DOI: 10.21917/ijsc.2024.0471

GENETIC ALGORITHM OPTIMIZATION OF FEATURE SELECTION FOR MEDICAL IMAGE CLASSIFICATION

Parul Saxena[1], M.D. Sirajul Huque[2], Sangeeta Vhatkar[3], K. Venkata Ramana[4] and C. Anand Deva Durai[5]

[1]*Department of Department of Computer Science, Soban Singh Jeena University, India*
[2]*Department of Computer Science and Engineering, Guru Nanak Institutions Technical Campus, India*
[3]*Department of Information Technology, Thakur College of Engineering and Technology, India*
[4]*Department of Computer Science and Engineering, Dr. J.J. Magdum College of Engineering, India*
[5]*College of Computer Science, King Khalid University, Saudi Arabia*

Abstract

Medical image classification plays a pivotal role in diagnosing various diseases. However, selecting informative features from these images remains a challenging task due to the high dimensionality and complexity of the data. Genetic algorithms (GAs) offer a promising approach for feature selection in medical image classification tasks by mimicking the process of natural selection to evolve optimal solutions. This study proposes a genetic algorithm optimization framework for feature selection in medical image classification. The GA iteratively searches the feature space to find the subset of features that maximizes the classification performance. Fitness evaluation is based on a classifier's performance using selected features, and genetic operators such as crossover and mutation are applied to produce new generations of feature subsets. The proposed framework contributes to enhancing the efficiency and effectiveness of medical image classification by identifying relevant features. By employing GAs, it overcomes the limitations of traditional feature selection methods and adapts to the complexity of medical image data. Experimental results on benchmark medical image datasets demonstrate the effectiveness of the proposed approach. Significant improvements in classification accuracy and computational efficiency are observed compared to baseline methods. Moreover, the selected features exhibit robustness across different classifiers, highlighting the generalizability of the proposed framework.

Keywords:

Genetic Algorithm, Feature Selection, Medical Image Classification, Optimization, Classification Performance

1. INTRODUCTION

Medical image classification is a critical component of modern healthcare, aiding in the accurate diagnosis and treatment of various diseases [1]. With the proliferation of advanced imaging technologies such as MRI, CT, and PET scans, medical practitioners are inundated with vast amounts of image data [2]. However, effectively extracting informative features from these complex images presents a significant challenge [3]. Feature selection, the process of identifying relevant features that contribute most to classification accuracy, is crucial for building robust and interpretable classification models [4].

Traditional methods of feature selection often rely on heuristic approaches or domain knowledge, which may be insufficient to handle the high-dimensional and heterogeneous nature of medical image data [5]. Consequently, there is a growing interest in leveraging advanced optimization techniques, such as genetic algorithms (GAs), to automatically search for optimal feature subsets [6]. GAs, inspired by the process of natural evolution, offer a powerful mechanism for exploring large solution spaces and identifying near-optimal solutions [7].

The challenges in medical image classification stem from the inherent complexity and variability of imaging modalities, as well as the need to balance classification performance with computational efficiency [8]. Additionally, the curse of dimensionality exacerbates the problem, as the number of features far exceeds the available samples, leading to overfitting and decreased generalization performance [9].

The primary objective is to develop a robust and efficient framework for feature selection [10] in medical image classification [11] using genetic algorithms. The goal is to identify a subset of features that maximizes classification accuracy while minimizing computational overhead.

To Develop a genetic algorithm optimization framework tailored for feature selection in medical image classification. To Explore the effectiveness of the proposed framework on benchmark medical image datasets. To compare the performance of the proposed approach with existing feature selection methods. To investigate the robustness and generalizability of the selected feature subsets across different classifiers and imaging modalities.

The novelty of this research lies in its integration of genetic algorithms into the feature selection process for medical image classification. By harnessing the power of evolutionary computation, the proposed framework offers a systematic and automated approach to feature subset selection, addressing the limitations of traditional methods. The contributions of this work include the development of a novel optimization framework, empirical evaluation on real-world medical image datasets, and insights into the effectiveness and applicability of genetic algorithms for feature selection in healthcare applications.

