.A \quad (9)$$

In this paper, a dynamic update procedure on the step size of chemotaxis is applied to guarantee the convergence of the BCO algorithm.

F) Proposed adaptive quantum BCO

The step-size adaptive chemotaxis mechanism, which is part of the Bacterial Colony Optimization (BCO) algorithm, allows the organisms to regulate the magnitude of movements during the chemotactic events in response to a range of optimization requirements and bacterial performance metrics. This method uses a dynamic step-size, unlike the traditional algorithms, which use a fixed step-size, so that at the early stages of the exploration, large increments are used to explore large areas of a search space, and at the later stages of the exploitation phase, smaller increments are used to allow the local search to be refined in promising areas. In addition, the step-size is varied with respect to how the objective function fitness changes: as they reach a better fitness value at a new point, the step-size is slightly increased; otherwise, the step-size is reduced. With this adaptive modulation, a balance is successfully achieved between global exploration and local exploitation and hence, convergence is accelerated, reducing local optimum entrapment. This is further enhanced by the addition of quantum-based extensions to BCO, where the principles of quantum computing are used to add quantum computing elements to bacterial search dynamics.

Here, the individual bacteria are no longer described by deterministic coordinates, but by probabilistic qubits (which encode possible locations). The algorithm makes use of quantum rotation and collapse operators to control the movement of bacteria and allow at the same time taking into account numerous candidate states. The algorithm uses probabilistic state transitions that involve quantum tunnelling mechanisms similar to those found in physical systems to reach remote and high-potential regions of a search. It is this combination of quantum-inspired position advances with adaptive chemotactic step-size that gives BCO a dual ability to not only explore widely but also focus narrowly in the final locations of its targets, which is the balance of exploration and exploitation. As a result, the convergence rate, stability, and strength of the algorithm are also improved and can be used to easily tackle high-dimensional, multifaceted and multimodal optimisation problems. The hybrid quantum-enhanced adaptive BCO is then seen to be a flexible instrument that can be used to solve modern-day optimisation problems.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the datasets, parameter settings, evaluation criteria, and the results and discussion are provided. Experiments were all conducted on a computer with an Intel Core i5 processor and 32GB RAM, and Windows operating system version 11. All the algorithms were created using MATLAB 2022b. To prove that the proposed BCO algorithm is better in terms of its effectiveness and efficiency, it was compared with the k-means [34], GA [35], PSO[36], improved PSO (IPSO)[37], BCO [6], and IBCO.

A. *Datasets*

The research paradigm is based on three known benchmark corpora, namely Reuters-21,578, Classic4, and WebKB, which are used as the core of all further analyses. The documents are put through a tripartite preprocessing regimen before being clustered, as it is outlined in reference [38]. First, the, a, an and which are function words that are expunged. This excision is guided by a lexicon of 571 stop words posted on the Journal of Machine Learning Research website. After this lexical pruning, the Porter Stemmer II algorithm is then used to downsize remaining tokens to canonical stems. The last step in the preprocessing steps is the use of a TF-IDF weighting scheme to give a vector-space picture that can be used to be clustered. The experiments rely on six different standard datasets, that is, Reuters-21,578, WebKB, Classic4, BBC, SMS, and DMOZ [38]. Table 2 describes the datasets, and will be discussed below.

- ***Reuters- 21578:*** The documents are unevenly spread in 135 thematic groupings, which are related to the Reuters-21578 collection, 21578 news pieces published by Reuters Newswire in 1987. In the current study, the random sample was selected on a population with eight distinct categories. Such distorted character of the classes distribution is the feature to be pointed out in these texts [39].
- ***WebKB:*** The world-wide knowledge base (WebKB) project of CMU text-learning group assembled the WebKB dataset. The four university data-set home pages were accessed by obtaining web pages and manually

categorised in seven different categories, including: student, faculty, staff, department, course, project, and other [39]. The dataset has a modest skewness in its distribution.

- ***Classic4:*** In Classic 4, there are 7,095 records in four different categories namely: CACM, CRAN, CISI, and MED. In the experiment conducted in the question, 2,000 articles were selected randomly, 500 articles selected in each group [40]. The absence of skewness is observed in the dataset.
- ***BBC*** dataset: BBC3 includes 2225 textual files obtained on the BBC news site and belonging to five different domains: business, entertainment, politics, sport, and technology [41, 42].
- ***SMS:*** This is a collection of SMS messages with tags of SMS spam research. The sample consists of 5,574 English SMS messages where the messages are classified as ham or spam [39, 43].
- ***DMOZ:*** DMOZ is a list of URLs, containing more than 100,000 URL descriptions. This paper will discuss four different types of documents, and Table 1 shows the distribution of the DMOZ data within the selected categories [44].

B. Parameters analysis

The algorithm parameters of each algorithm were taken according to the optimal settings in the founding research articles. The values of these parameters have been summarised in a table, as shown in Table 3. Parameters were selected based on the objective, which is the sum of squared error (SSE).

TABLE III
PARAMETER'S OF BCO

Parameter(s)	S	N_C	N_{re}	N_s	N_{ed}	P_{ed}	C_{min}	C_{max}
Values	50	100	4	5	2	0.2	0.01	0.2

C. Performance procedures

TABLE II
Details of datasets

S. No	Datasets	Categories	#Documents	Features
A.	Reuters	Acq	1596	7118
		Crude	253	
		Earn	2840	
		Grain	41	
		interest	190	
		money-fx	206	
		ship	108	
		trade	251	
		Total	**5485**	
B.	WebKB	Project	504	7178
		Course	930	
		Faculty	1124	
		Student	1641	
		Total	**4199**	
C.	Classic4	CACM	500	8830
		CRAN	500	
		CISI	500	
		MED	500	
		Total	**2000**	

Algorithm 1: Proposed AQBCO text clustering approach

Step 1: Preprocessing
- Step 1.1: Tokenizing
- Step 1.2: Stop word removal
- Step 1.3: Stemming
- Step 1.4: Item weighting

Step 2: Document representation

Step 3: Text clustering

Begin
- Step 3.1: Set population size, maximum iterations, and mandatory parameters
- Step 3.2: For all bacterial colonies
- Step 3.3: Chemotaxis and communication with adaptive step size
  1) If the new position makes things fit better, then increase the step size a little, and if the step size is too large, then decrease the step size to make minute adjustments.
  2) Use a quantum-inspired action to update the position of the bacterium, and explore further as well as randomly.
- Step 3.4: Reproduction and elimination
- Step 3.5: Migration
- Step 3.6: End for
- 3.7. Step. If the final state is not met, repeat step 3.2 and terminate the process otherwise.
- Step 3.8: Archive the ultimate partition of the documents as the best solution.

