

Artificial Bee Colony Algorithm with Cosine Similarity on Raspberry Pi Cluster for Enhancing Medical Data Classification

Anan Banharnsakun*

Computational Intelligence Research Laboratory, Computer Engineering Department,
Faculty of Engineering at Sriracha, Kasetsart University, Chonburi, Thailand
Email: anan@eng.src.ku.ac.th*

Received: June 9, 2025 / Revised: November 11, 2025 / Accepted: November 19, 2025

Abstract—Machine learning plays a very important role in our daily lives. Machine learning is used to solve difficult problems encountered in many fields of study. Especially in medical diagnosis, machine learning is used to support patient diagnosis. In this research, the artificial bee colony approach, a well-known bio-inspired algorithm in the field of machine learning, is proposed to enhance the classification of cancer diagnosis data by combining it with cosine similarity measurements and parallel computing on Raspberry Pi clusters. The performance of the proposed method is validated and compared with other well-known algorithms against the Breast Cancer Wisconsin dataset taken from the UCI machine learning repository. Experimental results indicate that the proposed method achieves superior classification accuracy compared to existing approaches, yielding improvements of up to 6.45%. Furthermore, it can scale effectively to support large-scale medical data for disease diagnosis.

IndexTerms—Artificial Bee Colony (ABC), Cosine Similarity, Classification, Parallel Computing, Raspberry Pi

I. INTRODUCTION

In the field of medical care, a large volume of data has been generated and collected, including patient information, symptoms, and treatment methods. These data need to be analyzed and managed effectively to support an accurate diagnosis [1]. However, data analysis is a complex and challenging task. Therefore, the classification of medical data, which is one of the important data analysis processes in healthcare, has attracted the attention of many researchers to present methods to help categorize data more efficiently [2].

Over the past decades, an increasing number of methods have been introduced to classify medical data. This has been done to support clinicians in

personalizing medical treatments and engaging in shared decision-making. Fuente-Tomas et al. [3] proposed using the *k*-means method for cluster-based classification to account for the varying degrees of severity of bipolar disorder. On the other hand, Nanglia et al. [4] presented a hybrid algorithm that combines Support Vector Machine (SVM) and neural networks to improve accuracy and reduce computational complexity in lung cancer classification. Hamed et al. [5] introduced a variant of the *k*-Nearest Neighbors (*k*-NN) algorithm for classifying COVID-19 data, aiming to improve patient survival rates through early detection. Alam et al. [6] proposed a feature-ranking-based methodology that leverages feature selection with the random forest algorithm to train and construct models for medical data classification.

While current algorithms for data classification are effective in improving accuracy and producing favorable outcomes, they are not without limitations [7]-[9]. These algorithms have various drawbacks, including sensitivity to initial values, incompatibility with high-dimensional datasets, susceptibility to overfitting, and extensive computational time. Despite these limitations, there remains a need for further advancements in developing an efficient and effective data classification method that can overcome these challenges.

Over the past two decades, the use of algorithms inspired by biology has demonstrated its effectiveness in addressing numerous challenges in science and engineering [10]-[13]. Various techniques utilize these biologically inspired algorithms to tackle problems in data classification [14]-[16]. However, as no single approach is sufficient to meet all the requirements of data classification, researchers are continually motivated to develop new algorithms that offer more efficient solutions and enhance classification precision.

The ABC algorithm, developed by Karaboga [17], is a widely used approach for finding optimal solutions to optimization problems [18]-[20]. This technique is based on the natural process of honeybees searching

for high-quality food. Prior research [21]-[23] has demonstrated the ABC method's efficacy and efficiency in resolving a diverse array of optimization problems, surpassing other methods. In this study, we employ the ABC approach incorporating cosine similarity (ABC-CS) to classify medical data. This study aims to demonstrate the utility and effectiveness of the ABC-CS method, a straightforward and efficient approach inspired by bee foraging, for classifying medical data. The significance of this work lies in its demonstration of the ABC-CS method's applicability to the medical domain.

However, the large volume of medical data necessitates parallel computing to efficiently process and analyze this wealth of information, enabling quicker and more accurate clinical decision-making and medical research [24]-[26]. The Raspberry Pi, originally designed as an educational tool, has evolved into a versatile platform for IoT applications due to its low cost, compact form factor, and energy efficiency [27]-[29]. In this work, we also present the Raspberry Pi combined with parallel computing techniques, specifically our proposed ABC-CS method, which enables processing and analysis of large medical datasets.

The paper is structured as follows: Section 2 provides an overview of the background and knowledge, including data classification with cosine similarity and the artificial bee colony algorithm. Section 3 proposes enhancing data classification using the ABC-CS method. Section 4 introduces the ABC-CS with parallel computing on Raspberry Pi clusters. Section 5 presents the experimental settings and results. Finally, Section 6 concludes the paper.

II. BACKGROUND AND KNOWLEDGE

A. Classification with Cosine Similarity

Data classification refers to the systematic arrangement of data into meaningful categories based on their characteristics. In this specific context, centroid-based data classification is used to categorize the data. This method involves calculating a centroid vector that represents the data of each class. To determine the class of new data, the cosine similarity function is used to compare it to each centroid vector. The new data is then assigned to the class with the most similar centroid vector, as measured by cosine similarity [30]. This approach is useful in identifying patterns and relationships within data [31], [32], making it easier to analyze and make informed decisions based on the data.

Let $x = (x_1, x_2, x_3, \dots, x_n)$ and $y = (y_1, y_2, y_3, \dots, y_n)$ be two vectors for comparison. Using the cosine measure as a similarity function, it can be defined as in (1):

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (1)$$

where $\|x\|$ and $\|y\|$ are the Euclidean norm of vectors, defined as $\sqrt{x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2}$ and $\sqrt{y_1^2 + y_2^2 + y_3^2 + \dots + y_n^2}$, respectively

The calculation of cosine similarity between two vectors, x and y , involves determining the cosine of the angle between them. A cosine similarity score of 0 indicates that the vectors are completely orthogonal, meaning they have no overlap or correlation. In contrast, a cosine similarity score of 1 means that the vectors are perfectly aligned, with no angle between them. The closer the cosine similarity score is to 1, the stronger the correlation and similarity between the vectors, as the angle between them is smaller.

