DESIGN AND IMPLEMENTATION OF NATURE-INSPIRED ALGORITHMS FOR BIOMEDICAL APPLICATIONS

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

by

PRERNA SHARMA

(2K19/PHD/CO/14)

Under the Supervision of

Prof. KAPIL SHARMA

Department of Information Technology, DTU

To the

Department of Computer Science & Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042. India

December, 2023

CERTIFICATE

Certified that Prerna Sharma (2K19/PHD/CO/14) has carried out their research work presented in this thesis entitled **"Design and Implementation of Nature-Inspired Algorithms for Biomedical Applications."** for the award of Doctor of Philosophy from Delhi Technological University, New Delhi, under my supervision. The thesis embodies results of original work and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Supervisor:

Prof. Kapil Sharma
Professor
Dept. of Information Technology
Delhi Technological University

Date: 14-12-2023

CANDIDATE'S DECLARATION

I hereby declare that the dissertation entitled **"Design and Implementation of Nature-Inspired Algorithms for Biomedical Applications"** to be submitted for the Degree of Doctor of Philosophy is my original work and the dissertation has not formed the basis for the award of any degree, diploma, associateship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: DTU, Delhi
Date:

This is to certify that the above statements made by the candidate is true.

Prerna Sharma
Roll No. 2K19/PHD/CO/14

ABSTRACT

Over the last few decades, the volume of data has become colossal. The process of attaining optimal solutions is progressively becoming more intricate due to the proliferation of data generation. The ability to process large volumes of data in a short period of time is facilitated by optimization algorithms. Intelligent metaheuristic algorithms have gained recognition for their ability to achieve optimal solutions in complex optimization problems, particularly when faced with multiple restrictions. Numerous unique algorithms are currently being researched to devise efficient methodologies for addressing these types of situations. These algorithm supports feature selection which aids to pick up appropriate and important features from the original feature space with the minimum redundancy and the highest discriminating capability. Now-a-days, algorithms are computationally intensive and time consuming. There is a need of an optimization technique to solve optimization problems that provide results timely as well as handle multidimensional datasets in the field of biomedical. Due to the presence of redundant features and the challenge of high dimensionality, the learning engine incurs a significant time cost, resulting in a decrease in the efficiency of the model. The utilization of application classification analysis is employed to enhance medical diagnostic decision-making processes and ultimately enhance the standard of care provided to the patients. Within the realm of biomedical applications, there exists a range of activities encompassing illness diagnosis and patient treatment. These activities involve the utilization of computer analysis to examine patient-related data, the application of clinical decision-making processes, the integration of medical informatics and the incorporation of artificial intelligence techniques. The biomedical application of disease diagnosis in the

healthcare system pertains to patients who are in a state that indicates the presence of a disease. The diagnostic procedure within the healthcare system is intricately connected to patients who have symptoms that are suggestive of a certain disease or condition. The expeditious identification and application of individualized pharmaceutical treatments have notably enhanced the overall well-being of patients, presenting prospective remedies for a multitude of ailments that affect the global populace. The proposed research targets to serve this unaddressed issue by using Nature Inspired Algorithms for dimensionality reduction on biomedical datasets. The nature-inspired algorithms tend to pick up features based on various feature selection approaches that tends to impact the target variable more effectively. The optimal features obtained by nature-inspired algorithms using various feature selection approaches greatly reduces the computational time and cost. The optimality of features could be evaluated against various performance measure parameters using machine learning classifiers. This can greatly contribute to diagnostic procedure by early diagnosis of the disease which can aid in timely dispense of treatment to the patient.

ACKNOWLEDGEMENT

I want to express my heartfelt gratitude to my worthy supervisor, **Prof. Kapil Sharma**, Professor, Department of Information Technology, Delhi Technological University, Delhi, India, for being a beacon of light and continuously guiding me through thick and thin. He is a treasure trove of all the qualities a pupil could have asked for in a mentor. His constant support, encouragement, immense knowledge and scientific perception lit up my way in the darkest times. I cannot imagine a better mentor for my Ph.D. study. His belief in me was always re-energizing and rejuvenating. Working with him has been a wonderful experience, pride, and bliss. I will never forget the fortitude, optimism, and encouragement he had counselled me throughout my research.

I will forever be thankful to **Prof. Prateek Sharma, Vice Chancellor**, Delhi Technological University, **Prof. Rajni Jindal, DRC Chairperson,** Department of Computer Science and Engineering, Delhi Technological University and all the faculty members of the CSE Department for their constructive comments and valuable feedback while ensuring the progress of my research work. I would also like to thank all the DTU staff, Delhi, for acquiescing me to embellish my technical skills and providing such a friendly working environment.

This thesis would never have borne fruit without the unconditional support of my family. My parents deserve a special mention here, who inculcated moral, ethical, and religious values in me. I would manifest my wholehearted appreciation, especially towards my father **Shri Vipin Sharma** and my father-in-law **Shri Ramesh Mann.**

They have encouraged me from the beginning of this degree and emboldened me to achieve every milestone throughout my research work. They have supported me in all respects in making this study a success. I want to extend my warmest thanks to my mother **Smt. Suman Sharma and my mother-in-law Smt. Sushila Devi** as they have inspired me in all the dimensions of life and instilled condolence in me to complete this journey.

I would also like to thank my husband, **Mr. Parminder Mann** for his wholehearted support and providing productive and stimulating ideas that motivated me to explore the tools and technologies for conducting the experiments during this research process. He helped me in all respects in making this study an accomplishment. Special mention to my sons, **Pratyaksh Mann, Kairav Mann and Kevin Mann** who have been a constant source of strength, motivation, and inspiration for me throughout this journey.

I want to thank all my PhD colleagues, with whom I have shared moments of deep anxiety but also of big excitement. Their presence was significant in a process often felt as tremendously solitaire.

I want to thank the almighty for giving me the strength and health to complete my work. I am indebted to so many for their encouragement and support.

PRERNA SHARMA

2K19/PHD/CO/14

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter introduces the basic concepts related to Design and Implementation of Nature-Inspired Algorithms for biomedical applications. The objectives of the research work are highlighted. Chapter wise thesis coverage is summarized at the end of the chapter.

1.1 Background of the Study

1.1.1 Optimization Problems

The optimization problems are generally encountered in all real-world problems and are usually challenging to solve due to the increase in complex analysis. An optimal solution to these NP-hard problems can be acquired using different optimization tools. Generally, these problems have no definite and efficient algorithms and often use different optimization techniques to determine the most competent algorithm by the method of trial and error. Furthermore, numerous innovative algorithms are currently under development in an effort to discover effective methods for addressing these intricate optimization problems. It is expected that these optimization algorithms developed will identify and solve various problems in the field of science and engineering effectively.

Optimal solution for these complex optimization problems is obtained from the intelligent metaheuristic algorithms. Further, metaheuristics imitates the behavior of animals as well as birds to solve complexity in nature that infers it to be nature-inspired algorithms. These metaheuristic intelligent algorithms are categorized as evolutionary, physical, bio-inspired, swarm-based and other nature-inspired algorithms. Each of these category deals with a certain portion of nature to handle the complex problems.

Swarm based algorithms are evaluated to be adaptive strategies considering that collective intelligence of swarms will handle the real-world problems by imitating the biological behavior of different animal species. Self-organization and division of work are considered important attributes of swarm intelligence. These swarm-based algorithms include five fundamental principles that are; quality principle, proximity principle, diverse principle, adaptability principle and stability principle.

The primary basis of optimization-based problems is to determine an optimal solution for a specified problem. However, it is difficult to determine the most efficient algorithm dueto the swift development of metaheuristic algorithms which makes tracking of the algorithmsdifficult. Researchers are not familiar with the recent developments, while the algorithms developed are gaining prominence as a tool to resolve these complex problems.

Due to limited knowledge of the domain, the new researchers often tend to "force-fit" the algorithms instead of exploring to obtain the most suitable one. Therefore, we review some popular metaheuristic algorithms exhaustively to identify their potential scope for different problems. Nature-Inspired Algorithms (NIAs) are the algorithms that originate their motivation from nature itself; the considerable expanse of algorithms and solutions are already conceptualized in nature; our task is to look around and adapt them to work out our problems [1].

1.1.2 Nature-Inspired Algorithms

Nature-Inspired Algorithms (NIAs) are fundamentally the algorithms for whom the nature becomes source of inspiration and motivation, nature is itself intellectualized with significant spread of algorithms and solutions buried in it; if we closely observe them, they could be of great help in figuring out our problems. Nature-inspired computing has got together numerous poles apart disciplines of biology, mathematics and computer science. These algorithms are based on the technique of bio-inspired computing which is pretty much progressive to new and different challenging techniques which are grounded on the principle of natures' evolution. Evolutionary Algorithms/Bio-inspired emulate the evolutionary act of creatures set up in nature. The search algorithms initiate with forming population using randomly generated solutions that tends to evolve throughout succeeding generations. Medical data when treated with various computational intelligence methods helps in getting better grip of efficiency in terms of speed and precision detection.

Fig. 1. 1: Taxonomy of Nature-Inspired Algorithms

In the medical field nature inspired techniques are categorized under two different heads by using single and hybrid methods and nature inspired algorithms dealing with medical data got exposure of various state- of-the- art methodologies [2], these days' applications of science and engineering likewise decision making, objective optimization and processing of the information etc. are making use of various techniques of nature inspired algorithm. Taxonomy of nature-inspired algorithms is depicted in Fig. 1. 1.

1.1.3 Feature Selection

Owing to redundancy of features and the burden of dimensionality, the learning engine reserves a considerable toll on time and the efficiency of the model drops [3]. Application classification analysis to support medical diagnostic decisions, improve the quality of patients' care etc. A subset of data sets containing vast volumes of data contained in medical records is chosen for instruction. If the training data set contains unrelated and inapt characteristics, the classification analysis could yield results that are less in precision and not as much as comprehensible as needed. The selection of the feature sub-sets is an extremely important pre-processing stage in the field of data extraction [4]. The enhancement of the health database increases the diagnostic efficiency. Some extra attributes can increase the computation time that has an effect on the accuracy of the diagnosis. Any data in the data collection may not be useful for diagnosis and may therefore be discarded prior to learning.

1.1.4 Biomedical Applications

Biomedical engineering is the process of gaining novel information and comprehension of biological systems by employing inventive and substantial utilization of experimental and analytical methods rooted in the field of engineering sciences. The advancement of biology and medicine is facilitated through the creation of novel technologies, algorithms, processes, and systems, which in turn enhance medical practice and the delivery of healthcare services.

Biomedical applications encompass the diagnosis and treatment of diseases, which involves the utilization of clinical decision making, medical informatics, artificial intelligence, and computer analysis of patient-related data. Biomedical application of disease diagnosis in healthcare system is associated with the patient that is in a disease-indicative state. The process of diagnosing within the healthcare system is closely linked to patients who exhibit symptoms indicative of a certain disease or condition. The quick diagnosis and implementation of personalized medicines have significantly improved the quality of life for patients, potentially offering solutions for numerous diseases that afflict the human population. The expedited identification and application of customized treatments have significantly enhanced the observed standard of living in patients and hold the potential to ultimately eradicate numerous ailments that afflict society.

Furthermore, there has been a notable acceleration in the accumulation of clinical data of greater depth and complexity in recent years, mostly driven by the "Big Data" movement within the healthcare industry. Hence, a notable development observed in the field of analytical science is the utilization of artificial intelligence and machine learning models to establish correlations between detected or imaged indicators obtained from patients and the process of diagnosis. Recent instances comprise a demonstration of an artificial intelligence system that exhibited superior performance compared to medical practitioners in the realm of breast cancer detection. Additionally, an investigation was conducted involving machine learning models that incorporated imaging biomarkers and predictive algorithms to enable expeditious identification of COVID-19.

1.2 Research Gaps

1. Multidimensional datasets require appropriate analytical methods to provide statistically valid models that can predict the target accurately.
2. The training process of existing algorithms is computationally intensive and time-consuming as a result the learning process becomes more complex; the algorithm needs more time on training.
3. Identifying a person suffering from a particular disease in early stages is a challenging task as the symptoms of the disease surface after some time i.e., gestational period.
4. Bio-informatics tools adapted to respond to new data types are not well equipped to handle the high throughput data received through acquisition technologies.
5. There is no generalized optimization technique for solving all optimization problems, which can provide results in less computational time and handle multidimensional datasets.