End

The evaluation of cluster quality in the text clustering field would use accuracy, precision, recall, and F -F-measure. These canonical evaluation protocols then enable the evaluation of document clusters and the establishment of the genuine partitioning in each cluster, by playing on class labels. The effectiveness of text-clustering algorithms is evaluated with the help of performance measures, and the four measures that can be used in the current article are the following:

1) *Accuracy:* The analysis of the accuracy measure is used to determine the right documents to include in every category of the presented dataset. This measure of accuracy is operationalized in the following way:

$$AC = \frac{1}{n}\sum_{i=1}^{k} \quad n_{i,i} \tag{10}$$

Where the component $n_{i,i}$ is the number of accurate candidates of the class in the cluster. The notation n represents the overall number of documents, whereas K represents the total number of different groups.

2) *Precision:* The accuracy is calculated according to the given class label in the datasets in each group. Precision test is a ratio of documents to the overall number of documents in the groups. The accuracy is calculated as a class i and j as follows,

$$P(i,j) = \frac{n_{ij}}{n_j} \tag{11}$$

Where, n_{ij} is the representative of the number of correct of the class i in the j group. n_j is represented by the total number of objects in the j group.

3) *Recall:* According to the class labels assigned, the recall (R) measure of each cluster is calculated. This recall value is expressed by the sum of all the items in the data set that are relevant, divided by the number of key documents in each category. The value of the recall in class i and cluster j are calculated using the following formula**.**

$$R(i,j) = \frac{n_{ij}}{n_i} \tag{12}$$

Where, n_{ij}-denotes the number of right of class i in the j group. n_j-denotes the total number of objects in the j group.

4) *F-Measures:* The F-measures are supposed to measure clusters in the investigated partition based on matching them with class label partition clusters that show the most significant correspondence. Combining precision with recall, this measure is a major criterion when it comes to cluster evaluation. The F-measure is calculated by using the following equation:

$$F(j) = 2 \times \frac{P(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \tag{13}$$

Where, $P(i,j)$ - precision value. $R(i,j)$ - recall value. Therefore, the following equation is how the f-measure is considered:

$$F = \frac{\sum_{j=1}^{K} \quad F_j}{K} \tag{14}$$

D) Results and discussions

The present section is also critical in showing how effective, reliable and above all superior the proposed AQBCO framework is. In this section, a specific analysis of the model performance in different circumstances is conducted and a comparison of the model with currently used optimization-based learning algorithms like Improved BCO (IBCO), standard BCO, GQPSO, IPSO, PSO, GA and k-means. The key idea of it is to convert the quantitative results of experiments, including Accuracy, Precision, Recall, and F-Measure, into interpretive meanings that prove the advancements made by the framework in the performance of text classification and clustering. The results of the proposed AQBCO algorithm compared to the existing optimization methods (IBCO, BCO, GQPSO, IPSO, PSO, GA, and K-Means) on six common text classification datasets, namely Reuters, WebKB, Classic4, BBC, SMS, and DMOZ, are shown in Tables 4-9 and illustrated in Figures 1-6. The accuracy, precision, recall, and the F-measure metrics were used to assess the methods against Word2Vec, TF-IDF and FastText word embedding models. 1.

According to Figure 1, the AQBCO model had the best classification performance among all the techniques of embedding. With the FastText representation, AQBCO achieved an accuracy of 99.37, a precision of 99.12, a recall of 99.78 and a F-measure of 99.33 when using the Reuters Dataset. This is much better than the IBCO and the BCO models. AQBCO has an improved chemotaxis adaptation and quantum-inspired search, which allows improved exploration-exploitation balance, mitigating misclassification and improving convergence. Conventional optimizers like PSO, GA, and K-Means had a relatively poor performance because they converged prematurely and their ability to learn features was weak.

TABLE IV
COMPARATIVE RESULTS ANALYSIS FOR THE REUTERS DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word2Vec	**TF-IDF**	**Fast Text**	**Word2Vec**	**TF-IDF**	**Fast Text**	**Word2Vec**	**TF-IDF**	**Fast Text**	**Word2Vec**	**TF-IDF**	**FastText**
AQBCO	**98.42**	**98.83**	**99.37**	**98.05**	**98.61**	**99.12**	**99.01**	**99.36**	**99.78**	**98.53**	**98.79**	**99.33**
IBCO	96.89	97.27	97.95	96.44	96.93	97.48	97.31	97.60	98.41	96.87	97.26	97.94
BCO	95.63	96.08	96.82	92.75	93.02	94.11	98.83	99.04	99.57	95.39	95.78	96.61
GQPSO	94.22	94.71	95.86	91.04	91.58	92.47	97.44	97.83	98.62	94.01	94.47	95.68
IPSO	93.65	94.23	95.44	90.85	91.29	92.18	96.95	97.38	98.41	93.37	93.96	95.28
PSO	89.74	90.58	93.73	87.41	88.19	91.04	91.12	91.96	95.42	89.36	90.28	93.41
GA	87.56	88.92	91.46	85.08	86.42	89.15	89.62	90.41	93.12	87.12	88.57	91.22
K-Means	84.11	85.72	87.94	82.23	83.74	85.69	85.43	86.89	88.67	83.71	85.29	87.76