In terms of using cosine similarity to classify the data, if x is the centroid vector of a class and y is a data vector, $\text{sim}(x, y)$ is used to determine whether the data vector y should belong in that class. When the data vector y is compared to the centroid vector of all classes, the data vector y will be assigned to the class that gives the most $\text{sim}(x, y)$ value.

B. ABC Method

In the last decade, the ABC approach has been the subject of a wide range of research and effective applications in the optimization domain. It is an optimization algorithm inspired by the intelligent foraging behavior of honey bee swarms. The bee agents in the ABC approach are categorized into three types. The first type is called employed bees. The next type is the onlooker bee, and the last type is the scout bee. The food source position represents a feasible solution in the optimization problem, and the food source quality measures solution quality, referred to as 'fitness'. The process of bees searching for a desirable food source through foraging is similar to identifying the optimal solution.

A randomly distributed initial food source position (solution) is generated in the first step of the ABC approach and assigned to each employed bee. Next, the employed bees search for new candidate food source positions around the old ones in their memory. When an employed bee discovers a new food source, it assesses the fitness of both the new and the previous food source positions. If the fitness of the new food source is greater than that of the old one, the bee will memorize the position of the new food source and forget the position of the old one. However, if the fitness of the old food source is still higher, the bee will retain the position of the old food source in its memory. The employed bees then share the information regarding their food sources with the onlooker bees. Each onlooker evaluates the fitness of the food source positions taken by all employed bees and then chooses a food source based on the fitness of each position. Similar to the employed bees, the onlooker bees then search for new candidate food

source positions around the old ones in their memory. An onlooker bee will memorize the position of a new food source if its fitness is higher than that of the previous one, and forget the old one. Any food source position discovered by employed or onlooker bees that does not show quality improvement will be discarded and replaced with a new position determined by a scout bee.

III. ENHANCEMENT OF DATA CLASSIFICATION BY USING ABC-CS

The proposed ABC-CS method aims to find optimal centroid values for data classification with improved efficiency. The centroid values, which serve as parameters for the data classes, are a crucial aspect of the classification process and need to be optimized. This is achieved using the ABC-CS method, which optimizes these values and improves the overall effectiveness of the data classification process.

The classification problem involves assigning N objects, represented as $O = \{o_1, o_2, o_3, \dots, o_N\}$, to K classes, represented as $C = \{c_1, c_2, c_3, \dots, c_K\}$. Objects and classes are characterized by D values that specify their positions within a multidimensional space. Each object, denoted as o_i , and each class centroid, denoted as c_j , is formally represented as a D -dimensional vector, where each component corresponds to a distinct dimension. The purpose of the classification process is to optimize the total

cosine similarity between each data point, also referred to as an “object,” and the central point, known as the centroid, of the class it is classified into. The aim is to maximize the sum of these cosine similarity scores to improve classification accuracy. This is achieved by adjusting the centroids’ positions to align as closely as possible with the objects they represent, thereby increasing the similarity between each object and its corresponding centroid. In this way, the classification process optimizes cosine similarity between objects and centroids, yielding more accurate and meaningful classifications. This is expressed as the objective function, which serves as the fitness function of ABC-CS and can be represented by (2). The objective is to allocate each object to one of the K classes in a way that maximizes the cosine similarity between the object and the centroid of its assigned class.

$$\text{Fitness} = \sum_i^N \sum_j^K w_{ij} \cdot \text{sim}(o_i, c_j) \quad (2)$$

where w_{ij} is the association weight between object o_i and class j . It is a value that indicates the degree of association between the object and the class. If i is allocated to class j , then w_{ij} equals 1. On the other hand, if o_i is not allocated to class j , then w_{ij} equals 0.

The ABC-CS algorithm for finding the optimal centroid values in the data classification process is described and shown in Fig. 1.

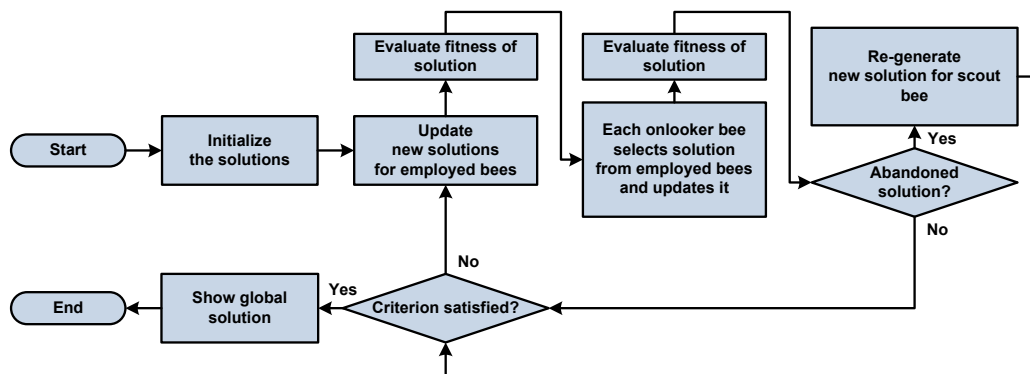


Fig. 1. Flowchart for finding optimal centroid values using the ABC-CS algorithm

As shown in Fig. 1, the artificial bee colony algorithm begins by randomly generating initial solutions (centroid values) that serve as the food source positions for the bee agents. These bee agents will then carry out three crucial functions. Firstly, the employed bees update the feasible food source positions by searching for new ones using (3) and comparing their own food source position with their neighbors’ previously selected food source positions. Secondly, the onlooker bees choose the feasible food source positions. Finally, the scout bees prevent further deterioration of the quality of the food sources.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (3)$$

where the solution v_{ij} is obtained by modifying its original value x_{ij} through a comparison with a randomly selected solution x_{kj} from its neighboring solutions. The alteration is performed using a random number ϕ_{ij} in the range $(-1, 1)$. This number is used to make a random adjustment to the previous solution, resulting in a new solution in the next iteration. The indices k and j are chosen randomly, with k , being any number from 1 to the number of food sources (SN), excluding i , and j being any number from 1 to D . The

difference between x_{ij} and x_{kj} represents a variation in the position of the solution in a specific dimension.