1.3 Motivation of Research

The motivation is to build a generalized framework for healthcare applications that researchers and practitioners can use to timely diagnose diseases with increasing prevalence of diseases like Alzheimer's, Parkinson's, Sepsis attack, Type -III diabetes, Covid-19, etc. Many loopholes of the medical industry have surfaced during pandemic Covid-19. As a result, the early and accurate diagnosis of the diseases has not reached culmination, and there is a wide scope for improvisations. As the timely diagnosis of diseases is a cost, effort, and time-consuming process, nature-inspired algorithms and machine learning techniques can prove to be effective in diagnosis of diseases efficiently at a lower cost in lesser time. Subsequently, it is conducive to the field of healthcare (i.e., human health), intelligent diagnosis, and medical education. The Artificial Intelligence (AI) will eternally change the medical industry and diagnose diseases with better accuracy.

1.4 Research Objectives

1. Study and analyze different nature-inspired algorithms and machine learning classifiers implemented for biomedical applications.
2. Investigate various feature extraction and feature selection approaches used at pre-processing stages to optimize nature-inspired algorithms.
3. To design optimized nature-inspired algorithms for dimensionality reduction in the field of biomedical applications.
4. To implement optimized nature-inspired algorithms against various benchmark biomedical datasets and to predict its performance based on number of measuring parameters using machine learning classifiers.

1.5 Research Methodology Overview

In order to start our research and fulfillment of the first objective a comprehensive and systemic review of various nature-inspired algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Binay Bat Algorithm (BBAT), Ant Lion Optimization (ALO), Moth Flame Optimization (MFO), Grey Wolf Optimization

(GWO), Whale Optimization Algorithm (WOA), Ant Colony Optimization (ACO), Spider Monkey Optimization (SMO), Cuttlefish Algorithm (CFA), Crow Search Algorithm(CSA), Chicken Swarm Optimization(CSO) has been performed. Machine learning algorithms used as classifiers for different types of applications has been studied. Nature inspired algorithms based on various feature selection approaches proposed by various researchers and practitioners were analyzed for various biomedical applications in the dimension of disease diagnosis of healthcare.

For the fulfillment of the second objective, various feature selection techniques like wrapper, filter and embedded were implemented to obtain optimal features from the dataset in csv format and feature extraction techniques like Haralick, Gabor, Local Binary Pattern and Hu moments were implemented to extract features from image dataset and further feature selection approaches-based NIAs are used to obtain optimal features in the preprocessing stage for the optimization. For this Grey Wolf Optimization algorithm and Whale Optimization Algorithm were implemented on caridiotocography dataset for optimal feature selection. Features were evaluated using various machine learning classifiers.

For the fulfillment of third objective nature-inspired algorithms BBAT and MFO were enhanced by making them quantum inspired, improved by modifying feature selection steps and various other methods. These elements aids in the optimization of existing Nature-Inspired Algorithms and further propose novel approaches for optimal dimensionality reduction.

For the fulfillment of fourth objective novel designed NIA approaches were implemented against various biomedical datasets like Cardiotocography Dataset, LISC dataset and thyroid dataset and their performance measures like accuracy, F1 score, precision were computed using various machine learning classifiers like Random Forest, Decision Tree, KNN and others.

1.6 Thesis Outline

The thesis is divided into seven chapters, with highlights from each chapter discussed below:

Second Chapter reviews the literature in the field of computational intelligence to bridge the knowledge gap, prominent nature-inspired optimization algorithms will be discussed in terms of their origins, guiding principles and domains of application. Precisely, an examination and investigation were conducted on various nature-inspired algorithms. This review may serve as a reference for selecting suitable algorithms for subsequent research endeavors. Due to the limited availability of substantial literature, we have found a range of nature-inspired algorithms that exhibit considerable potential in both theoretical and practical domains.

Third chapter reveals complications in pregnancies due to various reasons like health issues with mother or conditions that can hamper the development of the fetus which can later affect the health of the baby. CTG performed at the time of high-risk pregnancies can timely identify those associated complications. Fetuses with deficient oxygen amount are more susceptible to fetal distress which can also be fatal. To deal with such problems two nature-inspired algorithms GWO and WOA efficiently select optimal reduced set of features for classification of state of the fetus under normal, suspect and pathologic.

Fourth Chapter introduces the concept of doing differential and qualitative analysis of leukocytes as a means of promptly diagnosing these disorders. In this study, a systematized approach is presented for the classification of leukocytes in blood smear. The suggested model integrates the favorable elements of nature-inspired and quantum-inspired algorithms, resulting in a harmonious combination of both methodologies. The suggested model utilizes the quantum-inspired binary bat algorithm (QBBA) to reduce dimensionality by eliminating extraneous information.

Fifth chapter introduces the Enhanced Binary Bat algorithm, which is a modified version of the Binary Bat Algorithm specifically designed to address the multi-classification problem in Cardiotocography. The prompt evaluation of hypoxic fetuses using cardiotocography holds great importance due to the potential consequences of oxygen deprivation that can lead to fetal distress and the subsequent risk of fatality or neurological disorders.

Sixth chapter proposes an intelligent bio-inspired feature selection algorithm named Improvised Moth Flame Optimization (IMFO) to select the most significant attributes required for early and accurate diagnosis of thyroid disease. The proposed IMFO is modeled as a filter-based feature selection method and is an improvised variant of the recently proposed Moth Flame Optimization algorithm. IMFO successfully selected the optimal subset of thyroid attributes to reduce the computationally expensive learning time. The selected attributes significantly enhanced classification accuracy, improving the overall diagnostic performance.

Seventh Chapter elucidates the key findings of the preceding chapters and explores the prospects for future study that might be conducted to extend the present work. The references have been provided at the end of the thesis.

CHAPTER 2

LITERATURE SURVEY

This chapter presents an overview of recently published research on nature-inspired algorithm and machine learning classification and selection models. It also then presents an overview of research done to assess the impact of nature-inspired model selection methods, model selection criteria, biomedical applications, machine learning classifiers and performance estimation techniques on the selected model.

2.1 Introduction

2.1.1 Computational Intelligence

When treated with various computational intelligence methods, biomedical data helps get a better grip of efficiency in speed and precision detection. In the biomedical field, computational intelligence techniques are categorized under two different heads by using single and hybrid methods. The field of computational intelligence has been subjected to a range of cutting-edge approaches in the context of analyzing medical data [2]. Now a days' applications of science and engineering, likewise decision making, objective optimization, processing of the information etc. use various computational intelligence techniques.

Over the decades, arenas of Genetic Algorithms, Neural Networks, Evolutionary Algorithms are complemented by many newly developed algorithms and methods. In the upcoming years, anomalies and failure detection in different fields like medicine, science & technology and space are fixed and resolved more successfully with the help of intelligent optimization algorithms [5].

Various optimization issues have been tackled by evolutionary computing algorithms under nature-inspired algorithms, which is an area of Computational Intelligence.

The major finding of this chapter has been accepted and published in 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India.

Additionally, the relationship between natural evolutionary techniques' competency and the problem's genre has sparked a lot of interest in evolutionary algorithms.

All the algorithms presented in this study are founded on the underlying assumption that within a particular population of things or individuals, the presence of external pressures leads to the occurrence of natural selection, ultimately leading to an enhancement in the overall fitness of the population. Evolutionary algorithms exhibit variations in their structural implementation, which are contingent upon factors like as population initialization, selection methodology, candidate evaluation, recombination techniques, termination criteria, and mutation strategies.

The fundamental components of evolutionary algorithms include population initialization and termination criteria, fitness evaluation, mutation techniques, selection techniques, and the crossover/recombination process. Typically, these algorithms establish a population in a random manner and undergo evolution for a predetermined number of generations, irrespective of their individual structures. Therefore, the evolutionary process is facilitated by the utilization of reproduction, selection, mutation, and recombination operators.

2.1.2 Metaheuristics

An optimization algorithm is created in computer science to handle intricate problems. The challenge was solved using metaheuristics optimization algorithms that were inspired or modelled by the biological behavior of animals or birds i.e., bioinspired algorithms. A meta-heuristic is a heuristic method for solving complicated optimization issues. A metaheuristics process is a mathematical programming approach for finding an optimal solution to a problem. It utilizes a heuristic method to support the search process. It is possible for the heuristic search to be informed or blind [6].

2.1.2.1 Optimization

Optimization refers to the systematic procedure of identifying the most optimal and viable solution to a given problem, according to particular requirements. In various fields such as business, engineering, and economics, a multitude of intricate optimization problems have emerged. These problems cannot be effectively solved within reasonable timeframes and levels of accuracy using conventional approaches. Nevertheless, the field offers numerous mechanisms and principles that can be leveraged to develop computer intelligence methods for addressing optimization-related challenges[7]. Optimization refers to the process of identifying the optimal decisions of a function with the objective of maximizing or minimizing the function. Numerous engineering obstacles encountered in practical applications are associated with optimization problems [8] wherein decision variables are determined to ensure that systems function at their optimal state.

In recent decades, there has been a notable trend in which numerous optimization algorithms have drawn inspiration from natural intelligence. Nevertheless, the majority of these models incorporate strategies employed by other organisms for foraging, evading threats or preserving their species.

2.1.3 Nature-Inspired Algorithms

Nature-Inspired Algorithms (NIAs) are the algorithms that originate their motivation from nature itself; the considerable expanse of algorithms and solutions are already conceptualized in nature; our task is to look around and adapt them to work out our problems. Nature-inspired computing has brought different fields of biology, mathematics, and computer science under one roof. Optimization algorithms grounded on bio-inspired computing have advanced to novel competing techniques based on nature's biological evolution [9]. Bio-inspired optimization algorithms have been recognized in the field of science and engineering because they are capable of finding optimum solutions to challenging issues in science and engineering. However, the nature of these issues is non-linear, and finding the best solution is restricted by several non-linear constraints like time limitations and high dimensionality. The recent innovations tend to utilize bio-inspired optimization algorithms that symbolize the encouraging technique to resolve the issues provided by classic optimization algorithms in handling difficult optimization issues [10].

2.1.3.1 Nature Inspired Algorithms Categorization

Bio-inspired algorithms (EA), swarm intelligence (SI) and physics/chemistry-based algorithms are the three primary categories of nature-inspired algorithms., as shown in Fig 2.1 [11].

- **Bio-Inspired (BI) Algorithms:** NIAs major segment is made by the Bio-Inspired Algorithms. These algorithms are enthused from natural organisms or by any biological phenomena. Evolutionary Algorithms/Bio-inspired emulate the evolutionary act of creatures set up in nature. The search methods begin by constructing a population using randomly generated solutions, which tend to develop over generations. The next generation is created by mixing the best people, which is an EA's major strength since it stimulates population growth across several iterations. The most common kind of EAs may be well-thought-out as GA and differential evolution (DE) [12] algorithms.
- **Swarm Intelligence (SI) based Algorithms:** The intelligent social behavior of groups of animals is explained using swarm intelligence concepts. Usually, SI-based algorithms fully explore the information about search space and use it to advance the algorithm. However, EAs neglect this type of information from one generation to the other generation. SI category. Any managed group of Interacting Agents or individuals is referred to as a "swarm" like Cuckoo Search Algorithm [13], Grey Wolf Optimization Algorithm [14], Dragonfly Algorithm [15]. Artificial Bee Colony (ABC) Optimization, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Bee system (BS), Ant System (AS), Bee Colony Optimization (BCO), Fireflies Algorithm (FA) are some examples of SI-based algorithms [16].
- **Physics and Chemistry Based Algorithms:** Not all NIAs need to be BI; the algorithms that fall under the category of NIAs but are not bio-inspired are classified under the Physics and Chemical systems category. Examples of such algorithms are Black Hole, Water Cycle Algorithm, Stochastic Diffusion Search, Spiral Optimization, etc. Various physics laws and phenomenon such as electromagnetic force, inertia force, gravitational force, etc. inspires NIAs called as

a physics-based algorithm. Gravitational Search Algorithm (GSA) [17], Big-Bang Big-Crunch (BB-BC), Simulated Annealing (SA) [18], Black Hole, Water Cycle Algorithm, Stochastic Diffusion Search, Spiral Optimization, among others, are all examples of this category.