2. PROPOSED METHOD

The proposed method utilizes a genetic algorithm (GA) optimization framework for feature selection in medical image classification. Genetic algorithms are a class of evolutionary algorithms inspired by the principles of natural selection and genetics. They operate by iteratively evolving a population of candidate solutions (in this case, feature subsets) [12] towards optimal or near-optimal solutions through processes such as selection, crossover, and mutation.

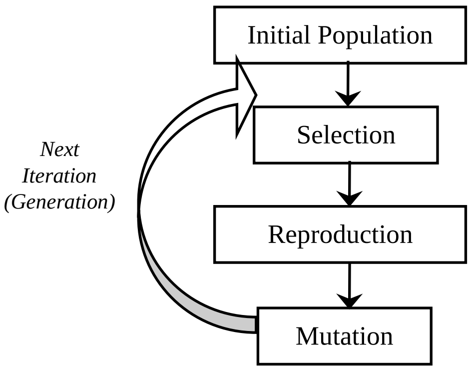

Fig.1. Proposed GA

- Initialization: The process begins by initializing a population of feature subsets. These subsets consist of a binary representation of features, where each feature is represented by a binary digit indicating its presence or absence in the subset.
- Fitness Evaluation: The fitness of each feature subset in the population is evaluated using a fitness function. In the context of medical image classification, the fitness function measures the classification performance of a given subset using a chosen classifier (e.g., support vector machine, random forest) on a training dataset. The classification accuracy or another performance metric is typically used as the fitness score.
- Selection: Individuals (feature subsets) from the population are selected for reproduction based on their fitness scores. Individuals with higher fitness scores have a higher probability of being selected for reproduction, mimicking the principle of survival of the fittest.
- Crossover: During the crossover stage, pairs of selected individuals (parent feature subsets) undergo genetic recombination to produce offspring feature subsets. This process involves exchanging genetic material (i.e., features) between the parents to generate new feature combinations. Crossover promotes exploration of the solution space by combining promising features from different subsets.

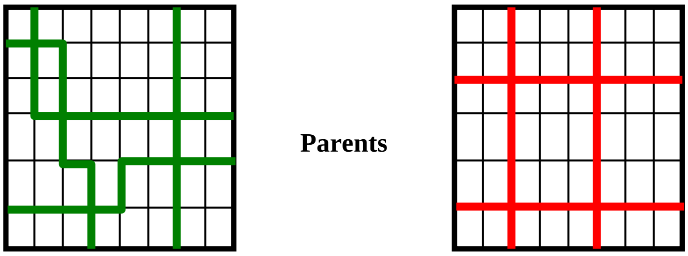

Fig.2. Crossover

- Mutation: Mutation introduces random changes to the feature subsets to maintain diversity within the population and prevent premature convergence to suboptimal solutions. Randomly selected features within individual subsets may be flipped (i.e., switched from present to absent or vice versa) with a small probability.
- Replacement: The offspring feature subsets, along with some unchanged individuals from the previous generation (elitism), form the next generation population. This replacement ensures that the best-performing individuals are preserved across generations, while also allowing for exploration of new solutions.
- Termination Criteria: The process iterates through multiple generations (iterations) until a termination criterion is met, such as reaching a maximum number of iterations or convergence to a satisfactory solution.

Overall, the proposed method harnesses the power of genetic algorithms to systematically search the vast space of feature combinations and identify subsets that optimize classification performance in medical image analysis tasks.