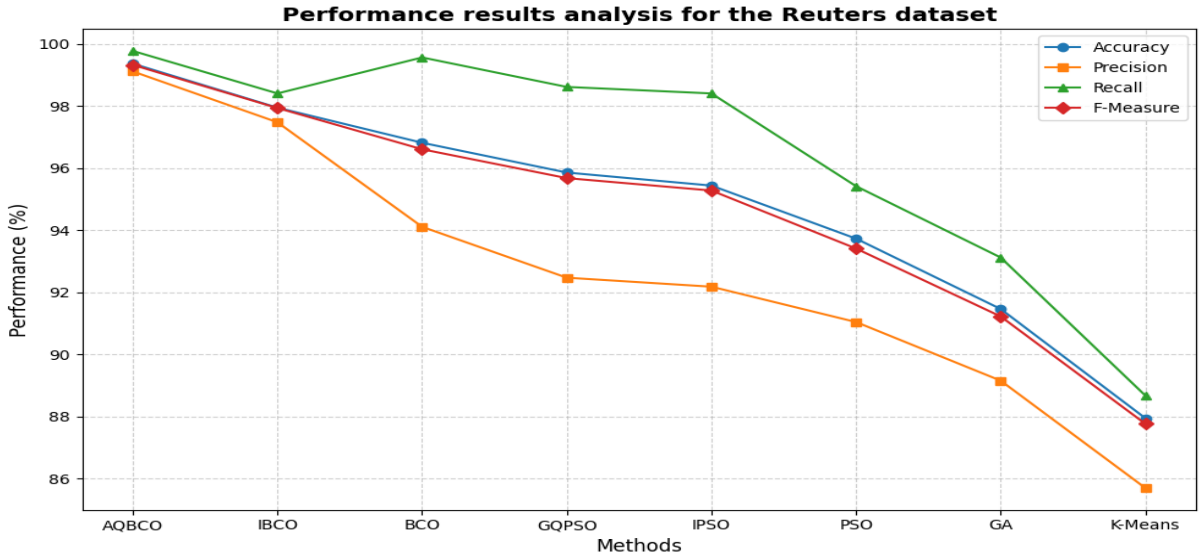

Fig. 1 : Performance results analysis for the Reuters dataset

TABLE V
COMPARATIVE RESULTS ANALYSIS FOR THE WEBKD DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word2 Vec	TF-IDF	FastText	Word2 Vec	TF-IDF	FastText	Word2 Vec	TF-IDF	FastText	Word2 Vec	TF-IDF	FastText
AQBCO	**97.85**	**98.26**	**98.92**	**97.53**	**98.07**	**98.64**	**98.61**	**98.92**	**99.46**	**97.79**	**98.18**	**98.85**
IBCO	96.64	96.96	97.71	96.28	96.93	97.43	96.97	96.99	97.98	96.62	96.96	97.71
BCO	94.43	95.11	95.60	90.23	90.94	91.28	98.51	99.21	99.92	94.19	94.89	95.40
GQPSO	91.46	92.32	94.02	87.93	88.41	89.76	94.61	95.91	98.13	91.15	92.00	93.76
IPSO	88.69	90.19	91.05	85.11	86.61	87.49	91.66	93.28	94.20	88.27	89.82	90.72
PSO	84.34	87.10	90.19	79.89	82.41	86.61	87.69	90.93	93.28	83.61	86.46	89.82
GA	81.17	83.05	85.01	77.93	79.31	80.73	83.33	85.71	88.28	80.54	82.39	84.34
K-Means	78.97	80.02	81.22	76.93	77.60	78.49	80.19	81.55	83.03	78.53	79.52	80.70