The employed bees in a hive are responsible for finding food sources and updating their memories with the positions of new candidate food sources. If the new food source has a better fitness value than the previous one, the memory of the employed bee will be updated accordingly. Once the employed bee has discovered a new food source, it returns to the hive and communicates its fitness value to the onlooker bees. In the second task, the onlooker bees use the information provided by the employed bees to select one of the food sources based on the fitness values. The selection process is determined by (4), which calculates the probability of a food source being chosen.

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (4)$$

where fit_i is the fitness value of the food source i , which is calculated by using (2).

The likelihood that onlooker bees select a proposed food source is proportional to its fitness value, with higher fitness values leading to a higher probability of selection. Once the onlooker bees have selected a food source, they visit that location and determine whether a nearby site is a potential new food source. The calculation for this new candidate food source site can be done using (3).

The third task involves replacing any food source position that has not shown an improvement in its fitness value. This is done to prevent suboptimal solutions. The new position that replaces the abandoned one will be selected randomly by a scout bee, using (5). This random selection helps to ensure that the solution is not stuck at a suboptimal point and allows the algorithm to explore new potential solutions.

$$x_{ij} = x_j^{\min} + rand[0,1](x_j^{\max} - x_j^{\min}), \quad (5)$$

where $x_j^{\min}$ and $x_j^{\max}$ are the lower bound and the upper bound of the food source position in dimension j , respectively.

The algorithm will repeat the three major tasks described earlier until the number of iterations reaches a pre-determined value. The number of iterations is used as a termination criterion, meaning it determines when the algorithm will stop running. Once the number of iterations reaches the specified value, the algorithm will end, and the final solution will be presented.

IV. ABC-CS WITH PARALLEL COMPUTING ON RASPBERRY PI CLUSTERS

From the past to the present, Raspberry Pi has undergone continuous development, improving its performance and enabling it to handle a variety of processing tasks, much like a single computer. The advantage of Raspberry Pi is its low cost and low energy consumption. For this reason, using a Raspberry Pi to create clusters for parallel processing of large datasets is both interesting and challenging. In our research, we have implemented parallel processing on a cluster system built from Raspberry Pi, as shown in Fig. 2, and applied the ABC-CS algorithm we proposed to this cluster. This is to evaluate the proposed algorithm's performance in terms of parallel processing efficiency.

In our parallel approach, referred to as parallel ABC-CS, we employ the Manager-worker model [33] for overseeing process management. This model comprises a singular manager and multiple workers, each with distinct responsibilities. The manager primarily handles tasks such as initialization, load balancing, partitioning the bee colony, distributing bee subgroups to workers, and collecting and presenting global results.