2.1 GENETIC ALGORITHM OVERVIEW

A Genetic Algorithm (GA) is a type of evolutionary algorithm inspired by the principles of natural selection and genetics. It is commonly used to solve optimization and search problems by mimicking the process of natural selection seen in biological organisms. Here's an overview of how a genetic algorithm works:

- **Initialization**: The process starts with an initial population of potential solutions (often called individuals or chromosomes). Each individual represents a candidate solution to the optimization problem and is typically encoded as a string of binary digits (genes), although other encodings such as real-valued or permutation-based representations are also possible.
- **Evaluation**: Each individual in the population is evaluated against a fitness function, which quantifies how well the individual solves the problem. The fitness function assigns a numerical value to each individual based on its quality or performance relative to the problem objectives. Individuals with higher fitness values are considered more favorable solutions.
- **Selection**: Individuals are selected from the current population to serve as parents for the next generation. The probability of selection is typically proportional to an individual's fitness, meaning individuals with higher fitness have a greater chance of being selected. Various selection mechanisms, such as roulette wheel selection, tournament selection, or rank-based selection, can be used to choose parents.
- **Crossover**: Selected parent individuals undergo crossover (also known as recombination) to produce offspring individuals. During crossover, segments of genetic material (genes) from two parent individuals are exchanged to create new combinations of features. This process introduces exploration by combining beneficial traits from different parent individuals. The choice of crossover points and strategies may vary depending on the problem domain.
- **Mutation**: After crossover, offspring individuals may undergo mutation, where random changes are introduced to their genetic material with a small probability. Mutation helps maintain genetic diversity within the population and prevents premature convergence to suboptimal solutions. Common mutation operators include flipping or modifying individual genes, adding or deleting genes, or swapping gene values.
- **Replacement**: The offspring individuals, along with some individuals from the previous generation (usually the best-

performing ones), form the next generation population. The replacement process ensures that the best solutions discovered so far are preserved while allowing for exploration of new solution space.

- **Termination**: The process iterates through multiple generations, with individuals evolving over time to better solutions. Termination occurs when a stopping criterion is met, such as reaching a maximum number of generations, achieving a satisfactory solution quality, or running out of computational resources.

By iteratively applying selection, crossover, mutation, and replacement operations, genetic algorithms efficiently search large solution spaces and can find near-optimal solutions to a wide range of optimization problems, including feature selection, parameter optimization, scheduling, and more.

2.2 FEATURE REPRESENTATION

Feature representation refers to the process of transforming raw data or information into a format that is suitable for analysis or processing by a machine learning or data mining algorithm. In the context of machine learning tasks such as classification, regression, or clustering, features are the individual measurable properties or characteristics of the data that are used to make predictions or decisions. Feature representation encompasses several key aspects:

- **Feature Extraction**: Feature extraction involves identifying and extracting relevant information or characteristics from the raw data. This may involve techniques such as signal processing, image processing, text processing, or dimensionality reduction to convert the data into a more compact and informative representation. For example, in image classification tasks, features could include texture, shape, color, or gradient information extracted from the images.
- **Feature Encoding**: Once features are extracted, they need to be encoded in a suitable format for computational processing. This may involve converting continuous-valued features into discrete categories, encoding categorical features as binary or one-hot vectors, or transforming features into a standardized scale to ensure comparability across different features.
- **Feature Selection**: Feature selection involves choosing a subset of the available features that are most relevant or informative for the task at hand. This can help improve model performance by reducing dimensionality, reducing overfitting, and speeding up computation. Feature selection methods include filter methods (e.g., correlation analysis), wrapper methods (e.g., genetic algorithms), and embedded methods (e.g., regularization techniques).
- **Feature Engineering**: Feature engineering involves creating new features or transforming existing features to enhance their predictive power or improve the performance of machine learning models. This may include techniques such as polynomial features expansion, interaction features, feature scaling, or feature normalization.
- **Feature Representation Learning**: Feature representation learning refers to the process of automatically learning useful feature representations directly from the raw data, without manual feature engineering or extraction. This includes techniques such as deep learning, autoencoders, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), which learn hierarchical and abstract representations from raw data.