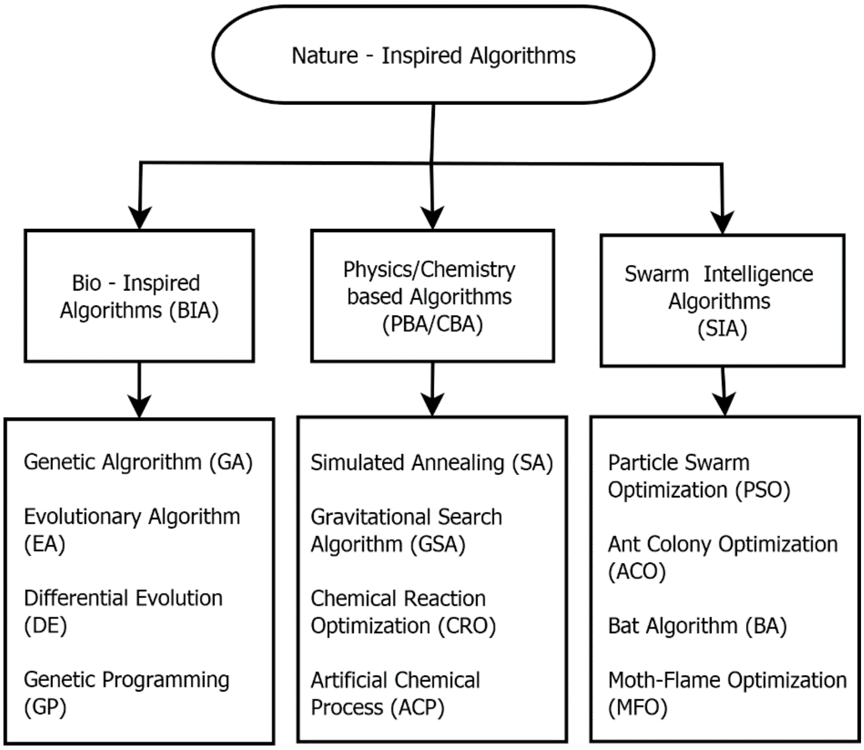


Fig. 2.1: Primary Categorization of Nature-Inspired Algorithms

Lightning Attachment Procedure Optimization (LAPO), Weighted Superposition Attraction (WSA), Spotted Hyena Optimizer (SHO) and many others are examples of modern nature-inspired algorithms. Bioinspired algorithms are subset of Nature-Inspired algorithms and Swarm Intelligence algorithms are a subset of Bio-inspired algorithms, but physics and chemistry-based algorithms are two distinct categories, as shown in Fig. 2.1.

The hierarchy of algorithms i.e., bioinspired algorithms are subset of nature-inspired algorithm and particle swarm optimization algorithm are subset of bioinspired algorithms which is depicted in Fig. 2.2.

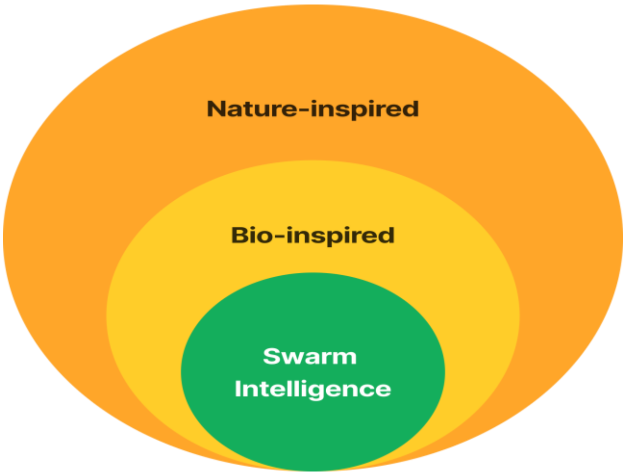


Fig. 2.2: Swarm Intelligence ⊂ Bio-Inspired ⊂ Nature-Inspired

Unfortunately, most of the above-mentioned fundamental metaheuristics algorithms fail to strike balance between exploitation and exploration, yielding disappointing results for real-world complex optimization issues. Exploration refers to the algorithm's global search capabilities, which guarantees that it searches the whole search space and then finds promising regions. At the same time, exploitation denotes the capacity to conduct local searches and enables the most effective search for the identified prospective areas. Increasing exploratory capacity results in a loss of computing resources while exploring within the search spaces' inner regions. As a result, it lowers the convergence rate while highlighting the capacity to utilize only cause's early population diversity losses, resulting in premature convergence or ideal localisation. This fact motivates implementing various strategies to improve the convergence rate and accuracy of standard metaheuristics algorithms [19].

Nature-inspired optimization algorithm mimics biological behavior or physical phenomena. For example, the genetic algorithm (GA) was based on Darwin's evolution hypothesis (the survival of the fittest). GA is one of the well-known population-based optimization strategies. Simulated Annealing (SA) [18] is an optimization technique based on single solution annealing metallurgy phenomenon. Other promising distributed intelligent paradigms for solving complicated optimization problems inspired by the social behavior of birds, colonizing organisms like ants, and honeybees include Particle Swarm Optimization (PSO) [20], Artificial Bee Colony (ABC) [21] and Ant Colony Optimization (ACO) [22]. These optimization approaches have outperformed standard heuristic approaches, notably in multimodal, discreet, and non-differential complicated optimization problems, according to the scientific computational intelligence community. Furthermore, these algorithms have been effectively applied in various scientific domains, like flexible job shop scheduling, biomedical signal processing, process control, image processing, and other engineering design issues [23] [24] [25] [26] [27].

2.2 Reviews on the basis of identified categories

Extensive literature from all of the different fields of nature-inspired algorithms for biomedical applications is collected in order to find the new area of research and find out what the current state of the art is. The literature is mostly research papers, review papers that explain only the ideas, or original contributions with a long list of references. To understand and make sense of the previous work on different aspects related to the formulation of the current research problem, the literature in different sub-areas of nature-inspired algorithms is looked at. This includes mostly nature-inspired algorithms, feature selection methods, model selection criteria, biomedical applications, machine learning classifiers, performance estimation techniques and other specific literature. An attempt is made to review the literature based on the broad goals that have been set and the parts of the literature that are known to be relevant to the proposed work. The identified categories are:

- Nature-Inspired Algorithms
- Feature Selection
- Biomedical Applications

All of the references listed under the different headings cannot be described in detail in this thesis. So, an attempt is made to show the whole body of literature in a way that only a few research articles from each category are used to explain what that category is all about. It helps to draw important conclusions about the direction of research and the potential for more research in that field. The literature isn't complete because not all the journals and books related to the research problem are available, so it's just a sample. However, it supports the development of the nature-inspired algorithm-based methodology for optimal selection of features with good performance measures in terms of accuracy, F1 score, precision, etc., which is done in the thesis. In the next few sections, we'll only talk about the most important and influential articles in different categories that are very important to the problem's core definition.

2.2.1 Nature-Inspired Algorithms

Some researchers have conducted reviews on nature-inspired algorithms [10] [11] , but most of these focuses only on the popular algorithms developed in the domain of nature- inspired algorithms. Fig. 2.2 depicts the detailed taxonomy of the nature-inspired algorithms. The reviews which are conducted on novel approaches are not focused to discuss and explore the scope of applications for these algorithms. Thus, these reviews conducted do not provide an exhaustive exploration of these algorithms across several applications to identify their future scope. Tabu search, simulated annealing, and other popular bio-inspired algorithms are more focused domains in metaheuristics for review purposes. There is no recent review of the novel intelligent approaches developed to explore and integrate different application scopes of these algorithms for ease of researchers to apply. This study hence presents insight into these algorithms for researchers to explore and select a suitable algorithm for application in several domains.

Xin-She Yang proposed an exclusive nature-inspired metaheuristic algorithm namely Bat algorithm [28] which tends to mimic the echolocation technique used by bats to identify an obstacle or a prey. Mirjalili and Xin-She Yang proposed an altered version of existing Bat algorithm for binary problems [29]. Considerable reduction in computational time and cost is desired, Binary Bat Algorithm feature selection approach has outdone various popular swarm-based techniques in numerous public datasets [30]. Modified version of Bat algorithm namely MBAFS [31] was proposed by Bin Yang and others which recorded its comparison with BA and BPSO based algorithms on numerous standard datasets. Similar application was also presented by Gupta et al. [32] [33]. A number of optimization algorithms which are Nature Inspired like Firefly algorithm (FA), Cuckoo Search (CSO), Particle Swarm Optimization

(PSO), Cuttlefish algorithm (CFO) striving local optima was presented by Xin-She Yang in his book [7] .

Thomas Back in his book [34] mentions a number of nature-inspired evolutionary algorithms that find there implementations in a number of areas. J. Kennedy et al. [35] introduced Particle Swarm Optimization (PSO) for Neural Network Optimizations in 1995, since then a number of variations of the PSO have been observed. AK Tripathi et al. [36] proposed a Novel clustering method using enhanced Grey wolf optimizer and map reduce. Bat Algorithm, proposed by Yang et al. [37] is another powerful nature-inspired algorithm which has seen many variations over the years. R. Y. M. Nakamura et al. [30] proposed the Binary Bat Algorithm, which was fit for feature selection. D. Gupta et al. [33] proposed a variation of the Binary Bat Algorithm that fits the classification of White Blood Cells. AK Tripathi et al. [38] proposed dynamic frequency based parallel k-bat algorithm for massive data clustering (DFBPKBA). D. Gupta et al. [33] proposed the application of the Binary Bat Algorithm for the classification of Leukocytes and observed that the Binary Bat Algorithm (BBA) performed better than any other heuristic algorithm for the problem at hand i.e., the classification of leukocytes in blood smear to perform automated differential analysis for medical personnel.

Bio-inspired algorithms came into engineering practices due to their ability to learn from the previous generation to solve more complex problems. In recent years, numerous algorithms have been proposed. Some of the popular algorithms proposed so far are the whale optimization algorithm [14] , the ant-lion optimization algorithm, and the grey-wolf optimization algorithm, in which their hunting mechanism is framed into a mathematical model to solve optimization problems. Other famous algorithms are the cuttlefish algorithm [39] which mimics the color-changing behavior of cuttlefish, the bat algorithm [28], which is inspired by the echolocation mechanisms used by microbats to find food and navigate, the moth-flame optimization algorithm [40], which is inspired by a unique navigation method called transverse orientation and chicken swarm optimization algorithm [41] inferring the hierarchical order in the chicken swarm.

Further, this section presents reviews of each algorithm independently. The algorithms presented in this section including the moth-flame optimization algorithm, cuttlefish algorithm, spider monkey algorithm, binary bat algorithm, island bat algorithm [42], new chaotic whale optimization algorithm [43] either replicate the swarm-based technique or the behavior of biological organisms to attain high-level adeptness even if the optimal solution is not obtained. It is further analyzed that the complexity to attain optimal solution increases non-linearly, as the dimension of these problems increases. To solve such NP-hard problems, swarm-based and related algorithms in the domain of meta- heuristics contribute efficiently. Swarm intelligence focuses more on collective intelligence than individual intelligence that is considered as an adaptive strategy [44]. The algorithms such as spider monkey, cuttlefish, grey wolf, and many more were used to give solutions to these NP-hard problems. Most of these mentioned algorithms provide efficient solutions for multi-objective problems. The intelligent approaches identified solve the multi-objective related problems by converting them to single-objective and further solving them by the prioritization method affecting their computational performance.

The comprehensive exploration of these algorithms highlighted that each algorithm has not been explored and implemented by the scholars which result in the availability of limited literature of these algorithms.

The studies of these explored algorithms highlight the dominant contributors and publication volume until December 2020, as depicted in Table 2.1. The references mentioned in the table consider only those articles where the name of algorithm is present in either title or abstract of the paper and is collected from ScienceDirect, IEEE Xplore, Scopus and Wiley database.