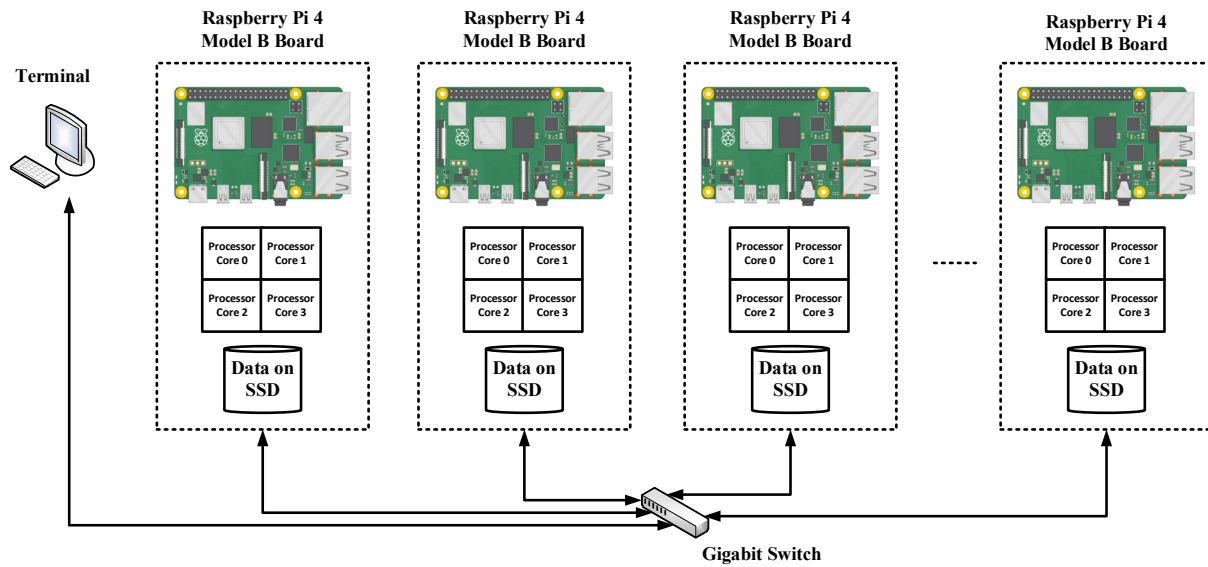


Fig. 2. Raspberry Pi 4 Cluster Topology in the proposed work

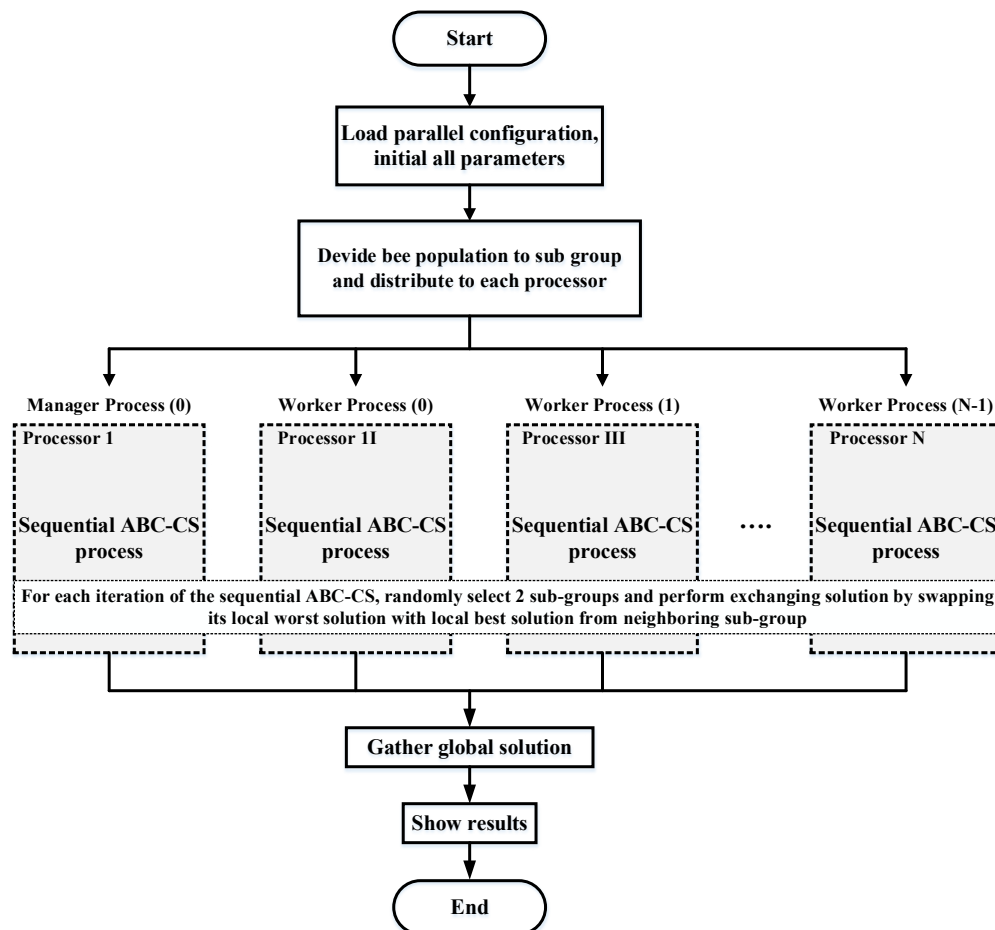


Fig. 3. The parallel ABC-CS flowchart

Meanwhile, the workers are responsible for executing the primary computations, including receiving bee subgroups from the manager, processing them, and returning the local best solution results at the final iteration.

The communication within our presented model occurs between the Manager and the Worker, as well as among the Workers themselves. This communication uses the Message-Passing Interface (MPI), a widely accepted standard communication framework. MPI provides a set of routines for message-passing operations, such as sending and receiving messages. The Manager and Workers employ these operations to exchange information and collaborate on their computations. Communicators, or process groups in MPI, define sets of processes that can communicate with one another. In the Worker-Manager Model, communicators help specify the group of processes involved in the computation, including both the Manager and Workers. Each process within the communicator is uniquely identified by an integer rank, ranging from 0 to the total number of processes minus 1. In the Worker-Manager Model, rank 0 is typically assigned to the Manager, and each Worker has a unique rank. MPI routines are integrated into the application code, facilitating seamless communication within the distributed system, including sending and receiving messages and other necessary message-passing operations.

In the presented work, each process loads a complete parallel ABC-CS code, as depicted in Fig. 3. The Manager's role involves dividing the entire bee swarm into subgroups and allocating these subgroups to individual Worker processes. Although coordinating computations is not the Manager's primary responsibility, the Manager actively participates in processing, similar to the Workers, in each round. Bees within each subgroup collaborate to improve their solutions. In the final step of each iteration, the Manager selects two random subgroups from the overall population. These two subgroups exchange their best solutions, replacing the worst solution in one group with the best solution from the other. This process emphasizes that data exchange occurs only at these synchronization points, thereby reducing communication time.

In the last round, the Manager consolidates the best solutions from all Worker processes at the local level and computes the globally optimal solution.

V. EXPERIMENTAL SETTINGS AND RESULTS

In this section, a comprehensive evaluation of the proposed method for improving the classification of medical data is presented. To assess the efficacy of the proposed method, several other widely-used approaches, such as the K-Nearest Neighbors (k-NN) [34], Support Vector Machine (SVM) [35], and the Artificial Bee Colony with K-Means (ABC-KMeans) [36], were utilized for comparison with the classification outcomes of the medical data. The experiment was performed using the Breast Cancer Wisconsin dataset obtained from the UCI Machine Learning Repository [37]. This dataset was selected as it provides a standardized benchmark for evaluating the performance of various classification algorithms. The results of the experiment aim to demonstrate the superiority of the proposed method in accuracy and efficiency compared with other well-known approaches. This dataset contains 699 instances, each with nine input attributes used to classify into two classes: Malignant or benign, as shown in Table I. The training set is randomly assigned from 75% of the data instances, and the testing set is the remaining 25%. Accuracy is used as the primary evaluation metric to compare the effectiveness of the proposed method with the previously mentioned methods.