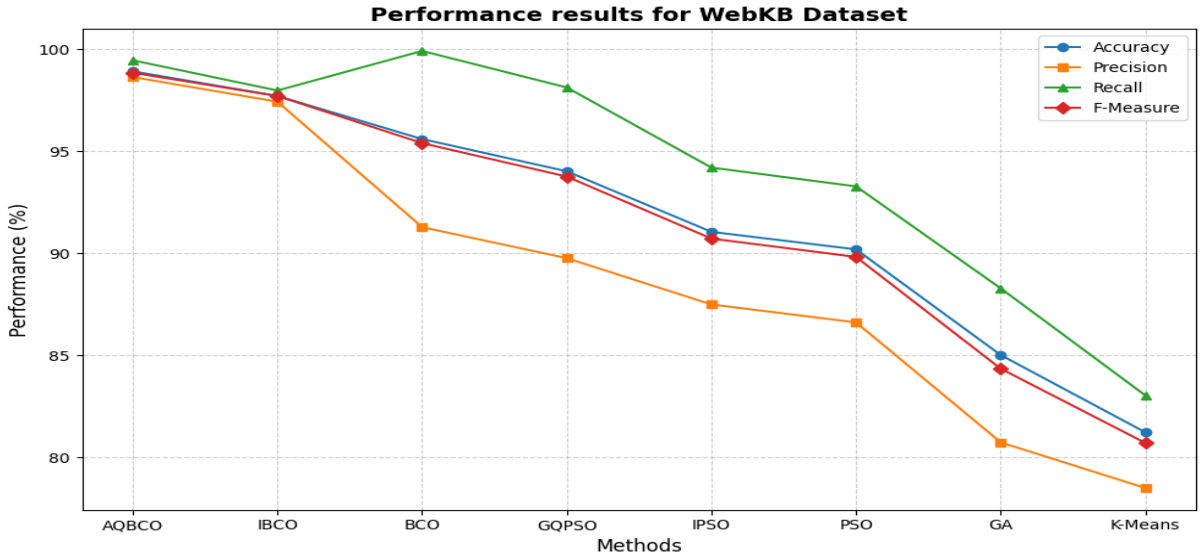

Fig. 2 : Performance results analysis for the WebKD dataset

TABLE VI
COMPARATIVE RESULTS ANALYSIS FOR THE CLASSIC4 DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word 2Vec	**TF-IDF**	**FastText**	**Word2Vec**	**TF-IDF**	**Fast Text**	**Word 2Vec**	**TF-IDF**	**Fast Text**	**Word 2Vec**	**TF-IDF**	**Fast Text**
AQBCO	**85.93**	**87.12**	**88.67**	**85.11**	**86.29**	**87.83**	**86.42**	**87.64**	**89.15**	**85.74**	**86.95**	**88.43**
IBCO	82.40	83.38	84.55	81.24	82.29	83.49	83.17	84.12	85.29	82.19	83.19	84.38
BCO	79.36	80.10	81.31	76.94	77.86	79.34	80.86	81.51	82.59	78.85	79.64	80.93
GQPSO	75.49	76.02	77.02	72.87	73.43	74.12	76.90	77.45	78.69	74.83	75.38	76.33
IPSO	73.71	74.23	76.20	71.28	71.98	73.43	74.91	75.38	77.74	73.05	73.64	75.52
PSO	70.03	72.21	74.22	69.62	70.98	72.14	70.20	72.76	75.27	69.91	71.86	73.67
GA	65.10	66.04	69.64	64.26	65.29	69.29	65.35	66.28	69.78	64.80	65.78	69.53
K-Means	57.11	59.81	61.79	54.94	57.93	59.81	57.43	60.19	62.28	56.15	59.04	61.02

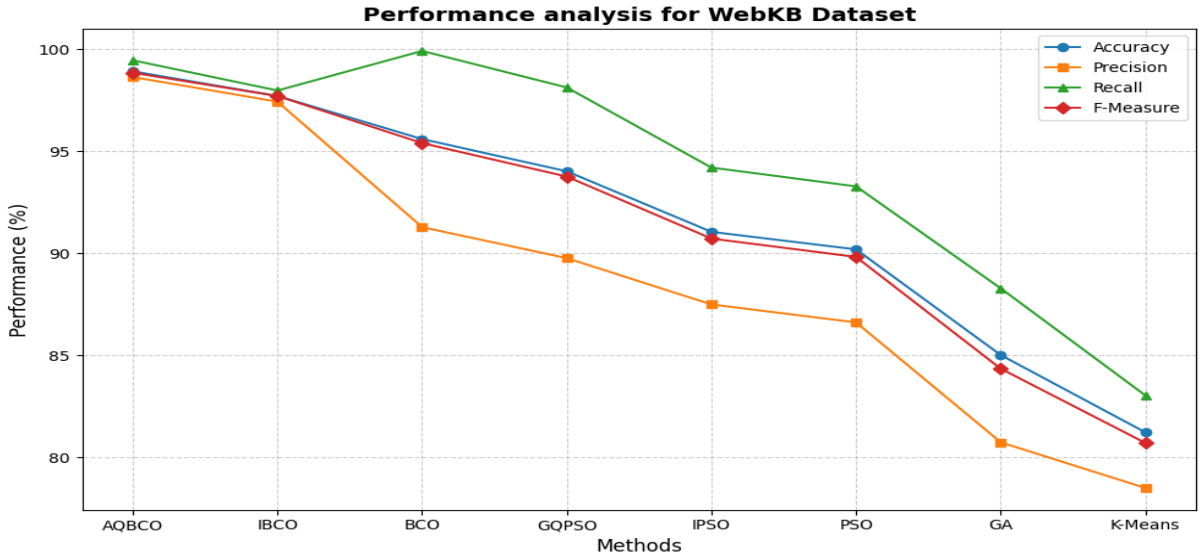

Fig. 3 : Performance results analysis for the classic4 dataset

TABLE VII
COMPARATIVE RESULTS ANALYSIS FOR THE BBC DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text
AQBCO	**95.82**	**96.57**	**97.38**	**94.71**	**95.33**	**96.28**	**97.95**	**98.43**	**99.11**	**95.60**	**96.45**	**97.29**
IBCO	94.53	95.28	96.17	91.78	92.74	93.41	97.12	97.71	98.87	94.38	95.16	96.06
BCO	92.75	93.03	94.21	90.27	90.78	91.48	94.99	95.05	96.75	92.57	92.87	94.04
GQPSO	89.18	90.38	91.09	82.12	85.78	86.24	95.62	94.47	95.50	88.36	89.91	90.63
IPSO	87.11	88.10	89.18	79.81	81.23	82.12	93.46	94.18	95.62	86.10	87.22	88.36
PSO	84.62	85.28	87.26	82.62	80.63	82.78	86.06	88.90	90.93	84.30	84.56	86.66
GA	81.70	82.85	85.11	79.46	81.24	83.43	83.18	83.94	86.33	81.28	82.57	84.85
K-Means	81.30	82.28	84.11	79.16	80.61	82.29	82.69	83.39	85.40	80.89	81.98	83