Table 2.1: Dominant contributors and published articles of various NIA algorithms

Algorithms	Dominant contributors	Articles
Genetic Algorithm	Holland; Krishnakumar; Manderick, Spiessens; Rocha; Narayanan A, Moore M; Harik; Falkenauer; Whitley, Rana S; Srinivas, Deb, Agrawal, Pratap	3,10,484

Particle Swarm Optimization	Kennedy, Eberhart; Farrukh, Baig, Masood, Kamran, Naveed; Mounir Ben, Van den berg, Frans, Andries; Parsopoulos, Vrahatis;	1,05,224
Ant Colony Optimization	Dorigo, Colorni; Gambardella, taillard, Dorigo; Thomas, Hoos; Richard, Bullnheimr; Maniezzo; Talbi	30,351
Grey Wolf Optimizer	Mirjalili, Mohammed, Lewis; Kishor, Singh; Malik, Mohideen; Kohli, Arora; Dhargupta, Ghosh, Mirjalili	7,166
Whale Optimization Algorithm	Mirjalili, Lewis; Gaganpreet, Sankalap; Hui Chen, Weide, Xuan	2,094
Ant Lion Optimization	Mirjalili; Zawbaa, Eary, Grosan; Saha; Dinkar, Shail, Kusum	1,013
Spider Monkey Optimization	Bansal, Sharma, Jadon, Clerc; Kumar, Sharma, Rajani; Gupta K., Deep K., Bansal; Siddhartha, Snasel, Dey, Konar	590
Cat Swarm Optimization	Shu-Chuan, Pei-wei, Jeng-Shyang; Tsai, Pei-wei, Jeng-Shyang, Shyi-Ming, Bin-Yih; Orouskhani, Mansouri, Teshnehlab; Kumar, Sahoo; X. Nie, Wang, H. Nie	1,928
Moth-Flame Algorithm	Mirjalili, Savsani, Mohamed A.; P. Jain, A. Saxena; Caiyang, Heidari, Asghar, Huiling; Mohamed, Ahmed A., Ewees, Songfeng	761
Grasshopper Optimization	Mirjalili, Saremi, Lewis; Ahmed A., Mohamed, Essam H., Houssein; Xinxing, Y.Ye, C. Dong; Arora, Anand; Dwivedi, Vardhan, Tripathi	832
Binary Bat algorithm	Nakamura, Pereira, Costa, Rodrigues, Yang; Xingwang, Xuewen Zeng, Rui Han; Varuna, Ramya	2,594
Chicken Swarm Optimization	Xianbing, Yu, Xiozhi, Hengzhen; Wang, Z. Li, Cheng, Zhang; Ahmed, Hassanien, Bhattacharayya; Han, Liu; Qu, Zhao, Yanming, Wei; Ahmed, Ezzat, Tsai	415
Cuttlefish Algorithm	Adel, Adnan, Zeynep; Mohammed, Morad; Mariam, M. A. Badr, Mostafa, Mahmoud	201
Crow Search Algorithm	Alireza; Sayed, Darwish, Ashraf, Aboul; Soheyl, Seyed; Shalini, Singh, Vijendar, Gupta, Rodrigues, Sunram, Khanna, Hugo	1,524

There may be certain significant contribution of these algorithms in other fields, the aforementioned table is determined on the basis of name of algorithm presented in the titleor subject.

However, despite these advances in the domain, researchers have less knowledge regarding the developments and applications across these algorithms. There is hardly any literature which has focused to provide comprehensive understanding of these algorithms along with their scope of application in several areas. Through our review we try to address the gap between the knowledge of novel approaches and the scholars who want to apply them. In the subsequent sub-sections, we present comprehensive theory and developments within these nature-inspired algorithms.

2.2.1.1 Genetic Algorithm

This algorithm [45] focuses on heredity and natural selection for developing search-based algorithms which are sub-division of evolutionary computation. This algorithm provides various solutions. Therefore, recombination and mutations of these solutions lead to new offspring and the process is replicated several times. Further, the fit solutions are obtained by allotment of fitness value to every solution which is evaluated by the value of the objective function. Selection, mutation, and crossover the basic operations performed for the implementation of this algorithm [46]. For use of these operators, a set of pre-optimized solution is chosen for every new possible solution to be obtained. It is notable that the genetic algorithm frequently struggles to resolve complex, multi-modal, and high-dimensional issues when evaluating fitness functions due to the significant range of iterations [47]. In terms of speed and time complexity, the efficiency of genetic algorithms greatly reduces in such problem areas.

2.2.1.2 Particle Swarm Optimization Algorithm

Particle Swarm algorithm [35] focuses on the organisms' behavior in a group to achieve the collective target based on the feedback from other individuals. Every member of the swarm attempts at any time to sense a possible solution. This sends out an indication that the candidate's solution is suitable for other particles present in the swarm. Therefore, any individual member or swarm particle can feel the power and suitability generated by the candidate solution that is obtained using the fitness function of the signal transmitted by other members [48]. If a particle focuses on a better candidate solution obtained from various locally available learning methods, a new direction of movement is established and initially influences the particles to gradually move towards optimal outcome [16] [49].

This algorithm was preliminary developed to solve discontinuous optimization problems but was enhanced with time to handle multi-objective and single-objective continuous optimization problems. Particle Swarm has also been enhanced to introduce various variants including speed coefficient and adaptive inertia weight to enhance precision and optimization speed ofthe algorithm.

2.2.1.3 Ant Colony Optimization Algorithm

Ant Colony algorithm [14] is classified as search-based that solves combinatorial problems of optimization which are swayed by the foraging behavior of the ants. Theprocess includes searching for information by ants through the artificial pathway which depicts indirect contact of the ants. The trails are used as numerical knowledge distributed toagents in order to create probabilistic search-based solutions. The results are acquired [50] in a relatively decent computational time [51]. The solution is achieved by specifying a pool of decision variables after a group of ants is chosen. For candidate solutions, the ants pick the concept variables. While these candidate solutions are explored by the ants, the solution is modified locally, according to its adequacy. Such solutions are utilized to adjust the trail value based on the local update such that solutions with better quality are chosen in successive sampling are used by the same group of ants for obtaining applicant solutions. Thus, the random solutions produced in the primary stage pave the way for an optimum solution.

2.2.1.4 Grey Wolf Optimization Algorithm

Grey Wolf algorithm [52] is a swarm-based optimization algorithm inspired by the technique used for hunting prey by the wolf packs. As per the algorithm, there is division of wolves' pack into four groups based on their dominance in the hierarchy. The search process for this algorithm is similar to other available swarm intelligence-based algorithms. Exploration as well as exploitation of the algorithm is balanced using two parameters to avoid local optimum stagnation. This algorithm is used to address both multi-objective and single-objective optimization problems, while the discrete search variant of this algorithm was proposed to handle the binary optimization problems. Grey wolf algorithm is further enhanced to solve global optimization problems along with developing hybridized approaches with different algorithms. Further chaos, quantum, opposition-based, and parallelism versions of this algorithm were introduced to improve the existing algorithm's performance.

2.2.1.5 Whale Optimization Algorithm

Whale algorithm [53] is also a swarm intelligence-based algorithm that is used to solve continuous optimization problems and is inspired by the behavior of humpback whales. The algorithm mimics the behavior as well as technique used by the whales to search for the location of prey followed by attacking them. The algorithm solves optimization problems while searching prey exploration is performed and during attack behavior exploitation occurs. This algorithm is used with various methods [54] including hybridization with meta-heuristic algorithms, artificial neural networks, support vector machine, and opposition- based learning apart from different variants introduced. Especially notable in this algorithm is the application of Levy flights, which is used to avoid local optimal points and maintain the equilibrium between the algorithm's exploration and exploitation phases. Further, chaos is introduced to improve the algorithm's ability to search and convergence rate.

2.2.1.6 Ant Lion Optimization Algorithm

Ant Lion optimization algorithm [55] is a swarm-based intelligence algorithm that is inspired by the foraging behavior of ant lion's larvae. It imitates the interactivity in bounded surroundings thus the algorithm solves the optimization problems in finite search space. Further, this algorithm [56] replicates five critical stages of larval hunting, including an ant's irrational foray, trap construction, ant entrapment, ant gliding towards ant lions, prey capture, and trap re-construction. Further, the last step of this evolutionary algorithm is elitism which includes preservation of optimal solution and fittest ant lion obtained during this generation which influences all the ants' movements during the iteration. Random walk of ants and adaptive shrinking of traps guarantee exploration and exploitation of search space. This algorithm decreases the movement of ants with iterations which causes improvement of convergence of the algorithm. This algorithm is used with various mechanisms including Levy flights and chaos and in addition to the algorithm under consideration, other algorithms such as differential evolution and particle swarm optimization can be employed to enhance the optimality of the algorithm.

2.2.1.7 Spider Monkey Optimization Algorithm

Spider Monkey algorithm [57] is considered as swarm intelligence-based algorithm that imitates their intelligent foraging behavior based on fission-fusion social structure. The populous directed by a female is divided into small groups for independent foraging. As per the algorithm, proximity to global optimal can be referred to as the fitness of a monkey at a particular position. This algorithm solves optimization problems in the continuous domain which consists, exploration and exploitation. During the search for an optimal solution, the algorithm strives to maintain a balance between the deviation and selection processes. This balance ensures that there is an equal emphasis on both exploration and exploitation phases. Binary variant of this algorithm addresses binary optimization problems. Further to search optimal solution and avoid premature convergence Nelder-Mead method [58] was incorporated with the original spider monkey algorithm.

2.2.1.8 Cat Swarm Optimization Algorithm

Cat Swarm optimization algorithm [59] is swarm intelligence algorithm that imitates the behavior of domestic cats. As per the algorithm, the swarm is divided into seeking and tracing mode. These modes are combined to maintain equilibrium between exploration and exploitation capabilities. Cat swarm algorithm solves all the optimization problems in continuous domain including single-objective and multi-objective. While the cat swarm algorithm is used mostly in the continuous domain, binary variant of this algorithm was presented to tackle binary optimization problems. This algorithm has also been enhanced to introduce various variants including modifying parameters, modifying the structure of the algorithm, altering steps, or hybridizing it with other algorithms. Further chaotic and quantum version [60] of this algorithm was developed to improve accuracy and avoid local optimum for potent global optimization.

2.2.1.9 Moth-Flame Optimization Algorithm

Moth-Flame optimization algorithm [40] is a population-based metaheuristic algorithm that is stimulated from the movement of moths in the night. Combining the local search technique with the population-based algorithm yields local exploitation and global exploration capability. As per the existing algorithm, initially, moths are randomly generated within the solution space followed by determination of fitness value of each moth followed by marking the best position by flame. Subsequently, the position of moths is updated using a spiral movement function to be tagged by flame and repeating the process until the termination criteria. The algorithm also combines with other algorithms to form a hybridized approach along with other introduced variants of the moth-flame algorithm. Notable enhancement is the application of chaos [61] to maintain the stability between exploration and exploitation abilities.

2.2.1.10 Grasshopper Optimization Algorithm

The Grasshopper algorithm [62] is based on the grasshoppers' foraging behavior. This metaheuristic algorithm presented solves the optimization problems in the continuous domain. The algorithm divides search process in exploration as well as exploitation trends. Further, the exploration stage is dealt with a quick movement of grasshoppers while the exploitation phase encourages their local movement. This algorithm is hybridized with various operators like pigeon colony landmark operator and gravity search operator to obtain global optimal solutions and maintain the balance between the phases of the algorithm.

2.2.1.11 Binary Bat Algorithm

Binary Bat algorithm [30] is enhancement of the conventional Bat algorithm that is inspired by the bats’ echolocation behavior. The algorithm integrates characteristics of the bats while searching its prey. It combines position, velocity, and frequency vector for each artificial bat which can be reformed using transfer functions like sigmoid function and formulae defined. It has been applied to deal with graph coloring problem using the sigmoid function to modify both local and global search strategies. This algorithm has also been used with other algorithms to form a hybridized approach along with other introduced variants. Notable enhancement is the application of the binary bat algorithm [63] with cross-entropy to uphold the reasonable stability between exploration and exploitation abilities.