All methods were implemented in C++ and tested on a Raspberry Pi 4 Model B with a Cortex-A72 (ARMv8) 64-bit core running at 1.8 GHz and 8 GB of memory. The number of iterations for each method was fixed at 200, and for the ABC methods, 20 bees were used as employed bees and onlooker bees. The radial basis function was used as the kernel in the SVM, with C and γ set to 1 and 0.1, respectively. In the k-NN method, K was set to 6. These parameter settings were determined to be optimal after a preliminary study was conducted as part of this work.

The Confusion matrix results for the proposed method, k-NN, SVM, and ABC-KMeans are presented in Table II, and clear visual representations of the comparison of classification accuracy, precision, recall, and F1 score for each class are provided in Fig. 4 through Fig. 7.

TABLE I
INFORMATION ON THE ATTRIBUTES OF THE BREAST CANCER WISCONSIN DATASET

No.	Attribute Description	Domain
1	Clump Thickness (CT): The clump thickness refers to the concentration or grouping of cancer cells that are stacked on top of each other. This measurement provides insight into tumor density and how cancer cells are growing and spreading.	1-10
2	Uniformity of Cell Size (UCS): The uniformity of cell size is used to assess the extent of metastasis, or the spread of cancer, to the lymph nodes. This metric can help determine whether cancer cells have spread to other parts of the body and are growing there.	1-10
3	Uniformity of Cell Shapes (UCSh): The uniformity of cell shapes is used to identify cancerous cells that vary in size. This metric helps to distinguish between normal cells and cancerous cells, which can have irregular shapes and sizes.	1-10
4	Marginal Adhesion (MA): Marginal adhesion refers to a loss of the ability of cancer cells to adhere to one another. The loss of this property is a hallmark of malignancy, as the cancer cells can become more invasive and spread to other parts of the body. Conversely, the retention of adhesion can indicate the presence of malignant cells.	1-10
5	Single Epithelial Cell Size (SECS): The size of single epithelial cells is another factor used to assess the presence of cancer. If the SECS become larger, it may indicate malignancy, as cancer cells can grow and divide faster than normal cells.	1-10
6	Bare Nuclei (BN): Bare nuclei refer to cells that lack a cytoplasmic coating. This is a common feature of benign tumors and helps to distinguish them from malignant tumors.	1-10
7	Bland Chromatin (BC): Bland chromatin is typically found in benign cells and is characterized by a uniform, light appearance. This contrasts with the denser, darker appearance of malignant cells.	1-10
8	Normal Nucleoli (NN): Normal nucleoli are usually very small in benign cells, whereas they can be larger and more prominent in cancer cells.	1-10
9	Mitoses: Mitoses refer to the process of cell division by which the nucleus of a cell splits into two new nuclei. This process is an important part of cell growth and development, and an increase in the number of mitoses can indicate a rapidly growing cancer.	1-10

The results in Table II indicate that the proposed method outperforms the other methods in terms of True Positive and True Negative values, and also has lower False Positive and False Negative values. Furthermore, Figs. 4 through 7 also suggest that the proposed method provides the best quantitative evaluation results. Fig. 8 provides a clear visual comparison of the classification accuracy results obtained from the various algorithms evaluated, including k-NN, SVM, and ABC-KMeans. The proposed method showed

an improvement in accuracy of 6.45%, 3.12%, and 4.43%, respectively. It is evident that the proposed method outperforms the other methods and provides the best results. Note that the distinction between ABC-CS and ABC-KMeans is that the fitness or similarity between data points in ABC-CS is measured using cosine similarity. In contrast, the fitness in ABC-KMeans is calculated by minimizing intra-cluster distances.

TABLE II
CONFUSION MATRIX FOR K-NN, SVM, ABC-KMEANS, AND ABC-CS CLASSIFIERS

Predicted/Actual	k-NN		SVM		ABC-KMeans		ABC-CS	
	Benign	Malignant	Benign	Malignant	Benign	Malignant	Benign	Malignant
Benign	102	5	102	6	104	6	105	4
Malignant	15	53	9	58	11	54	6	60