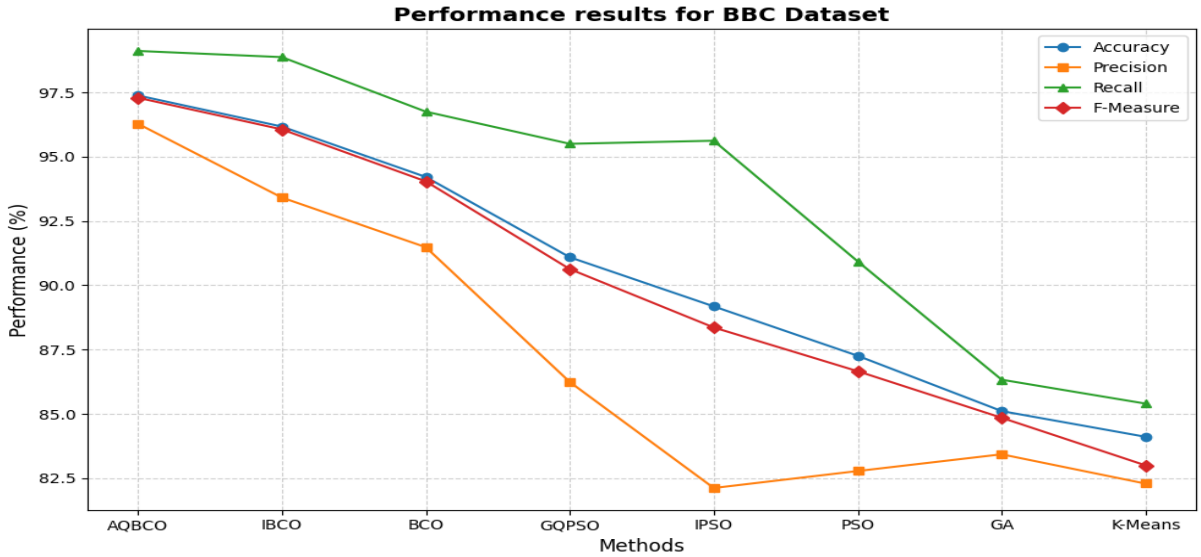

Fig. 4 : Performance results analysis for the BBC dataset

TABLE VIII
COMPARATIVE RESULTS ANALYSIS FOR THE SMS DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text
AQBCO	**94.91**	**95.83**	**96.79**	**92.67**	**93.56**	**94.32**	**97.25**	**98.10**	**98.96**	**94.74**	**95.68**	**96.61**
IBCO	93.57	94.29	95.04	90.93	91.77	92.43	96.01	96.65	97.53	93.40	94.14	94.91
BCO	91.38	92.14	93.17	89.48	89.81	90.62	93.01	94.20	95.50	91.21	91.96	92.99
GQPSO	88.78	89.25	90.10	85.62	86.49	86.94	91.40	91.54	92.81	88.41	88.94	89.77
IPSO	88.78	90.09	92.97	86.76	88.41	89.26	90.42	91.48	96.42	88.55	89.92	92.70
PSO	86.63	88.12	90.53	85.29	86.77	88.26	87.64	89.18	92.47	86.45	87.96	90.31
GA	85.59	85.99	86.87	83.27	83.52	83.94	87.32	87.86	89.16	85.24	85.64	86.47
K-Means	81.22	83.04	85.19	79.96	81.29	82.91	82.03	84.24	86.88	80.98	82.74	84.84

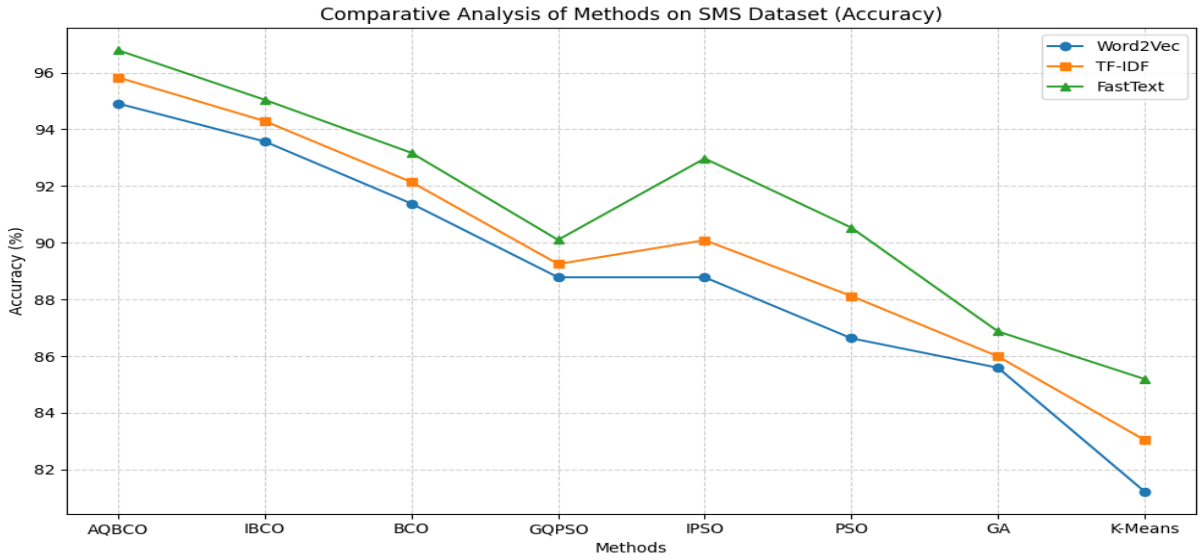

Fig. 5 : Performance results analysis for the SMS dataset

TABLE IX
COMPARATIVE RESULTS ANALYSIS FOR THE DMOZ DATASET

Methods	Accuracy			Precision			Recall			F-Measure		
	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text	Word 2Vec	TF-IDF	Fast Text
AQBCO	**91.87**	**92.75**	**93.68**	**90.92**	**91.56**	**92.48**	**93.15**	**94.05**	**95.14**	**91.73**	**92.61**	**93.55**
IBCO	90.63	91.41	92.27	89.48	89.99	90.78	91.59	92.63	93.56	90.52	91.29	92.15
BCO	87.11	88.27	89.63	85.77	86.77	87.99	88.14	89.46	90.98	86.94	88.09	89.46
GQPSO	86.60	87.68	88.11	84.93	85.44	86.77	87.87	89.45	89.17	86.37	87.40	87.95
IPSO	85.90	86.70	87.18	83.79	84.46	84.93	87.48	88.42	88.94	85.59	86.39	86.89
PSO	85.71	86.14	86.71	83.42	83.79	84.49	87.42	87.92	88.42	85.37	85.80	86.41
GA	84.64	85.22	86.46	82.11	82.94	83.92	86.49	86.91	88.40	84.24	84.87	86.10
K-Means	84.18	85.06	86.13	81.70	82.69	83.76	85.96	86.81	87.92	83.78	84.70	85.79