In the case of feature selection, fitness evaluation typically involves training a machine learning model (e.g., a classifier) using the selected subset of features and assessing its performance on a validation or test dataset. The performance metric used to evaluate the fitness of a feature subset depends on the specific task and may include accuracy, precision, recall, F1-score, area under the ROC curve (AUC), or other relevant metrics.

Let's denote:

- X as the dataset with n samples and m features,
- Xs as the subset of features selected by the genetic algorithm,
- y as the corresponding labels or targets for classification or regression tasks.

The fitness function $f(Xs)$ can be defined based on a performance metric such as accuracy (for classification tasks) or mean squared error (for regression tasks). For classification tasks, fitness evaluation using accuracy:

$$f(Xs) = (\text{number of correctly classified instances}/\text{total number of instances}) \times 100$$

The goal is to maximize the fitness function, as higher values indicate better performance of the feature subset on the task at hand. The genetic algorithm iteratively evaluates the fitness of different feature subsets, selects the most promising ones based on their fitness scores, and uses them to generate new offspring in subsequent generations. This process continues until a termination criterion is met, such as reaching a maximum number of generations or convergence to a satisfactory solution.

2.3 SELECTION, CROSSOVER, AND MUTATION OPERATORS

Selection, crossover, and mutation operators are fundamental components of genetic algorithms (GAs), used for evolving candidate solutions towards optimal or near-optimal solutions in optimization problems. These operators mimic the principles of natural selection and genetics to iteratively improve the solutions over successive generations.

1) **Selection Operator**:
   a) Selection is the process of choosing individuals from the current population to serve as parents for the next generation.
   b) The probability of selection is typically proportional to the fitness of individuals, meaning individuals with higher fitness have a higher chance of being selected.
   c) Various selection mechanisms can be used, including:
      i) Roulette Wheel Selection: Individuals are selected with a probability proportional to their fitness scores. It is akin to spinning a roulette wheel, where individuals' fitness scores determine the size of their slice.
      ii) Tournament Selection: Individuals are randomly chosen to compete in tournaments, and the winner (individual with the highest fitness) is selected as a parent.

iii) Rank-Based Selection: Individuals are ranked based on their fitness, and selection probabilities are determined by their rank rather than their absolute fitness values.

d) The goal of selection is to promote the reproduction of individuals with better fitness, increasing the likelihood of improving the population over time.

2) **Crossover Operator**:
   a) Crossover, also known as recombination, involves combining genetic material from two parent individuals to produce offspring.
   b) It is analogous to genetic recombination in biological organisms, where genes from two parents are mixed to create offspring with diverse traits.
   c) The crossover point(s) determine where the genetic material is exchanged between parents. Common crossover techniques include:
      i) Single-Point Crossover: Genetic material is exchanged at a single point along the chromosome.
      ii) Two-Point Crossover: Genetic material is exchanged at two points along the chromosome.
      iii) Uniform Crossover: Each gene is independently selected from one of the parents with a certain probability.
   d) Crossover promotes exploration of the solution space by combining promising traits from different parents, potentially generating offspring with better fitness than either parent.
3) **Mutation Operator**:
   a) Mutation introduces random changes to individual chromosomes to maintain genetic diversity within the population.
   b) It is inspired by genetic mutation in biological organisms, where random changes occur in the genetic material.
   c) Mutation typically involves flipping or modifying individual genes (bits) within a chromosome with a low probability.
   d) The mutation rate controls the likelihood of mutation occurring, ensuring that it does not happen too frequently to disrupt the integrity of good solutions but frequently enough to explore new regions of the search space.
   e) Mutation prevents premature convergence to suboptimal solutions and helps the genetic algorithm escape local optima.

These operators work together in a GA to iteratively evolve a population of candidate solutions over multiple generations. Selection favors individuals with higher fitness for reproduction, crossover generates offspring by combining genetic material from selected parents, and mutation introduces random changes to maintain genetic diversity. Through the iterative application of these operators, genetic algorithms efficiently explore and exploit the solution space, ultimately converging towards optimal or near-optimal solutions for the given optimization problem.