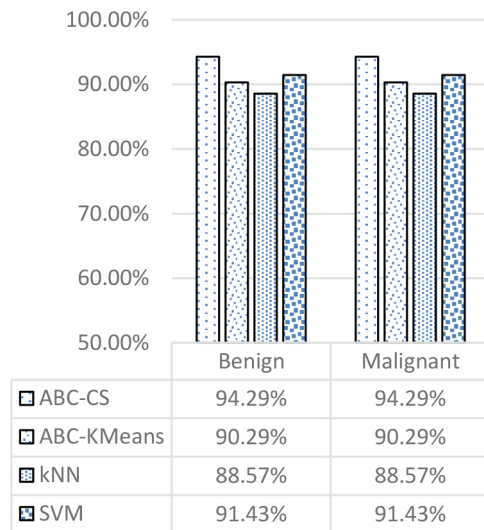


Fig. 4. The comparison of accuracy for each class

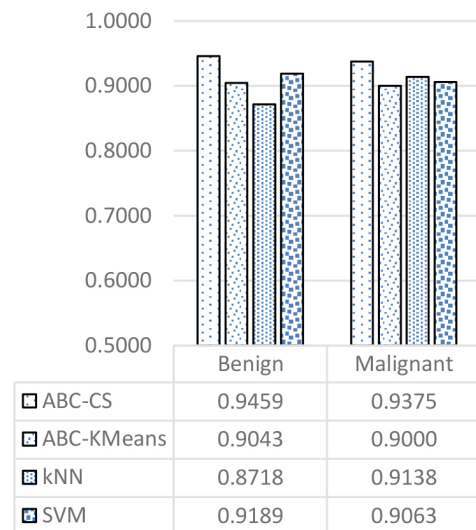


Fig. 6. The comparison of recall for each class

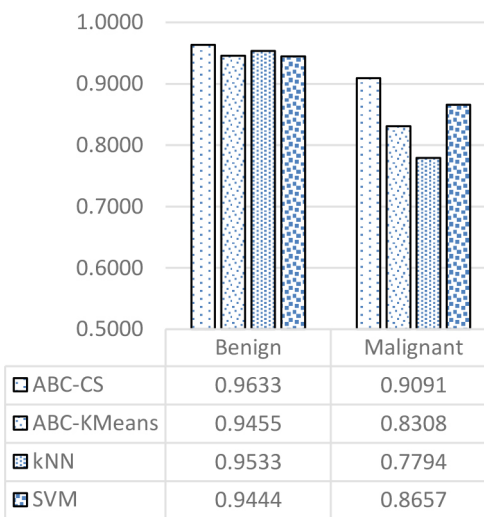


Fig. 5. The comparison of precision for each class

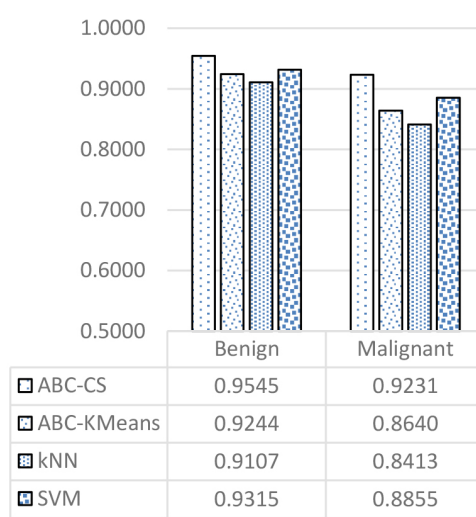


Fig. 7. The comparison of the F1-score for each class

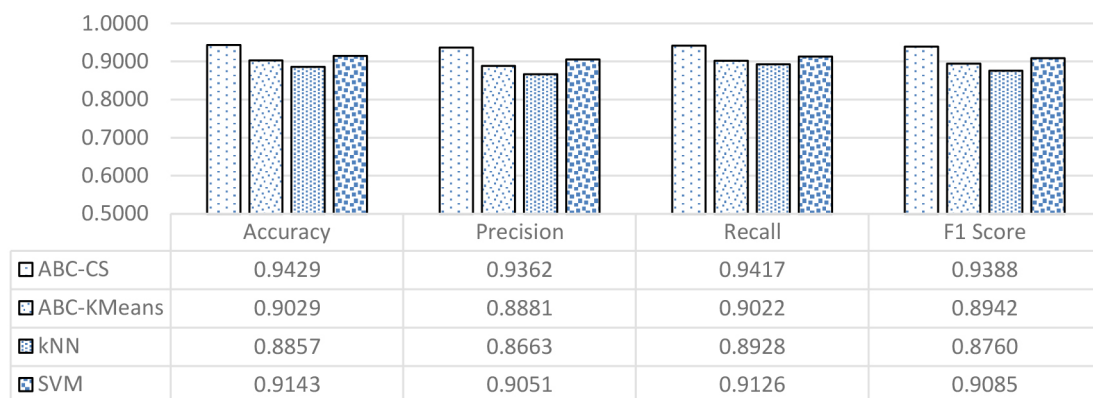


Fig. 8. The comparison of classification accuracy percentages, precision, recall, and F1 score of techniques

To evaluate the performance of our ABC-CS method on a large dataset, we used synthetic datasets by duplicating all records to produce datasets up to about 699,000 records. All methods in this experiment were programmed in C++, and all experiments were

run on a Raspberry Pi 4 Model B cluster of 3 boards, each with a quad-core CPU running at 1.8 GHz and 8 GB of memory. The operating system running on all boards was Ubuntu version 22.04.3 with MPICH2 version 4.1.2.

To assess the quality of clustering in this large dataset, we evaluated the outcomes produced by the parallel ABC-CS algorithm and compared them with those of the parallel kNN algorithm [38] and the parallel SVM algorithm [39]. The clustering quality results, reflecting the F1 score achieved by these algorithms on this dataset, are depicted in Fig. 9.

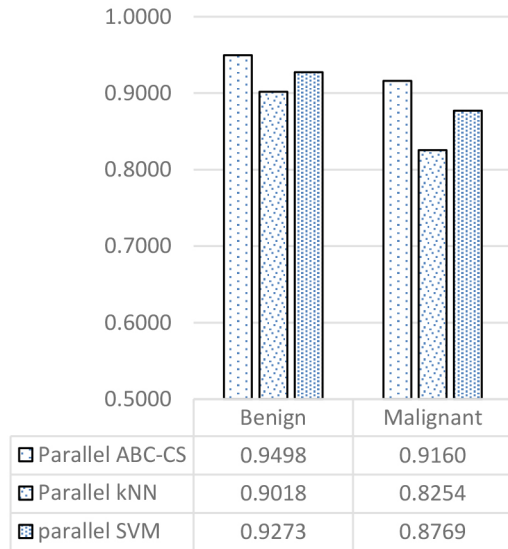


Fig. 9. The comparison of the F1 score for each class on the large dataset of parallel techniques

Fig. 9 illustrates the favorable performance of our parallel ABC-CS algorithm in the large dataset. The F1 scores obtained using the parallel ABC-CS method surpass those of the parallel kNN and parallel SVM methods, indicating that our proposed method achieves the best quantitative performance.

In the context of parallel computing [40], scalability refers to a system's ability to increase overall throughput as additional resources (processors) are added to handle increased workloads. Speedup, on the other hand, measures how much faster a parallel algorithm is than its corresponding sequential algorithm. The speedup (S_p) can be calculated using (6):

$$S_p = \frac{T}{T_N} \quad (6)$$

where the T is the running time using one processor and T_N is the running time using N processors.