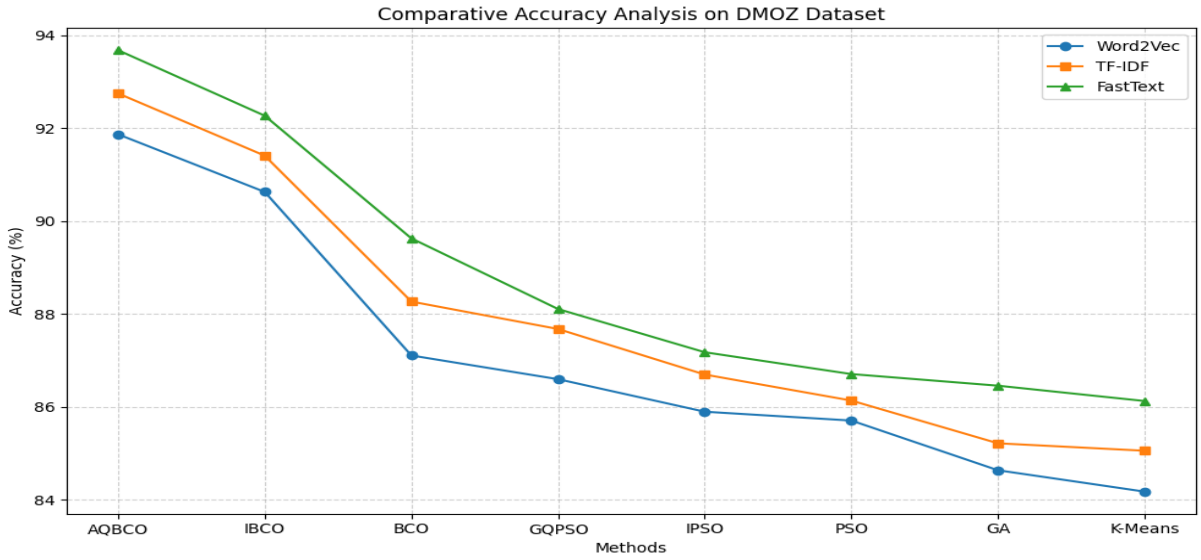

Fig. 6 : Performance results analysis for the DMOZ dataset

Figure 2 shows that the same thing goes with the WebKB dataset. AQBCO model attained an accuracy of 98.92 percent and an F-measure of 98.85 percent, which is better than IBCO (97.71) and BCO (95.40%). This data set has web page categories that overlap, and thus, efficient feature selection and optimization is essential. AQBCO is an adaptive step size and quantum mechanism, which ensures no stagnation occurs and makes the process more localized in exploitation. The GQPSO, IPSO, and PSO methods have relatively good performance, whereas GA and K-Means were at the back because of low adaptability. 3.

Figure 3 demonstrates the results at the Classic4 dataset, which is smaller and less complicated and has diverse document types. In this case, FastText gave AQBCO the highest accuracy (88.67) and F-measure (88.43). Despite having less data diversity than the Reuters and WebKB datasets, AQBCO nonetheless performed better than IBCO, BCO, and swarm-based ones. Adaptive chemotaxis mechanism enhanced parameter tuning, whereas quantum search enhanced stability when local refining. 4.

The BBC data (Figure 4) contains concise news items on various subjects. Under FastText, the AQBCO model once again performed the best with an accuracy of 97.38% and a precision of 96.28%. The recall and the F-measure were 99.11% and 97.29% respectively. This was followed by 96.17% accuracy by IBCO, 94% by BCO, and GQPSO. The sensitivity to detect the relevant news categories is high as indicated by the high value of the recall of AQBCO. AQBCO is more robust and learns more semantic relationships in the text than PSO and GA.

Figure 5, SMS Spam data, demonstrates the flexibility of AQBCO in the classification of short texts. AQBCO performed with 96.79% and 96.61% accuracy and F-measure, which is 1-3 points higher than IBCO and BCO. The dynamic quantum operator that is suggested in the proposed method boosts the diversity of movement of bacteria, eliminating stagnation and increasing accuracy even in binary classification. GQPSO and IPSO showed fair recall with no stability of precision of AQBCO. PSO, GA, and K-Means were the feeblest, meaning that they are not efficient when the data is limited to short texts. The DMOZ dataset is a large multi-classification problem in web directory. AQBCO had an accuracy of 93.68% and an F-measure of 93.55, which is better than that of IBCO (92.27%) and BCO (89.63%).

The findings indicate that AQBCO is scalable and can keep the convergence at a constant level in very large datasets. The search space is also searched efficiently by the adaptive chemotaxis behavior, and global diversity is retained by quantum movement. The disadvantageous performance of PSO, GA, and K-Means means that there was little global exploration and poor local convergence. In each of the datasets (Figures 1 to 6), AQBCO has achieved superiority over existing optimization algorithms in all the metrics of performance and embeddings. FastText has always achieved the best scores, then TF-IDF and Word2Vec. AQBCO had a better recall and F-Measure in all datasets, and this is in respect to a better accuracy and strength. AQBCO continued to have better performance with the small (SMS, Classic4) and large (Reuters, DMOZ) data sets. The potential early convergence was avoided in adaptive chemotaxis and quantum updating, leading to quicker convergence to optimal feature subsets. The tables below (Table 10) show the most successful performance of the suggested AQBCO-based text classification model on the different benchmark datasets through the application of different embedding techniques. Within the Reuters dataset, the AQBCO model with FastText embedding gave the highest accuracy of 99.37% and F-measure of 99.33, which indicates a very high level of feature optimization, and it can better adapt to a wide variety of news. The WebKB data set had a high level of accuracy and F-measure (98.92 and 98.85, respectively) that demonstrated the strength of the model when it comes to multi-

topic web pages. The Classic4 data set created 88.67% precision and 88.43% F-score, which means a great portion of accuracy and decreases the subject variety in smaller text corpora.