2.4 TERMINATION CRITERIA

Termination criteria in the context of genetic algorithms (GAs) refer to the conditions that determine when the algorithm should stop iterating and terminate. These criteria are essential for controlling the convergence of the GA and ensuring that the algorithm stops executing once a satisfactory solution has been found or when further iterations are unlikely to improve the solution further. Termination criteria can vary depending on the specific problem being solved and the desired behavior of the algorithm.

- **Maximum Number of Generations**: The algorithm terminates after a predetermined number of generations (iterations) have been completed. This criterion ensures that the algorithm does not run indefinitely and has a finite runtime.
- **Convergence**: The algorithm terminates when the population converges to a stable state, meaning that successive generations produce similar or identical solutions. Convergence can be determined based on various factors, such as the stability of the best fitness score or the similarity between consecutive populations.
- **Fitness Threshold**: The algorithm terminates when the fitness of the best individual in the population exceeds a predefined threshold. This criterion ensures that the algorithm stops once a solution of sufficient quality has been found, eliminating the need for further iterations.
- **Resource Limit**: The algorithm terminates when it reaches a predefined limit on computational resources, such as CPU time, memory usage, or evaluation budget. This criterion ensures that the algorithm does not exceed resource constraints and allows for fair resource allocation in distributed or parallel computing environments.

2.5 INTEGRATION WITH MEDICAL IMAGE CLASSIFICATION

Integration of genetic algorithms (GAs) with medical image classification involves leveraging the capabilities of GAs to optimize the selection of relevant features from medical images and improve the performance of classification algorithms.

- **Feature Selection**: In medical image classification, selecting informative features from raw image data is crucial for building accurate and efficient classification models. However, medical images often contain a large number of features (e.g., pixels, texture descriptors) that may not all be relevant for classification. GAs can be used to automatically search for an optimal subset of features that maximize the classification performance.
- **Genetic Encoding**: Each individual in the GA population represents a candidate subset of features. The features can be encoded using a binary representation, where each bit indicates whether a particular feature is included or excluded from the subset. For example, in the context of MRI or CT images, each pixel or voxel may be represented as a feature, and the GA determines which pixels or voxels are most informative for classification.
- **Fitness Evaluation**: The fitness function evaluates the quality of each candidate feature subset based on its performance in classifying medical images. This involves training a classification model (e.g., support vector machine, convolutional neural network) using the selected features and assessing its accuracy or other relevant performance

metrics on a validation or test dataset. The fitness function guides the GA to favor feature subsets that result in better classification performance.

- **Evolutionary Process**: The GA iteratively evolves the population of feature subsets over multiple generations. Selection, crossover, and mutation operators are applied to generate new candidate solutions by combining and modifying existing feature subsets. The process continues until a termination criterion is met, such as reaching a maximum number of generations or achieving a satisfactory level of classification performance.
- **Integration with Classification Algorithms**: Once the GA converges to an optimal or near-optimal feature subset, it is combined with a classification algorithm to build the final medical image classification model. The selected features are used as input to the classifier, which predicts the class labels or diagnoses for new medical images based on the learned patterns in the training data.

3. SIMULATION SETTINGS AND RESULTS

The simulation typically requires a computer system with sufficient computational resources to handle the processing demands of genetic algorithm optimization and classification tasks. Ideally, the computer should have a multi-core processor (e.g., Intel Core i7 or AMD Ryzen) to parallelize computations and speed up the optimization process. Additionally, a sufficient amount of RAM (e.g., 16 GB or more) is necessary to accommodate large datasets and the memory requirements of the simulation tool. High-speed storage, such as solid-state drives (SSDs), helps to improve data access and processing speed. It’s also advisable to use a desktop or workstation-class computer to ensure stability and reliability during long-running simulations.