2.2.1.12 Crow Search Algorithm

Crow Search algorithm [64] emulates the intelligent behavior of crows and is even considered a metaheuristic algorithm. The algorithm imitates the behavior of the crows to hide surplus food storage and retrieving it when required. As per the algorithm, crows apply their intelligence to protecting their food and stealing food from others. The further crow search algorithm works on four basic principles including, crows stay in form of the flock, they memorize their hiding place's position, and they stick to each other for thievery and defend their food from being stolen. This population-based algorithm uses adjustable parameters including flight length together with awareness probability which makes its application feasible in distinct engineering areas. This algorithm is used with various mechanisms including the introduction of adaptive inertia weight, spiral search mechanism, and wheel selection scheme to enhance the optimization accuracy. Especially notable in this algorithm is the application of opposition-based learning, which is used to improve exploration virtue. Further, chaos is introduced to enhance its convergence rate along with solving constrained optimization problems.

2.2.1.13 Chicken Swarm Optimization Algorithm

Chicken Swarm algorithm [65] is a stochastic-based optimization algorithm that is inspired from the social behavior of chickens. The algorithm considers the hierarchy order of chicken in the swarm and imitates their behavior. As per the algorithm, the chicken swarm is split up into several clusters, which include a rooster, some hens, and some other roosters. The behavior of each chicken is idealized using certain rules and their motion laws specified. In each hierarchy order, competition among different subgroups indicates this algorithm to be a global optimization algorithm. Chicken swarm algorithm solves multi-objective optimization problems. The binary variant of this algorithm is presented to address binary optimization problems. This algorithm has also been enhanced to introduce various variants including feature selection to reduce input parameters and even applying it to find a global optimal solution for low dimensional and high dimensional problems. Further chaos version of this algorithm was developed to improve search ability and avoid local optimum.

2.2.1.14 Cuttlefish Algorithm

Cuttlefish Algorithm [39] is a metaheuristic optimization algorithm stimulated by the behavior of the color-changing cuttlefish. The algorithm depends on two essential processes, including reflection, and visibility that are used as a search strategy to detect a global optimal solution. Light reversal mechanism is simulated using the reflection process, while the clarity of matching patterns is imitated using the visibility process. The algorithm splits the population into four groups which further carries two global together with two local searches to obtain the global optimal. A discrete variant of this algorithm was proposed to address the optimization problem in the discrete domain. Further to detect the optimal solution of the traveling salesman problem, feature selection has been incorporated with the existing algorithm.

Table 2.2 depicts main contributions of each nature-inspired algorithm along with their short description.

Table 2.2: Main Contributions of various NIA algorithms and their variants

Algorithm	Variant	Author	Short description
Genetic Algorithm (GA)	GA	Holland [45]	
	Micro GA	Krishnakumar [66]	Evolved small populations efficientin locating areas
	Cellular GA	Manderick et al. [67]	Based on theory and concepts of GAand structured population
	Contextual GA	Rocha [68]	Developed on basis of biological system of RNA
	Quantum-inspired GA	Narayanan et al. [69]	Employed concepts and principles ofquantum mechanics
	Linkage learning GA	Harik [70]	Integrated genetic linkage learning toalgorithm
	Grouping GA	Falkenauer [71]	Utilized for solving clustering problems
	Island GA	Whitley et al. [72]	Incorporated multiple subpopulationsto prevent genetic diversity
	Nondominated sorting GA	Srinivas et al. [73]	Employed to handle multi-objectiveoptimization problems
	Non-dominated sorting GAII	Deb et al. [74]	Extension of NSGA for fast non-dominated sorting
	Non-dominated sorting GAIII	Deb et al. [75]	Extension of NSGA II for referencepoint approach
	Jumping gene GA	Man et al. [76]	Incorporated concept of biologicalmobile genes
	Hierarchical cellular GA	Janson et al. [77]	Augmented the population structurewith hierarchy
	Dynamic rule-based GA	He et al. [78]	Integrated heuristic rules to GA
	Tribe competition-based GA	Ma et al. [79]	Divided population of tribes in groups
	Interactive GA	Takagi [80]	Employed the value of fitness function in algorithm
Particle Swarm Optimization	PSO-AVP	Mounir BenGhalia [81]	Based on Active Velocity Penalty forconfinement of particles within search space
	Cooperative PSO	Van den Berg etal. [82]	Employing cooperative behavior of multiple swarms
	Adaptive PSO	Zhi-Hui Zhan etal. [83]	Develop a systematic parameter adaptation scheme and an elitist learning strategy
	Constrained optimization-based PSO	Parsopoulos etal. [84]	Incorporated non-stationary multi-stage assignment penalty function

Algorithm	Variant	Author	Short description
	Stretched PSO	Parsopoulos etal. [85]	Implement Stretching function to alleviate the local minima problem
	Stretched PSO II	Parsopoulos et al. [86]	Extension of SPSO for global minima
	Nbest PSO	Brits et al. [87]	Developed Nbest technique by adapting standard Gbest technique
	Guaranteed Convergence PSO	Van den bergh [88]	Introduced new parameter to Standard PSO
	Niche PSO	Brits et al. [89]	Utilizing sub-swarms along with GCPSO and congnition-only PSO
	Niche PSO II	Engelbrecht [90]	Extension of Niche PSO I by changing merging and absorption methods
Ant Colony Optimization	ACO	M. Dorigo et al. [50]	
	Elitist AS	Gambardella etal. [91]	Incorporated local search with standard ACO
	Ant-Q	Gambardella etal. [92]	Extension of Ant system with incorporation of reinforcement learning
	Max-Min Ant System	Thomas et al. [93]	Modifying update mechanism, avoidstagnation by initializing trails
	Rank-based AS	Richard et al. [94]	Extension of Elitist AS by considering ants based on ranking
	ANTS	Maniezzo V. [95]	Incorporated elements from branchand bound techniques into ACO algorithm
	Best-worst AS	Cordon O et al. [96]	Implemented the concept to eliminatethe worst ant
	Population based ACO	Guntch et al. [97]	Employed the population management mechanism to set of elite solutions
	Beam-ACO	Blum et al. [98]	Implemented heuristic approach ofbeam search to standard ACO
	Hyper Cube ACO	Van den bergh [99]	Introduced new parameter to Standard PSO
	Cunning AS	Shigeyoshi [100]	Utilizing part of existing solutions to generate betterresults
	Quantum ACO	Ling et al. [101]	Incorporating concepts of quantum in standard ACO
	Hybrid ACO	Ding et al. [102]	Introduced disaster operator and adjusted pheromone approach

Algorithm	Variant	Author	Short description
	ACS-SPSP	Xiao J. et al. [103]	Incorporating method to handle software project scheduling problem
	Parallel ACO	E. G. Talbi etal. [104]	Implementation of parallelmodel to standard ACO
	Modified ACO	Chiabwoot [105]	Application of mutation concept to ACO for avoiding local optimum trap
Grey Wolf Optimization	GWO	Mirjalili et al. [14]	
	Modified GWO	Kishor et al. [106]	Incorporated crossover operator to standard GWO
	Modified GWO II	Chandra et al. [107]	Extension of MGWO I with population-based technique
	GWO-EPD	Saremi et al. [108]	Introduced evolutionary population dynamics in GWO
	Complex-valued encoding GWO	Luo et al. [109]	Employed complex-valued encoding method instead of simple encoding
	MAL-IGWO	Long et al. [110]	Incorporated IGWO and Modified Augmented Lagrangian
	Weighted distance GWO	Malik et al. [111]	Enhanced location updates using weighted sum strategy
	Lévy-embedded GWO	Heidari et al. [112]	Integrated Lévy flight and greedy selection strategies along with modified hunting phases
	Chaotic GWO	Kohli et al. [113]	Introduced chaotic theory and concepts to GWO
	COGWO2D	Ibrahim et al. [114]	Employed chaotic logistic map, opposition-based learning, differential evolution and disruption operator to GWO
	Selective opposition-based GWO	Dhargupta et al. [115]	Integrated opposition-based learning to standard GWO
Whale Optimization Algorithm	WOA	Mirjalili et al. [53]	
	Chaotic WOA	Gaganpreet etal. [116]	Employing various chaotic maps to WOA
	Lévy flight trajectory-based WOA	Ying Ling et al. [117]	Based on Levy flight trajectoryto enhance population diversity
	Opposition-Based WOA	Hammoudeh etal. [118]	Employing opposition-based learning to WOA
	Adaptive WOA	Indrajit et al. [119]	Based on less parameter dependency
	Chaos mechanism based on quasi-opposition WOA	Hui Chen et al. [120]	Employing chaotic maps and quasi-opposition learning to WOA

Algorithm	Variant	Author	Short description
Ant Lion Optimization	ALO	Mirjalili [55]	
	Chaotic ALO	Zawbaa et al. [121]	Introduced chaotic maps to Standard ALO
	Enhanced ALO	Subhashini etal. [122]	Introduced novel optimization technique to Standard ALO
	Quasi-Oppositional Chaotic ALO	Saha et al. [123]	Employing quasi-opposition based learning
	Adaptive ALO	Naveen Sihag [124]	Developed and integrated adaptive technique
	Lévy ALO	Emary et al. [125]	Incorporated Lévy flight random walk
	Opposition-based Lévy flight ALO	Dinkar et al. [126]	Implemented Lévy flight random walk in conjunction with Opposition-based learning
	Nelder-Mead ALO	Chengbin et al. [127]	Introducing Nelder-Mead algorithm to Standard ALO
Spider Monkey Optimization	SMO	Bansal et al. [57]	
	Self-Adaptive SMO	Kumar et al. [128]	Using self-adaptive strategy to update position
	Modified Position Update SMO	Kumar et al. [129]	Employing Golden Section Search process to StandardSMO
	Adaptive Step-size SMO	Hazrati et al. [130]	Utilizing fitness to asses step-size
	Constrained SMO	Gupta et al. [131]	Deb's constraint handling technique
	Tournament Selection-based SMO	Kavita et al. [132]	Enhanced probability scheme using Tournament Selection method
	Nelder-Mead SMO	Prabhat et al. [133]	Using transformations ofNelder-Mead method to Standard SMO
	Quantum-inspired SMO	Siddhartha etal. [134]	Concepts and principles of quantum have been applied tothe algorithm
	Ageist SMO	Sharma et al. [135]	Introduced age and changing features to Standard SMO
	Quantum-inspired SMO and ASMO	Alokananda et al. [136]	Introduced concepts of quantum to standard and Ageist variant of SMO based on automatic clustering techniques
Cat Swarm Optimization	CSO	Shu-Chuan etal. [59]	
	Parallel CSO	Tsai et al. [137]	Concept of parallel structure on Standard CSO

Algorithm	Variant	Author	Short description
	Enhanced PCSO	Tsai et al. [138]	Utilizing the orthogonal array of Taguchi method to the algorithm
	Average-Inertia Weighted CSO	Orouskhani etal. [139]	Introduced inertia value to the algorithm
	Adaptive Dynamic CSO	Orouskhani etal. [140]	Enhanced AICSO by employing adaptive techniques
	Opposition-Based Learning-Improved CSO	Kumar et al. [141]	Introduced Cauchy mutation operator to the algorithm
	Opposition-Based Learning-Improved CSOII	Kumar et al. [142]	Extension of OL-ICSO by utilizing opposition-based learning and heuristic mechanisms
	Multi-Objective CSO	Pradhan et al. [143]	Concepts of external archiveand Pareto dominance are incorporated
	Chaos Quantum based CSO	Nie et al. [144]	Introducing tent map techniqueto QCSO
	Normal Mutation Strategy-Based CSO	Pappula et al. [145]	Employing normal mutation technique to Standard CSO
	Compact CSO	Zhao [146]	Employing differential and probability model to Standard CSO
Moth Flame Optimization	MFO	Mirjalili [40]	
	Non-Dominated MFO	Vimal et al. [147]	Utilizing elitist non-dominated sorting and crowding distance approach
	Enhanced MFO	Asmaa A. et al. [148]	Introducing Lévy flight to the algorithm
	Enhanced MFO	Xu et al. [149]	Employing mutation strategy to the algorithm for global optimization
	Opposition-Based MFO	Pooja et al. [150]	Incorporating opposition theory to Standard MFO
	CLSGMFO	Yueting Xu etal. [151]	Implementing Gaussian mutation and Chaotic LocalSearch to algorithm
	Quantum-Behaved Simulated-AnnealingMFO	Caiyang Yu etal. [152]	Implementing simulated annealing strategy and quantum rotation gate to algorithm
	OMFODE	Mohamed et al. [153]	Implementing opposition-based learning and differential evolution approach to algorithm
Grasshopper Optimization Algorithm	GOA	Mirjalili et al. [62]	
	Opposition-based learning GOA	Ahmed A. et al. [154]	Based on opposition-based learning applied to GOA