TABLE X
BEST PERFORMANCE RESULTS FOR DIFFERENT DATASETS

Dataset	Top Model	Embedding	Accuracy (%)	F-Measure (%)	Observation
Reuters	AQBCO	FastText	**99.37**	**99.33**	Excellent feature optimization
WebKB			**98.92**	**98.85**	Robust multi-topic classification
Classic 4			**88.67**	**88.43**	Good balance for a smaller dataset
BBC			**97.38**	**97.29**	High generalization across topics
SMS			**96.79**	**96.61**	Effective short-text classification
DMOZ			**93.68**	**93.55**	Strong performance in large-scale web data

In the case of the BBC data, the AQBCO-based classifier achieved an accuracy and F-measure of 97.38 and 97.29 percent, respectively, which proves the good generalization capacity of the model to various news categories. The SMS dataset, which consisted of short and informal messages, had 96.79% accuracy and 96.61% F-measure, which shows that the model was efficient in light-text sentiment classification. Lastly, the DMOZ data provided an accuracy of 93.68 and a F-measure of 93.55, which indicates that the AQBCO model has good scalability and predictability when used in large-scale data in the form of web directory data. Altogether, these findings prove the suggested AQBCO framework to be effective in improving the feature selection and optimization and attains high accuracy and F-measure with consistency regardless of the size, complexity, and structure of the used datasets.

V. CONCLUSIONS

In this study, an AQBCO algorithm was suggested and implemented on text classification problems with various word embedding features: Word2Vec, TF-IDF, and FastText. The AQBCO approach also used adaptive chemotaxis to adjust search dynamically and quantum-based position update in order to improve global search and eliminate local stagnation. The experimental results showed that AQBCO was always better than other optimization techniques, such as IBCO, BCO, GQPSO, IPSO, PSO, GA, and K-Means in accuracy, precision, recall, and F-measure. These findings indicate that AQBCO offers an effective trade-off between exploration and exploitation, which results in accelerated convergence and better performance of classification. Nevertheless, the offered strategy has its limitations, even though it has its benefits. One, the AQBCO algorithm entails the use of several control parameters that must be tuned with care to various datasets, and this may be computationally costly. Second, the deep word embedding models make the models more expensive in terms of training time and memory consumption, especially when training on large datasets is needed. Lastly, the approach has been mostly applied to static data sets; its ability to support streaming or real-time streams has not yet been investigated. To be improved in the future, the AQBCO framework can be expanded with auto-tuning or meta-learning frameworks to allow the auto-tuning of its parameters during training.

The study of distributed or parallel versions of AQBCO can also minimize the computational expense and enhance scalability. In addition, further research may be undertaken in the application of the AQBCO model to multilingual domain-adaptative and real-time text analysis tasks to ensure further practical application of this model in other natural language processing tasks.

REFERENCES

[1]. J. M. Sanchez-Gomez, S. Santander-Jiménez, and M. A. Vega-Rodríguez, "Multi-objective swarm-intelligence algorithm for document clustering," *Expert Systems with Applications*, vol. 289, p. 128348, 2025.

[2]. B. Zeng, X. Shang, R. Lu, and Y. Zhang, "Particle swarm optimization-based NLP methods for optimizing automatic document classification and retrieval," *PloS one*, vol. 20, no. 7, p. e0325851, 2025.

[3]. J. Rautaray, S. Panigrahi, A. K. Nayak, P. Sahu, and K. Mishra, "Deep learning based feature extraction technique for single document summarization using hybrid optimization technique," *IEEE Access*, 2025.

[4]. O. M. Alyasiri, Y.-N. Cheah, H. Zhang, O. M. Al-Janabi, and A. K. Abasi, "Text classification based on optimization feature selection methods: a review and future directions," *Multimedia Tools and Applications*, vol. 84, no. 15, pp. 14187-14233, 2025.

[5]. M. Vinitha and S. Vasundra, "An Efficient Ant Colony Optimization Optimized Deep Belief Network Based Text Summarization Using Diverse Beam Search Computation for Social Media Content Extraction," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 6, 2024.

[6]. S. Dhanalakshmi, S. Sathiyabama, and D. Ayyamuthukumar, "A novel text clustering approach based on bacterial colony optimization," in *2023 7th international conference on electronics, communication and aerospace technology (ICECA)*, 2023: IEEE, pp. 1847-1853.

[7]. H. Dabjan and M.-B. Kurdy, "Efficient Streamlined Online Arabic Web Page Classification Using Artificial Bee Colony Optimization," *Iraqi Journal of Science*, pp. 7220-7236, 2024.

[8]. S. Periyasamy and R. Kaniezhil, "Original Research Article Enhanced feature selection with bacterial foraging and rough set analysis for document clustering," *Journal of Autonomous Intelligence*, vol. 7, no. 5, 2024.

[9]. I. M. Widiartha, R. S. Hartati, D. M. Wiharta, N. P. Sastra, and L. G. Astuti, "Multi-Document Summarization Using Tuna Swarm Optimization and Markov Clustering," *JOIV: International Journal on Informatics Visualization*, vol. 9, no. 4, pp. 1563-1570, 2025.

[10]. M. A. Al-Betar, A. K. Abasi, G. Al-Naymat, K. Arshad, and S. N. Makhadmeh, "Bare-bones based salp swarm algorithm for text document clustering," *IEEE Access*, vol. 11, pp. 100010-100028, 2023.

[11]. B. M. Gurusamy, P. K. Rangarajan, and A. Altalbe, "Whale-optimized LSTM networks for enhanced automatic text summarization," *Frontiers in artificial intelligence*, vol. 7, p. 1399168, 2024.

[12]. M. Azarshab, M. Fathian, and B. Amiri, "An enhanced Teaching-Learning-Based Optimization (TLBO) with Grey Wolf Optimizer (GWO) for text feature selection and clustering," *arXiv preprint arXiv:2402.11839*, 2024.