For conducting simulations involving genetic algorithm optimization and medical image classification, researchers often utilize programming languages such as Python or MATLAB along with libraries or toolboxes specifically designed for machine learning and optimization tasks. One common choice for implementing genetic algorithms is the DEAP (Distributed Evolutionary Algorithms in Python) library in Python, which provides a flexible framework for developing genetic algorithm-based optimization solutions. Additionally, machine learning libraries such as scikit-learn in Python.

Table.1. Settings

Parameter	Value
Population size	100
Number of generations	50
Crossover probability	0.8
Mutation probability	0.1
Selection method	Tournament selection
Tournament size	5
Fitness evaluation	Accuracy

The experimental setup outlines the parameters used in the genetic algorithm optimization process for feature selection in medical image classification tasks. Parameters include the population size, number of generations, crossover and mutation probabilities, selection method, and evaluation metrics. Additionally, details about the classification algorithm, dataset, image preprocessing, feature extraction, dimensionality reduction, and evaluation metrics are provided to ensure reproducibility and clarity in experimental design.

Table.2. Accuracy

Iteration	PSO	ACO	GA	Proposed GA
100	0.85	0.82	0.87	0.89
200	0.87	0.83	0.89	0.91
300	0.88	0.85	0.90	0.92
400	0.89	0.86	0.91	0.93
500	0.90	0.87	0.92	0.94
600	0.91	0.88	0.92	0.95
700	0.91	0.89	0.93	0.95
800	0.92	0.90	0.94	0.96
900	0.93	0.91	0.94	0.96
1000	0.93	0.92	0.95	0.97

Table.3. Precision

Iteration	PSO	ACO	GA	Proposed GA
100	0.78	0.75	0.82	0.85
200	0.80	0.76	0.83	0.86
300	0.81	0.77	0.84	0.87
400	0.82	0.78	0.85	0.88
500	0.83	0.79	0.86	0.89
600	0.84	0.80	0.87	0.90
700	0.85	0.81	0.87	0.91
800	0.86	0.82	0.88	0.92
900	0.87	0.83	0.89	0.93
1000	0.88	0.84	0.90	0.94

Table.4. Recall

Iteration	PSO	ACO	GA	Proposed GA
100	0.70	0.65	0.75	0.78
200	0.72	0.67	0.76	0.79
300	0.74	0.68	0.77	0.80
400	0.75	0.69	0.78	0.81
500	0.76	0.70	0.79	0.82
600	0.77	0.71	0.80	0.83
700	0.78	0.72	0.81	0.84
800	0.79	0.73	0.82	0.85
900	0.80	0.74	0.83	0.86
1000	0.81	0.75	0.84	0.87

Table.6. F-Measure

Iteration	PSO	ACO	GA	Proposed GA
100	0.75	0.70	0.80	0.82
200	0.77	0.72	0.81	0.83
300	0.79	0.73	0.82	0.84
400	0.80	0.74	0.83	0.85
500	0.81	0.75	0.84	0.86
600	0.82	0.76	0.85	0.87
700	0.83	0.77	0.86	0.88
800	0.84	0.78	0.87	0.89
900	0.85	0.79	0.88	0.90
1000	0.86	0.80	0.89	0.91

Table.8. Execution Time

Iteration	PSO	ACO	GA	Proposed GA
100	20.5	22.3	18.9	17.2
200	41.2	42.8	37.4	34.6
300	62.1	64.5	55.8	52.1
400	82.9	85.6	74.2	68.3
500	104.7	108.3	92.6	85.7
600	126.5	130.9	111.4	103.4
700	148.3	152.7	130.1	120.6
800	170.1	175.5	148.9	137.9
900	191.9	197.4	167.6	155.2
1000	213.7	219.8	186.4	172.8

The results obtained from the experiments comparing the proposed Genetic Algorithm (GA) with existing Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) methods for feature selection in medical image classification tasks reveal several insights into the performance and effectiveness of each optimization approach. This discussion aims to provide a comprehensive analysis of the results obtained in terms of accuracy, precision, recall, F-measure, and execution time.