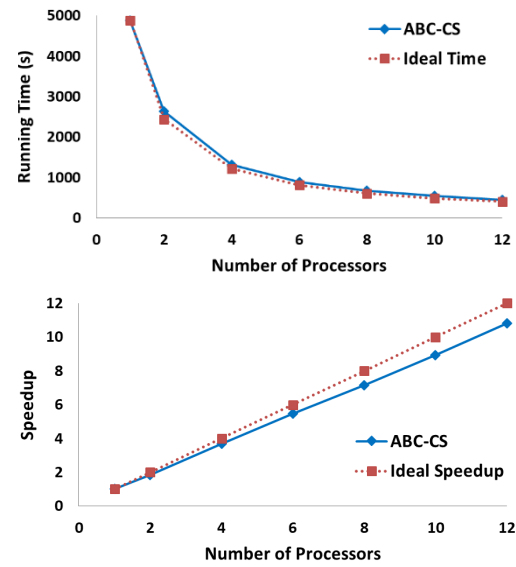


Fig. 10. Running time and speedup results of ABC-CS with 12 Raspberry Pi cluster processors

The scalability of our proposed method on the large dataset is shown in Fig. 10. As expected, increasing the number of processors used in our algorithm tends to reduce the runtime without sacrificing accuracy. Note that the ideal speedup is linear, in which the performance improvement is directly proportional to the number of processors. Thus, adding N processors to a task would reduce its execution time by a factor of N .

VI. DISCUSSION

This study demonstrates the feasibility of using the Artificial Bee Colony (ABC) algorithm with Cosine Similarity on a Raspberry Pi Cluster for medical data classification. The results show promising performance; however, the approach has limitations. Scalability with very large datasets may be constrained by the Raspberry Pi's limited processing power and memory, which can affect execution speed and efficiency.

Compared with previous studies, which primarily use high-performance computing, this work highlights the potential of low-cost, distributed hardware for medical data analysis. The integration of Cosine Similarity improved feature selection and classification

accuracy, supporting prior findings on similarity-based optimization while offering a novel application in embedded systems.

Future research could explore applications to medical imaging datasets, integration with deep learning models, and hybrid optimization techniques to enhance performance and robustness. Overall, this study provides a foundation for cost-effective and scalable medical data analysis, with clear directions for further investigation.

VII. CONCLUSIONS

This research aims to enhance medical data classification through the use of a new method called “ABC-CS”, which stands for Artificial Bee Colony with Cosine Similarity. The key contribution of this work is the development of an effective approach to optimize class centroids in a supervised manner using the ABC-CS technique. The performance of the proposed method was compared with well-established classification techniques, including k-NN, SVM, and ABC-KMeans, on the Breast Cancer Wisconsin dataset from the UCI machine learning repository. The results showed that the proposed ABC-CS method achieved superior classification accuracy compared to the other methods. Furthermore, the results obtained with the proposed method on Raspberry Pi clusters show that the parallel ABC-CS method is well-suited to handling massive amounts of data while maintaining the quality of the classification results. It can be concluded that the ABC-CS method offers a promising solution for enhancing medical data classification.