[13]. W. A. ABUAIN, "IMPROVED SALP SWARM ALGORITHM FOR TEXT DOCUMENT CLUSTERING," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 14, 2024.

[14]. R. Dodda and A. S. Babu, "Text document clustering using chaotic northern goshawk optimization with K-means algorithm," *Int. J. Intell. Eng. Syst*, vol. 17, no. 3, pp. 630-642, 2024.

[15]. B. Radomirović *et al.*, "Text document clustering approach by improved sine cosine algorithm," *Information Technology and Control*, vol. 52, no. 2, pp. 541-561, 2023.

[16]. M. K. Muddada, J. Vankara, S. S. Nandini, G. R. Karetla, and K. S. Naidu, "Multi-objective ant colony optimization (moaco) approach for multi-document text summarization," *Engineering Proceedings*, vol. 59, no. 1, p. 218, 2024.

[17]. J. Rautaray, S. Panigrahi, and A. K. Nayak, "Integrating particle swarm optimization with Backtracking search optimization feature extraction with two-dimensional convolutional neural network and attention-based stacked bidirectional long short-term memory classifier for effective single and multi-document summarization," *PeerJ Computer Science*, vol. 10, p. e2435, 2024.

[18]. B. A. S. Emambocus *et al.*, "A clustering algorithm employing salp swarm algorithm and K-means," in *2024 20th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 2024: IEEE, pp. 108-113.

[19]. X. Sun, "Intelligent Text Clustering Analysis of Novels Based on Digital Semantic Similarity Calculation," 2024.

[20]. E. Hassan *et al.*, "A Hybrid K-Means++ and Particle Swarm Optimization Approach for Enhanced Document Clustering," *IEEE Access*, 2025.

[21]. M. Naderi and M. Amiri, "A Novel Hybrid Method for Clustering Text Documents using Evolutionary Optimization," in *2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2023: IEEE, pp. 369-374.

[22]. M. H. Mousa, A. E. Khedr, and A. M. Idrees, "Hierarchical Method for Automated Text Documents Classification," *International Arab Journal of Information Technology (IAJIT)*, vol. 22, no. 1, 2025.

[23]. M. A. Mosa, "Optimizing text classification accuracy: a hybrid strategy incorporating enhanced NSGA-II and XGBoost techniques for feature selection," *Progress in Artificial Intelligence*, pp. 1-25, 2025.

[24]. B. Niu and H. Wang, "Bacterial colony optimization," *Discrete Dynamics in Nature and Society*, vol. 2012, 2012.

[25]. K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems magazine*, vol. 22, no. 3, pp. 52-67, 2002.

[26]. S. D. Muller, J. Marchetto, S. Airaghi, and P. Kournoutsakos, "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 16-29, 2002, doi: 10.1109/4235.985689.

[27]. M. Bey, P. Kuila, B. B. Naik, and S. Ghosh, "Quantum-inspired particle swarm optimization for efficient IoT service placement in edge computing systems," *Expert Systems with Applications*, vol. 236, p. 121270, 2024.

[28]. R. S. Parpinelli, G. F. Plichoski, R. S. D. Silva, and P. H. Narloch, "A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms," *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 1-20, 2019.

[29]. R. Li and Z.-J. Hu, "A self-adaptive particle swarm optimisation and bacterial foraging hybrid algorithm," *International Journal of Wireless and Mobile Computing*, vol. 11, no. 3, pp. 258-265, 2016.

[30]. S. S. Babu and K. Jayasudha, "A simplex method-based bacterial colony optimization algorithm for data clustering analysis," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 12, p. 2259027, 2022.

[31]. V. Prakash, V. Vinothina, K. Kalaiselvi, and K. Velusamy, "An improved bacterial colony optimization using opposition-based learning for data clustering," *Cluster Computing*, vol. 25, no. 6, pp. 4009-4025, 2022.

[32]. K. Tamilarisi, M. Gogulkumar, and K. Velusamy, "Data clustering using bacterial colony optimization with particle swarm optimization," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021: IEEE, pp. 1-5.

[33]. P. Sathya and R. Kayalvizhi, "Optimal segmentation of brain MRI based on adaptive bacterial foraging algorithm," *Neurocomputing*, vol. 74, no. 14-15, pp. 2299-2313, 2011.

[34]. V. K. Singh, N. Tiwari, and S. Garg, "Document clustering using k-means, heuristic k-means and fuzzy c-means," in *2011 International conference on computational intelligence and communication networks*, 2011: IEEE, pp. 297-301.

[35]. W. Song and S. C. Park, "Genetic algorithm for text clustering based on latent semantic indexing," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1901-1907, 2009.

[36]. S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Open Computer Science*, vol. 3, no. 2, pp. 69-90, 2013.

[37]. Y. A. Alhaj *et al.*, "A novel text classification technique using improved particle swarm optimization: A case study of Arabic language," *Future Internet*, vol. 14, no. 7, p. 194, 2022.
[38]. K. K. Bharti and P. K. Singh, "Chaotic gradient artificial bee colony for text clustering," *Soft Computing*, vol. 20, pp. 1113-1126, 2016.
[39]. A. C. Cachopo. "Datasets for single-label text categorization." https://ana.cachopo.org/datasets-for-single-label-text-categorization (accessed.
[40]. "Classic3 and Classic4 Datasets." ftp://ftp.cs.cornell.edu/pub/smart/ (accessed.
[41]. B. Bose. "BBC News Classification." https://storage.googleapis.com/dataset-uploader/bbc/bbc-text.csv. (accessed.
[42]. D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 377-384.
[43]. J. H. Tiago Almeida. "SMS Spam Collection." https://archive.ics.uci.edu/dataset/228/sms+spam+collection (accessed.
[44]. A. Shawon. "URL Classification Dataset [DMOZ]." https://www.kaggle.com/datasets/shawon10/url-classification-dataset-dmoz (accessed.)