Algorithm	Variant	Author	Short description
	Gaussian Chaos GOA	Xingxing et al. [155]	Employed Gaussian and logistic chaotic map to enhance GOA
	Chaotic GOA	Arora et al. [156]	Introduced chaotic concepts and theory to GOA
	ECAGOA	Dwivedi et al. [157]	Integrated Ensemble of Feature Selection and Chaotic Adaptive GOA
Binary Bat Algorithm	BBA	Nakamura et al. [30]	
	Cross-entropy BBA	Guocheng et al. [158]	Embedded cross-entropymethod to BBA
	Improved BBA	Xingwang et al. [159]	Employed inertia concept with neighborhood search to enhance BBA
	BBA and Naïve Bayes	Varuna et al. [160]	Integrated Naïve Bayes andBBA for intrusion detection
Crow Search Algorithm	CSA	Alireza Askarzadeh [64]	
	Chaotic CSA	Sayed G. et al. [161]	Incorporated chaotic concepts to the algorithm
	Sine Cosine CSA	Soheyl et al. [162]	Implementing Sine Cosine Algorithm to Standard CSA
	Improved CSA	Jain et al. [163]	Introducing experience factor, adaptive adjustment operator and Levy flight distribution
	Modified CSA	Gupta et al. [164]	Utilizing for usable feature extraction from hierarchical model
	Intelligent CSA	Shalini et al. [165]	Incorporated cosine function and opposition-based learning to algorithm
Chicken Swarm Optimization	CSO	Xianbing et al. [65]	
	Non-linear inertia weight MCSO	Wang et al. [166]	Introduction of non-linear decreasing in the update mechanism of rooster
	Chaotic CSO	Ahmed et al. [167]	Incorporating tent map and logistic map with swarm intelligence to algorithm
	Binary Improved CSO	Han and Liu [168]	Application of mutation operator is applied to populationwith worst fitness value
	Elite Opposition-based learning CSO	Qu et al. [169]	Incorporated elite opposition-based learning to algorithm
	Adaptive CSO	Ahmed et al. [170]	Conversion of basic CSO to discrete swarm algorithm by

Algorithm	Variant	Author	Short description
			encoding and decoding
Cuttlefish Algorithm	CFA	Adel et al. [39]	
	CFA-Clustering	Adel Sabry etal. [171]	Employed CFA to enhance the performance of clustering problems
	Discrete CFA	Mohammed etal. [172]	Developed discrete CFA tosolve travelling salesman probems
	PSO & CFA	Mariam et al. [173]	Developed optimization control schemes for PV system using hybrid of PSO and CFA

The potential of these algorithms has been summarized using their application scope and objectives that can be contented based on these algorithms' existing literature. This has been exemplified through Table 2.3.

Table 2.3: Description of problem domains and potential solution objectives of various NIA algorithms

S. No.	Algorithms	Description of Problem domain	Potential solution objectives
1	Genetic Algorithm	Multi-objective, Single-objective problems; static solution space having low scalability	Search, maximization or minimization, traveling salesman problem, minimum vertex cover problem, prioritization, sorting, selection, job scheduling, parallel computation, network path routing, image processing, data mining, computer games, load balancing problems, dispatch problem
2	Particle swarm optimization	Multi- objective, single-objective problems (continuous and discontinuous domain), high scalability	Feature selection problems, clustering problems, constrained optimization problems, numerical optimization problems, job-schedulingproblems, regulation, classification, resource allocation, chaotic systems, global optimization, path optimization, traveling salesman problems, oscillatory sytems, optimal power flow, Bayesian networks, power dispatch problems, photovoltaic power generation, designing PID controllers, power systems adaptive learning, network training, minimization, maximization.

S. No.	Algorithms	Description of Problem domain	Potential solution objectives
3	Ant Colony Optimization	Single variable and multi-variable problems, discreteand continuous optimization problems, NPhard-based problems	Feature selection problems, clustering problems, constrained optimization-based problems, numerical optimization problems, job-scheduling problems, environmental and economic dispatch problems, classification problems, parameter estimation, traveling salesman problem, data compression, routing, demand forecasting, gaming theory, objective tracking, layout design, optimal power flow, routing, timing optimization, resource consumption optimization.
4	Grey Wolf Optimization	Population-based multi-objective and single-objective optimization problems	Feature selection problems, global optimization problems, discrete search-based problems, clustering problems, knapsack problems, chaotic problem, power dispatch problems, design and tuning controllers, scheduling problem, traveling salesman problem, image processing, wireless sensor networks, environment modeling applications, medical applications, global and constrained optimization problem.
5	Whale Optimization Algorithm	Continuous and discrete search-based problems and multi-objective optimization problems	Feature selection problems, problems including inertia weight parameter, knapsack problems, clustering problems, task scheduling problems, chaos problem, levy-based problem, opposition-based problem, electrical engineering, mechanical engineering, image processing, control engineering, industrial engineering, signal processing and robot path networks.
6	Ant Lion Optimization	Single, multi-objective optimization problems (continuous domain)	Constrained optimization problems, global optimization problems, discrete search problems, binary optimization problems, chaos problem, Levy fly problem, optimal power flow, economic dispatch problems, training neural networks, feature selection, structural design problems, image segmentation, power systems and designing PID controllers.
7	Spider Monkey Optimization	Single-objective optimization problems	Search, maximization / minimization, constrained optimization problems, self-adaptive problem, binary problem, hydro-thermal coordination, design of wireless telecommunications networks, data mining and clustering.

S. No.	Algorithms	Description of Problem domain	Potential solution objectives
8	Cat Swarm Optimization	Single-objective, multi-objective optimization problems (continuous domain)	Discrete search-based problems, clustering, knapsack problems, numerical optimization, combinatorial optimization problems, constrained and unconstrained optimization, global optimization, adaptive, chaos and quantum problem, system management and combinatorial optimization, signal processing, computer vision, wireless and WSN.
9	Moth-Flame Optimization	Multi-objective optimization problem	Feature selection problems, global optimization problems, scheduling problems, parameter estimation and binary search-based problems, chaos problem, economical applications, image processing, networks, medical domain, PID control, power dispatch, power flow, power energy, electrical engineering and wind energy.
10	Grasshopper Optimization	Multi-objective optimizationproblem (continuous and discrete domain)	Data clustering problems, feature selection problems, binary optimization problems, opposition-based problem, employing non-linear convergence parameter, optimal power flow, image segmentation and financial stress prediction.
11	Binary Bat algorithm	Binary based optimization problems	Feature selection problems, optimization problems in binary space, set covering problems, graph coloring problems, knapsack problems, power system, intrusion detection, image enhancement and medical applications.
12	Crow Search Algorithm	Single-objective, multi-objective optimization problems	Parameter estimation, global optimization problems, discrete search problems, data clustering problems, numerical optimization problems, scheduling problems, chaos problem, opposition-based problem, optimal power flow, medical applications, economic environmental dispatch problems, vehicle routing problems, electromagnetic optimization, economic load dispatch problems, image segmentation, agricultural applications and optimal control problems.
13	Chicken Swarm Optimization	Single-objective, multi-objective optimization problems, high and low dimensional problems.	Binary search-based problems, feature selection problems, job-scheduling problems, knapsack problems, global optimization, chaos.
14	Cuttlefish Algorithm	Optimization problems.	Feature selection problems, clustering problems, traveling salesman problems, design of PIA controllers, intrusion detection using feature selection, image segmentation and medical applications.

Based on scope and exploration regarding the development and application of these algorithms in different surroundings as mentioned in the literature, a direction for further exploration of each of these algorithms has been mentioned in the successive section.

2.2.2 Feature Selection

Feature set reduction is a process that involves evaluating the relevance and redundancy of features in relation to a specific output. In a more precise manner, a feature is typically classified into four categories: 1) strongly relevant, 2) weakly relevant yet non-redundant, 3) irrelevant, and 4) redundant. An optimal feature subset always requires the inclusion of a highly relevant feature, as its removal will inevitably impact the original conditional target distribution.

The primary objective of feature selection is to choose a subset of variables from the input that effectively captures the characteristics of the input data, while minimizing the impact of noise or irrelevant variables and yielding accurate prediction outcomes.

The procedure of identifying the feature subset generally involves four fundamental steps: 1) formation of the subset 2) evaluation of the subset 3) establishment of a stopping condition and 4) validation of the obtained results. There are various ways to classify feature selection methods. One prevalent approach involves categorizing approaches into filters, wrappers, embedded and hybrid techniques.

The variable elimination approaches were categorized into two main categories in: filter methods and wrapper methods. Filter methods are utilized as a preprocessing step to prioritize the features, with the intention of selecting and applying the most highly ranked characteristics to a predictor. Wrapper approaches include selecting features based on the performance of a predictor. This is achieved by wrapping the predictor with a search algorithm that identifies the subset of features that yields the maximum predictor performance. Embedded techniques incorporate variable selection inside the training process, eliminating the need for data separation into training and testing sets. Numerous filter methods have been documented in scholarly literature. Table 2.4, presents a compilation of prevalent approaches, along with corresponding sources that offer comprehensive information

Table 2.5 displays various search strategies that are commonly used.

Table 2.4: Common filter methods for feature selection [174]

Name	Filter class	Applicable to Task
Information gain [175]	univariate, information	Classification
Gain ratio [176]	univariate, information	Classification
Symmetrical Uncertainty [177]	univariate, information	Classification
Correlation [177]	univariate, statistical	Regression
Chi-square [176]	univariate, statistical	Classification
Inconsistency criterion [178]	multivariate, consistency	Classification
Minimum redundancy, maximum relevance (mRmR) [179]	multivariate, information	Classification, regression
Correlation-based feature selection [176]	multivariate, statistical	Classification, regression
Fast correlation-based filter (FCBF) [177]	multivariate, information	Classification
Fisher score [180]	univariate, statistical	Classification
Relief and ReliefF [181]	univariate, distance	classification, regression
Spectral feature selection (SPEC) and Laplacian Score (LS) [182]	univariate, similarity	classification, clustering
Feature selection for sparse clustering [183]	multivariate, similarity	Clustering
Localized Feature Selection Based on Scatter Separability (LFSBSS) [184]	multivariate, statistical	Clustering
Multi- Cluster Feature Selection (MCFS) [182]	multivariate, similarity	Clustering
Feature weighting K-Means [185]	multivariate, statistical	Clustering
ReliefC [186]	univariate, distance	Clustering

Table 2.5: Search strategies for feature selection [174]

Algorithm Group	Algorithm Name
Exponential	Exhaustive Search Branch and bound
Sequential	Greedy Forward selection or Backward elimination Best First Linear Forward Selection Floating forward or backward selection Beam Search (and beam stack search) Race Search
Randomized	Random Generation Simulated Annealing Evolutionary Computation Algorithms Scatter Search

The selection of feature selection methods varies across different domains of application. In the subsequent subsections, we examine comparative research on feature selection as it relates to a variety of widely recognized application domains. In Table 2.6, a summary of the findings obtained from the studies that were reviewed is presented.