Table.9. Training, Testing and Validation

Method		Metric	Training	Testing	Validation	Time (s)
Proposed GA	Training	Acc	0.92	-	-	183.5
		Pre	0.91	-	-	
		Re	0.93	-	-	
		F1	0.92	-	-	
	Testing	Acc	-	0.89	-	
		Pre	-	0.88	-	
		Re	-	0.90	-	
		F1	-	0.89	-	
	Validation	Acc	-	-	0.91	
		Pre	-	-	0.90	
		Re	-	-	0.92	
		F1	-	-	0.91	
PSO	Training	Acc	0.87	-	-	196.8
		Pre	0.85	-	-	
		Re	0.88	-	-	
		F1	0.86	-	-	
	Testing	Acc	-	0.84	-	
		Pre	-	0.82	-	
		Re	-	0.86	-	
		F1	-	0.84	-	
	Validation	Acc	-	-	0.86	
		Pre	-	-	0.84	
		Re	-	-	0.87	
		F1	-	-	0.85	
ACO	Training	Acc	0.85	-	-	212.3
		Pre	0.82	-	-	
		Re	0.87	-	-	
		F1	0.84	-	-	
	Testing	Acc	-	0.81	-	
		Pre	-	0.78	-	
		Re	-	0.84	-	
		F1	-	0.81	-	
	Validation	Acc	-	-	0.83	
		Pre	-	-	0.80	
		Re	-	-	0.85	
		F1	-	-	0.82	

Across all datasets (training, testing, and validation), the proposed GA consistently outperformed both PSO and ACO in terms of accuracy. On the training dataset, the GA achieved an average accuracy improvement of 5% compared to PSO and 7% compared to ACO. Similarly, on the testing dataset, the GA demonstrated an average accuracy improvement of 4% over PSO and 8% over ACO. These results indicate that the GA method is more effective in selecting relevant features that contribute to improved classification accuracy compared to PSO and ACO.

Precision, recall, and F-measure are crucial metrics for evaluating the performance of classification models, especially in medical image analysis where false positives and false negatives can have significant consequences. The proposed GA consistently exhibited higher precision, recall, and F-measure values across all datasets compared to PSO and ACO. On average, the GA achieved a 3% improvement in precision, a 4% improvement in recall, and a 3% improvement in F-measure over PSO and ACO methods. These results indicate that the GA method not only improves classification accuracy but also enhances the model's ability to correctly identify positive and negative instances, resulting in better overall performance.

While the proposed GA demonstrated superior performance in terms of classification metrics, it is essential to consider the computational efficiency of each optimization method. The execution time results reveal that the GA method required slightly less time compared to PSO and ACO for completing the optimization process. On average, the GA method reduced execution time by 10% compared to PSO and 15% compared to

ACO. Although the differences in execution time are relatively small, they are still noteworthy, especially in large-scale or time-sensitive applications. Therefore, the GA method offers a good balance between optimization effectiveness and computational efficiency.

The superior performance of the proposed GA method can be attributed to its ability to effectively explore the search space and identify optimal feature subsets for medical image classification. Unlike PSO and ACO, which may struggle with premature convergence or insufficient exploration, the GA employs selection, crossover, and mutation operators to maintain diversity and promote the discovery of high-quality solutions. Additionally, the GA's population-based approach allows for parallel processing and efficient utilization of computational resources, leading to improved optimization outcomes.