REFERENCES

- [1] H. Khan and M. Javaid, “Big data applications in medical field: A literature review,” *J. Ind. Integr. Manag.*, vol. 6, no. 1, pp. 53-69, Jul. 2021.
- [2] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, and A. T. Chronopoulos, “Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions,” *Neurocomputing*, vol. 276, pp. 2-22, Feb. 2018.
- [3] L. D. L. Fuente-Tomas et al., “Classification of patients with bipolar disorder using k-means clustering,” *PLoS One*, vol. 14, no. 1, p. e0210314, Jan. 2019.
- [4] P. Nanglia, S. Kumar, A. N. Mahajan, P. Singh, and D. Rathee, “A hybrid algorithm for lung cancer classification using SVM and Neural Networks,” *ICT Express*, vol. 7, no. 3, pp. 335-341, Sep. 2021.
- [5] A. Hamed, A. Sobhy, and H. Nassar, “Accurate classification of COVID-19 based on incomplete heterogeneous data using a KNN variant algorithm,” *Arab. J. Sci. Eng.*, vol. 46, no. 9, pp. 8261-8272, Sep. 2021.
- [6] M. Z. Alam, M. S. Rahman, and M. S. Rahman, “A random forest based predictor for medical data classification using feature ranking,” *Inf. Med. Unlocked*, vol. 15, p. 100180, Apr. 2019.
- [7] R. A. Haraty, M. Dimishkieh, and M. Masud, “An enhanced K-means clustering algorithm for pattern discovery in healthcare data,” *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 6, p. 615740, Jun. 2015.
- [8] Z. Wang, J. Na, and B. Zheng, “An improved kNN classifier for epilepsy diagnosis,” *IEEE Access*, vol. 8, pp. 100022-100030, May 2020.
- [9] J. Cervantes, F. García-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” *Neurocomputing*, vol. 408, pp. 189-215, May 2020.
- [10] X. Fan et al., “Review and classification of bio-inspired algorithms and their applications,” *J. Bionic Eng.*, vol. 17, no. 3, pp. 611-631, Aug. 2020.
- [11] A. Yadav and D. K. Vishwakarma, “A comparative study on bio-inspired algorithms for sentiment analysis,” *Clust. Comput.*, vol. 23, no. 4, pp. 2969-2989, 2020.
- [12] F. Valdez, O. Castillo, and P. Melin, “Bio-inspired algorithms and its applications for optimization in fuzzy clustering,” *Algorithms*, vol. 14, no. 4, p. 122, Dec. 2021.
- [13] V. Rajput, P. Mulay, and C. M. Mahajan, “Bio-inspired algorithms for feature engineering: Analysis, applications and future research directions,” *Inf. Discov. Deliv.*, vol. 53, no. 1, pp. 56-71, Jan. 2025.
- [14] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *J. Big Data*, vol. 6, no. 1, pp. 1-18, Dec. 2019.
- [15] A. K. Dubey, A. Kumar, and R. Agrawal, “An efficient ACO-PSO-based framework for data classification and preprocessing in big data,” *Evol. Intell.*, vol. 14, no. 2, pp. 909-922, Jun. 2021.
- [16] R. Divya and R. S. S. Kumari, “Genetic algorithm with logistic regression feature selection for Alzheimer’s disease classification,” *Neural Comput. Appl.*, vol. 33, no. 14, pp. 8435-8444, Dec. 2021.
- [17] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Tech. Rep. TR06, Erciyes Univ., Eng. Fac., Comput. Eng. Dept., Turkey, 2005.
- [18] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: Artificial Bee Colony (ABC) algorithm and applications,” *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21-57, 2014.
- [19] B. Akay and D. Karaboga, “A survey on the applications of Artificial Bee Colony in signal, image, and video processing,” *Signal Image Video Process.*, vol. 9, no. 4, pp. 967-990, Mar. 2015.
- [20] B. Akay, D. Karaboga, B. Gorkemli, and E. Kaya, “A survey on the Artificial Bee Colony algorithm variants for binary, integer and mixed integer programming problems,” *Appl. Soft Comput.*, vol. 106, p. 107351, Jul. 2021.
- [21] A. Banharsakun, “Artificial bee colony algorithm for solving the knight’s tour problem,” in *Proc. Int. Conf. Intell. Comput. Optim.*, 2018, pp. 129-138.
- [22] A. Banharsakun, “Artificial bee colony algorithm for content-based image retrieval,” *Comput. Intell.*, vol. 36, no. 1, pp. 351-367, Jan. 2020.
- [23] A. Banharsakun, “Low-Light Image Enhancement with Artificial Bee Colony Method,” in *Proc. Int. Conf. Intell. Comput. Optim.*, Jan. 2021, pp. 3-13.
- [24] L. Syed, S. Jabeen, S. Manimala, and A. Alsaedi, “Smart healthcare framework for ambient assisted living using IoMT and big data analytics techniques,” *Future Gener. Comput. Syst.*, vol. 101, pp. 136-151, Jun. 2019.
- [25] P. Galetsi, K. Katsaliaki, and S. Kumar, “Big data analytics in health sector: Theoretical framework, techniques and prospects,” *Int. J. Inf. Manage.*, vol. 50, pp. 206-216, Feb. 2020.
- [26] L. Wang and C. A. Alexander, “Big data analytics in medical engineering and healthcare: Methods, advances and challenges,” *J. Med. Eng. Technol.*, vol. 44, no. 6, pp. 267-283, Jun. 2020.
- [27] M. D. Mudaliar and N. Sivakumar, “IoT-based real-time energy monitoring system using Raspberry Pi,” *Internet Things*, vol. 12, p. 100292, Dec. 2020.

- [28] D. K. Dewangan and S. P. Sahu, "Deep learning-based speed bump detection model for intelligent vehicle system using Raspberry Pi," *IEEE Sens. J.*, vol. 21, no. 3, pp. 3570-3578, Feb. 2020.
- [29] V. Gonzalez-Huitron et al., "Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4," *Comput. Electron. Agric.*, vol. 181, p. 105951, Feb. 2021.
- [30] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition*. Massachusetts: Morgan Kaufmann Publishers, 2011, pp. 83-124.
- [31] L. Jiang and Y. Wang, "A wind power forecasting model based on data decomposition and cross-attention mechanism with cosine similarity," *Electr. Power Syst. Res.*, vol. 229, p. 110156, Jan. 2024.
- [32] M. Ahmad and M. Mazzara, "Scsnet: Sharpened cosine similarity-based neural network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1-4, Feb. 2024.
- [33] R. W. Nash, M. R. Crusoe, M. Kontak, and N. Brown, "Supercomputing with MPI meets the Common Workflow Language standards: An experience report," in *Proc. IEEE/ACM Workflows Support Large-Scale Sci.*, 2020, pp. 17-24.
- [34] R. Ehsani and F. Drabløs, "Robust distance measures for kNN classification of cancer cata," *Cancer Inform.*, vol. 19, pp. 1-9, Oct. 2020.
- [35] S. Huang et al., "Applications of Support Vector Machine (SVM) learning in cancer genomics," *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41-51, Dec. 2018.
- [36] M. Krishnamoorthi and A. M. Natarajan, "ABK-means: An algorithm for data clustering using ABC and K-means algorithm," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 4, pp. 383-391, Oct. 2013.
- [37] W. Wolberg, "Breast cancer wisconsin (original)." *UCI Machine Learning Repository*. 1990. [Online]. Available: <https://doi.org/10.24432/C5HP4Z> [Accessed: Sep. 25, 2025].
- [38] S. Athani, S. Joshi, B. A. Rao, S. Rai, and N. G. Kini, "Parallel implementation of kNN algorithm for breast cancer detection," in *Proc. Evol. Comput. Intell. FICTA 2020*, 2020, pp. 475-483.
- [39] Z. Kang, N. Xiao, Z. Chen, Y. Ou, and X. Li, "An improved parallel SVM algorithm on distributed system," in *Proc. Int. Conf. CyberC*, 2020, pp. 204-209.
- [40] D. Jaiswal and P. Kumar, "A survey on parallel computing for traditional computer vision," *Concurrency Comput.: Pract. Exp.*, vol. 34, no. 4, p. e6638, Sep. 2022.



Anan Banharnsakun is an associate professor in the Department of Computer Engineering, Kasetsart University Sriracha Campus, Thailand. He received a Ph.D. degree in Electrical and Computer Engineering from King Mongkut's University of Technology Thonburi, Thailand. His research interests include biologically inspired computational intelligence, robust optimization, pattern recognition, and high-performance computing.