Table 2.6: Summarized findings of relevant feature selection methods in various application area [174]

Application area	Subfield	Datasets	Feature selection methods	Evaluation Metrics	Best performing	Study
Text mining	Text Classification	229 text classification problem instances gathered from Reuters, TREC, OHSUMED, etc.	Accuracy, accuracy balanced, bi-normal separation, chi-square, document frequency, F1-measure, information gain, odds ratio, odds ration numerator, power, probability ratio, random	Accuracy, F-measure, precisionand recall	Information gain (precision), bi-normal separation (accuracy, F-measure, recall)	[187]
	Text clustering	Reuters-21578, 20Newsgroups, Web Directory	Information gain, chi-square, document frequency, term strength, entropy-based ranking, term contribution, iterative feature selection	Entropy, precision	Iterative feature selection	[188]
Image processing / computer vision	Image classification	Aerial Images, The Digits Data, Cats and Dogs	Relief (R), K-means (K), sequential floating forward selection (F), sequential floating backward selection (B), various combinations R + K + F/B	Average MSE of 100 neural networks	R+K+B / R+K+F R+K, depending on the size of feature subset	[189]
	Breast density classification from mammograph-ic Images	Mini-MIAS, KBD-FER	Best-first with forward, backward and bi-directional search, genetic search and random search (KNN and Naïve Bayesian classifiers)	Accuracy	Best first forward, best first backward	[190]
Bioinformatics	Biomarker discovery	Three benchmark datasets deriving from DNA microarray experiments	Chi-square, information gain, symmetrical uncertainty, gain ratio, OneR, ReliefF, SVM-embedded	Stability, AUC	Chi-square, symmetrical uncertainty, information gain, ReliefF	[191]
	Microarray gene expressiondata classification	Two gene expression datasets (Freije,	Information gain, twoing rule, sum minority, max minority, Gini index,	Accuracy	Consensus of all methods	[192]

Application area	Subfield	Datasets	Feature selection methods	Evaluation Metrics	Best performing	Study
		Phillips)	sum ofvariances, t-statistics, one-dimensional SVM			
Industrial applications	Fault diagnosis	Wind turbine test rigdataset	Distance, entropy, SVM wrapper, neural network wrapper, global geometric similarity scheme	Accuracy	Global geometric similarity scheme with wrapper	[193]

2.2.3 Biomedical Applications

Subrota Mazumdar et.al. proposed a technique that monitors fetal heart rate using Artificial Neural Network [194]. Vinayaka Nagendra et.al. discussed the significance of machine learning for backing obstetricians in pathologic suspect cases [195], Huang and Hsu [196] evaluated the fetal distress on CTG data using various techniques of ANN, discriminant analysis(DA) and decision tree with the accuracies of 97.78%, 82.1%, 86.36% respectively. Anish Batra et.al. evaluated the fetal distress using combination of various machine learning algorithms [197]. Sundar et al. [198] proposed classification of CTG data using supervised ANN.

Afaf Tareef et al. [199] put forth a technique for fully automated image segmentation of leukocytes based on the colour and textures of the microscopic cell images. Der-Chen Huanga et al. [200] proposes segmentation by enhancing the nucleus of the cell and suppressing the other regions like blood smear. A variety of other techniques such as Geometric Active Contours and Level Set Method have also been implemented by Khamael ALDulaimi et al. [201], which can easily segment nuclei from cell walls and cytoplasm.

The work done in [202] clearly explains the risk of cancer due to radiation. According to the study presented in [203], x-ray exposure leads to neoplastic disease, due to which the cells of the body show abnormal growth, which may gradually lead to cancer.

To overcome the limitations above and side effects of the primary detection methods, various artificial intelligence models for diagnosing thyroid disease have been proposed and implemented in the literature. These models require medical data of patients and ordinary people, confined in a tabular manner, covering maximum factors for which disease may occur. In work done by [204], various machine learning classifiers were implemented on the thyroid dataset. They achieved the maximum accuracy of 97.23% with the decision tree classifier, followed by 96.04% using the support vector machine, and obtained the least accuracy of 6.31% by Naive Bayes. In [205], the authors implemented an auto-associative Neural Network and obtained a maximum accuracy of 95.1%. Recently, an optimal feature-based multi-kernel support vector machine framework was proposed in [206] and outperformed all the approaches proposed with an accuracy of 97.49%. In the presented work, various machine learning classifiers have been implemented along with a bio-inspired algorithm that enhances the performance of classifiers in terms of both time and accuracy.

2.3 Inference of Critical Review of the Algorithms

The analysis of the scope and objective of each algorithm in section 2.2.1 concluded that the literature and exploration of these algorithms are extremely skewed. There are some algorithms which require a literature to provide deep insight. Further, these algorithms have been classified into four classes where each of them has a different scope. The classifications of these algorithms into quadrants have been depicted in Table 2.7. The inference of each algorithm to be in a specific quadrant has been elucidated subsequently.

Table 2.7: Categorization of NIA algorithms based on their scope

Quadrant 1: Theory Development Zone • Binary Bat Algorithm • Chicken Swarm Optimization • Cuttlefish • Grasshopper	Quadrant 2: Applications Zone • Spider monkey optimization • Cat swarm optimization • Moth-Flame optimization • Crow search
Quadrant 3: Rediscovery Zone • Whale optimization • Ant Lion optimization	Quadrant 4: Commercialization Zone • Grey Wolf Optimization algorithm • Ant Colony Optimization • Particle Swarm Optimization • Genetic Algorithm

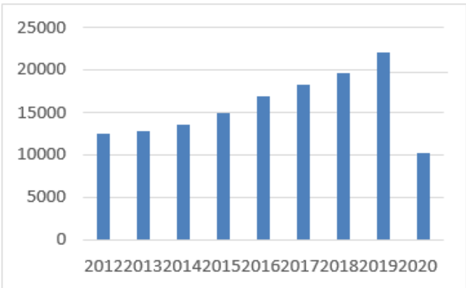
All the algorithms have been grouped in the above-mentioned categorization obtained on the basis of our literature review and the way they can be further studied. Quadrant 1 represents the field of evolution of theory, where there is a great deal of variety both in terms of incremental algorithm growth and comparative analytics among algorithms. The results of such research should be added to the existing literature in the smart systems and metaheuristic systems. Besides, the absence of literature in these algorithms highlights the immense potential of researchers working in this area to explore how novel enhancements and measures including chaos, uncertainty, or constraints can improve these algorithms.

Quadrant 2 covers the application domain where the algorithms are mature in terms of the advancement of the theory. Revisiting these algorithms and further applying them in various fields would be interesting. There could be various areas that identify these algorithms applicable for Quadrant 2 which perform implementation in business and engineering-related fields. These researches would greatly benefit from the literature around the use of these expert systems, metaheuristics and intelligent systems and would thus attract the attention of journals publishing research in their particular area. It will be of interest to the scientific community to apply these hybridized approaches for these algorithms.

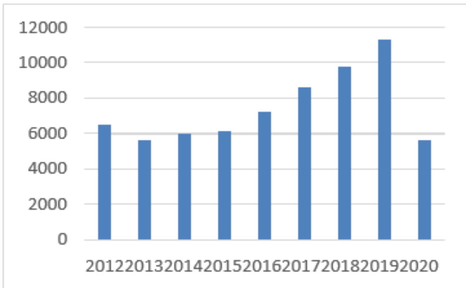
Quadrant 3 covers the algorithms implemented, but it did not catch the research community's attention. Although these mentioned algorithms have literature in terms of both theoretical growth as well as implementation, there was comparatively less interest in the application of these algorithms in various studies. The academic communities have perhaps faced the difficulties of applying algorithms in different fields. Perhaps it is interesting to try to combine these algorithms with other hypotheses and to investigate the consistency of the outcome. There could be better results for combining these algorithms with these such as rough sets, fuzzy sets, chaotic systems, high-value systems, further documenting their theoretical foundations. It would be useful for the academic research-based fraternity to explore the application of such algorithms in novel problem domains.

Quadrant 4 identifies algorithms that are specifically implemented by a broad community of scientists that have been implemented for various applications. It can be a daunting job to find unexplored applications that can concern scientists. Furthermore, several studies have studied the development of hypotheses in these algorithms and the empirical ramifications for the scientific community will be less important until significant new developments occur in the creation of new methods within those algorithms. Therefore, these algorithms are ready to be applied in practice considering that the research community has extensively used these algorithms. The algorithms may be easily implemented for industrial applications because over the scientific exploration era much of their pseudo-codes and algorithmic logic have beendeveloped as well as been standardized.

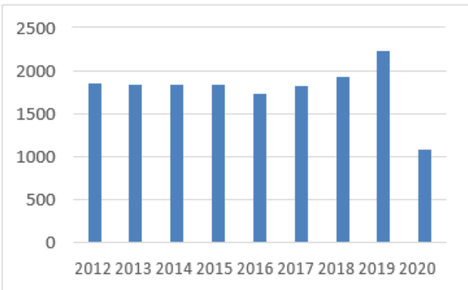
Apart from the above discussed future scope of these algorithms, Fig. 2.3: No of published articles of various NIA algorithms in ScienceDirect, IEEEXplore, Scopus, Wiley Fig. 2.3 depicts the growth of all the algorithms through analyzing their number of publications in different databases including ScienceDirect, IEEEXplore, Scopus and Wiley.