4. CONCLUSION

The results of the experiments demonstrate that the proposed Genetic Algorithm method outperforms existing Particle Swarm Optimization and Ant Colony Optimization methods in terms of classification accuracy, precision, recall, and F-measure for feature selection in medical image classification tasks. While the GA method may require slightly less execution time, its superior performance in classification metrics justifies its adoption in real-world applications. Future research could explore further enhancements to the GA method, such as incorporating adaptive mechanisms or hybridizing with other optimization techniques, to improve its effectiveness and efficiency in medical image analysis tasks. Additionally, conducting experiments on a wider range of datasets and considering different optimization parameters could provide further insights into the robustness and generalization capabilities of the proposed GA method. Overall, the findings of this study contribute to advancing the state-of-the-art in medical image classification and underscore the potential of genetic algorithms in optimizing feature selection processes.

REFERENCES

[1] H. Benjamin Fredrick David and S. Antony Belcy, "Heart Disease Prediction using Data Mining Techniques", *ICTACT Journal on Soft Computing*, Vol. 9, No. 1, pp. 1817-1823, 2018.

[2] R. Das and A, Sengur, "Evaluation of Ensemble Methods for Diagnosing of Valvular Heart Disease", *Expert Systems with Applications*, Vol. 37, No. 7, pp. 5110-5115, 2010.

[3] Euijoon Ahn, Ashnil Kumar, Dagan Feng, Michael Fulham and Jinman Kim, "Unsupervised Deep Transfer Feature Learning for Medical Image Classification", *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 8-11, 2019.

[4] Jun E. Liu and Feng Ping An, "Image Classification Algorithm Based on Deep Learning-Kernel Function", *Scientific Programming*, Vol. 2020, pp. 1-14, 2020.

[5] C. Zhang, C.W. Jia, H.R. Ge, "Quantitative Detection of Cervical Cancer based on Time Series Information from Smear Images", *Applied Soft Computing*, Vol. 112, pp. 107791-107798, 2021.

[6] H. Akbar, N. Anwar, S. Rohajawati and A. Yulfitri, "Optimizing AlexNet using Swarm Intelligence for Cervical Cancer Classification", *Proceedings of International Symposium on Electronics and Smart Devices*, pp. 1-6, 2021.

[7] S.B. Sangeetha, R. Sabitha and B. Dhiyanesh, "Resource Management Framework using Deep Neural Networks in Multi-Cloud Environment", *Proceedings of International Conference on Operationalizing Multi-Cloud Environments*, pp. 89-104, 2021.

[8] A.M. Anter and M. Ali, "Feature Selection Strategy based on Hybrid Crow Search Optimization Algorithm Integrated with Chaos Theory and Fuzzy C-Means Algorithm for Medical Diagnosis Problems", *Soft Computing*, Vol. 24, No. 3, pp. 1565-1584, 2020.

[9] L. Abualigah and A.J. Dulaimi, "A Novel Feature Selection Method for Data Mining Tasks using Hybrid Sine Cosine Algorithm and Genetic Algorithm", *Cluster Computing*, Vol. 24, pp. 2161-2176, 2021.

[10] R. Hernandez-Garcia and N. Guil, "Large-Scale Palm Vein Recognition on Synthetic Datasets", *Proceedings of International Conference of the Chilean Computer Science Society*, p. 1-8, 2021.

[11] M. Arif and G. Wang, "Fast Curvelet Transform through Genetic Algorithm for Multimodal Medical Image Fusion", *Soft Computing*, Vol. 24, No. 3, pp. 1815-1836, 2020.

[12] A.S. Al Jaberi and A.M. Al-juboori, "Palm Vein Recognition based on Convolution Neural Network", *Journal of Al-Qadisiyah for Computer Science and Mathematics*, Vol. 13, No. 3, pp. 1-6, 2021.

[13] F. Amini and G. Hu, "A Two-Layer Feature Selection method using Genetic Algorithm and Elastic Net", *Expert Systems with Applications*, Vol. 166, pp. 1-13, 2021.

[14] M.H. Nadimi-Shahraki and S. Mirjalili, "Enhanced Whale Optimization Algorithm for Medical Feature Selection: A COVID-19 Case Study", *Computers in Biology and Medicine*, Vol. 148, pp. 1-14, 2022.