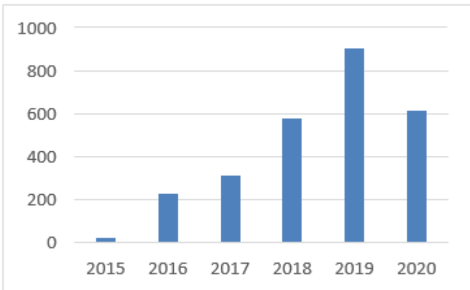
(a) GA



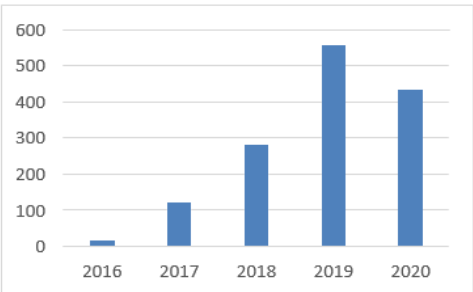
(b) PSO



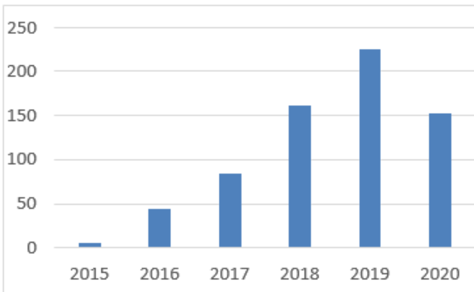
(c) ACO



(d) GWO



(e) WOA



(f) ALO

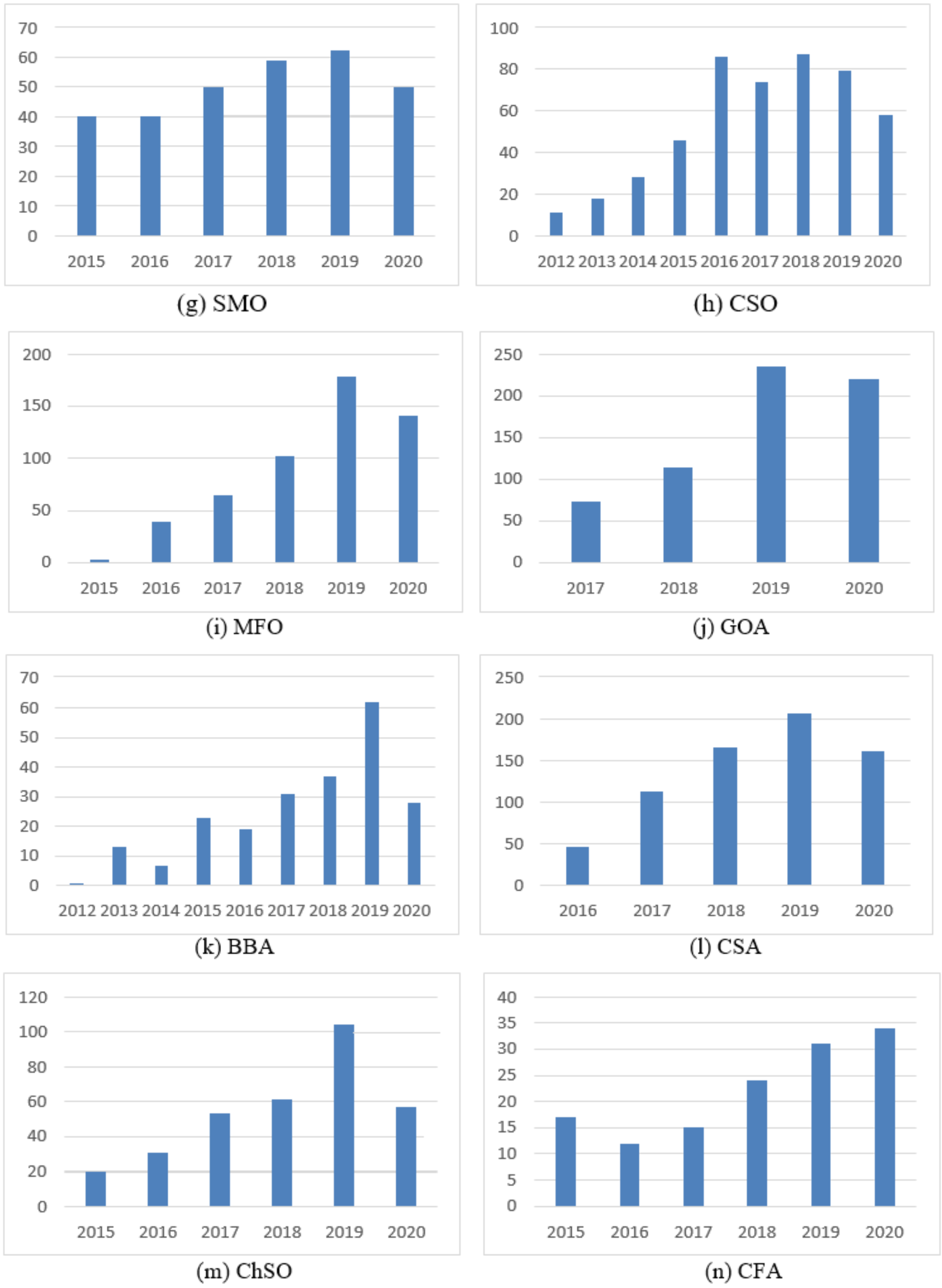


Fig. 2.3: No of published articles of various NIA algorithms in ScienceDirect, IEEEXplore, Scopus, Wiley

2.4 Conclusion

Some of the available nature-inspired algorithms including neural networks, particle swarm, genetic, ant colony, and grey wolf have a wide range of perspectives on stability and convergence of these reviewed algorithms. The purpose of this research work is to provide a primary insight into the nature and purpose of the novel algorithms developed instead of figuring the method to address the real-life optimization problems.

Some researchers have conducted reviews on metaheuristic algorithms, but most of these focuses only on the popular algorithms developed in the domain of nature-inspired algorithms. There have been limited reviews on swarm intelligence-based and other nature-based algorithms. Although some modern algorithms have been reviewed individually, they do not provide the reader with an overview of their ability to use certain similar algorithms when the problems are identical. Such reviews can only be useful if the scientist is familiar with various forms of such algorithms. This analysis also guides the readers to review the research work related to such algorithms that can help gain an understanding of the application and advancement of theory.

The preliminary knowledge regarding these algorithms further would be convenient for researchers working under the data science and research field for applying them in the real-world NP-hard combinatorial problems. This research may be used as an initial reading point to understand and explore the mentioned nature-inspired algorithms for their real-life application in the organizational, social, as well as management space, which depends on their business problem's nature as well as issues connected with goals, computation space, and constraints.

CHAPTER 3

NATURE-INSPIRED ALGORITHMS FOR EALTHCARE: GWO & WOA

This chapter puts forward Grey Wolf Optimization Algorithm and Whale Optimization Algorithm for optimal feature selection from the dataset of cardiotocography. Features selected by GWO and WOA are optimal reduced set of features for classification of state of the fetus under normal, suspect and pathologic.

3.1 Introduction

Electronic Fetal Monitor (EFM) is a general name for cardiotocograph (CTG) machine that checks the well-being of fetus by monitoring fetus's heartbeat and uterine contractions by making use of ultrasound transducer for sensing mother's abdomen during pregnancy [207]. The process carried out is formally known as Cardiotocography. A mother with medical conditions tends to face high risk pregnancies which require incessant monitoring for the fetus's well-being and reduces the chance of neo-natal seizures at the time of labor [208].

Fetal hypoxia is a condition in which amount of oxygen falls short for the fetus that leads to fetal distress, temporary impairment and can further prove to be fatal. Risk associated with prolonged effect of hypoxia is baby born with disabilities or still births. At the time of monitoring the condition of the fetus, proper distinction has to be maintained between maternal heart rate and fetal heart rate. One needs to ensure that by mistake; maternal heart rate is not taken into consideration instead of fetal heart rate. Thus, in order to avoid confusion maternal heart rate is checked separately. Fetal Scalp Electrode (FSE) is used, if by chance one monitors maternal heart rate instead of fetal heart rate [209].

The major finding of this chapter has been accepted and published in 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet Tunisia.

Vaginal bleeding, placenta insufficiency, lessened fetal movements, multiple pregnancy, intra uterine growth are the conditions that heightens the risk associated with the fetus. This research reflects the optimal feature selection behavior of Grey Wolf Optimization algorithm and Whale Optimization Algorithm for the cardiotocograph dataset.

Various machine learning classifiers are used to depict the optimality of features selected with the help of accuracies computed for GWO and WOA. The cardiotocography dataset can be obtained from UCI Machine Learning repository (https://archive.ics.uci.edu/ml/machine-learning-databases/00193/). GWO and WOA feature selection comprises of striking features with an average accuracy of 98.74% and 98.11%.

Highlights of the research are as mentioned below:

- Grey Wolf Optimization algorithm and Whale Optimization Algorithm are used for classification of fetal state.
- GWO and WOA selects the most distinguishing features.
- The mentioned algorithms GWO and WOA have been exercised on the cardiotocography dataset with the average accuracy of 98.74% and 98.11%.
- To understand the optimality value of features selected by GWO and WOA; machine learning classifiers: (i) DecisionTree (ii) K-Nearest Neighbor (KNN) (iii) Random Forest are used to obtain the accuracy values.
- Comparison of nature inspired algorithms GWO and WOA, has been presented in terms of features selected and computed accuracies.

3.2 Background Study

3.2.1 Grey Wolf Optimization Algorithm(GWO)

Mirjalili et al. 2014, developed a swarm intelligence bio-inspired algorithm namely Grey Wolf Optimization (GWO). Grey wolves are renowned for their skills of group hunting while targeting the prey. They possess a superseding social chain of command. A male or a female can represent the pack of wolves as a leader and is represented as the Alpha ($\boldsymbol{\alpha}$);

who is responsible for taking important decisions and imparting orders that are to be followed by wolves in the pack. The next in the hierarchy are Betas. They are next in rank to Alpha and help in their role of decision making. Their role is to uphold discipline in the pack. Next in the hierarchy are the delta wolves, who control their subordinates in the hierarchy i.e. omega wolves. The mathematical modeling of technique of hunting of grey wolves, their ranking hierarchy and their roles and responsibilities frames Grey Wolf Optimization in lieu of feature selection to get to the bottom of optimization problems [52] [210]. In terms of exploration potentials GWO outpaces other swarm intelligence techniques [36]. GWO addresses an important drawback faced by majority of the swarm intelligent techniques i.e., leaders tend to be in command for whole duration over the pack of wolves whereas in other techniques leaders have control for limited duration. In comparison to other algorithms, GWO is easy to implement as it comprises of fewer parameters.

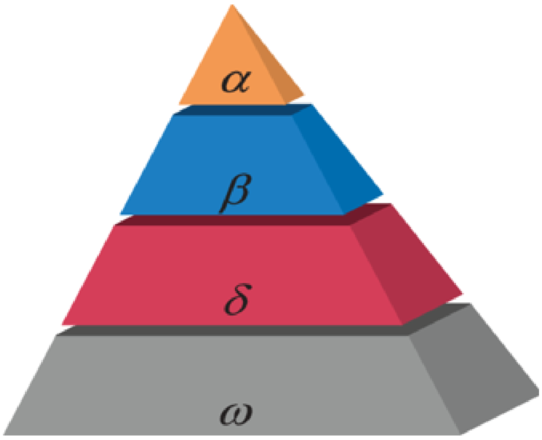


Fig. 3.1: Hierarchy in pack of Grey Wolves

The three best solutions i.e., alpha, beta and delta grey wolves demonstrate the leadership hierarchy of grey wolves. Other candidate solutions are represented as omega wolves. Fig. 3.1 represents pecking order of grey wolves. At the time of hunting initially the prey is encircled which is depicted by equation (3.1) and equation (3.2).

$$\vec{D} = |\vec{C}.\vec{X}_{\text{prey}}(t) - \vec{X}_{\text{wolf}}(t)| \tag{3.1}$$

$$\vec{X}_{\text{wolf}}(t+1) = \vec{X}_{\text{prey}}(t) - \vec{A}.\vec{D} \tag{3.2}$$

$$\vec{A} = 2\vec{a}\,.\,\vec{r}_1 - \vec{a} \tag{3.3}$$

$$\vec{C} = 2\vec{r}_2 \tag{3.4}$$

In the above equations $\overrightarrow{A}$ and $\vec{C}$ represents the coefficient vectors. Their values can be obtained using equation (3.3) and equation (3.4). X_{prey} and X_{wolf} depicts the position vector for the prey and grey wolf. r_1 and r_2 are the vectors generated randomly and their value lies in the interval of [0, 1]. For total count of iterations, linearly value of $\vec{a}$ decreases from 2 to 0.

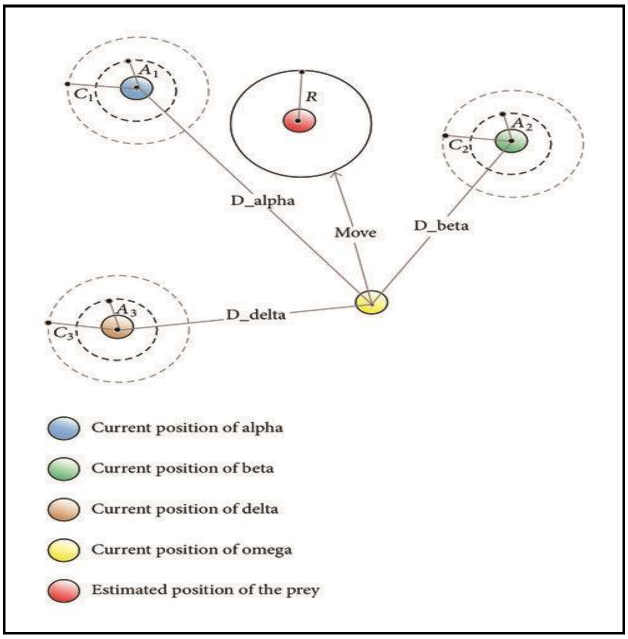


Fig. 3.2: Current and updated positions of Grey Wolves

The process of hunting is led by alpha wolves which in turn are assisted in making decision and hunting by the beta and delta wolves. The pecking order and hunting behavior of grey wolves is imitated by the three best solutions. The solutions computed are saved and the position of omega wolves i.e. the search agents is updated consequently. Fig. 3.2 depicts the update in position of omega grey wolves. Equations (3.5) – (3.11) exhibits the position update of grey wolves.

$$\vec{D}_{alpha} = |\vec{C}_1 . \vec{X}_{alpha} - \vec{X}| \tag{3.5}$$

$$\vec{D}_{beta} = |\vec{C}_2 . \vec{X}_{beta} - \vec{X}| \tag{3.6}$$

$$\vec{D}_{delta} = |\vec{C}_3 . \vec{X}_{delta} - \vec{X}| \tag{3.7}$$

$$\vec{X}_1 = \vec{X}_{\text{alpha}} - \vec{A}_1.\ \vec{D}_{\text{alpha}} \tag{3.8}$$

$$\vec{X}_2 = \vec{X}_{\text{alpha}} - \vec{A}_2.\ \vec{D}_{\text{beta}} \tag{3.9}$$

$$\vec{X}_3 = \vec{X}_{\text{delta}} - \vec{A}_2.\ \vec{D}_{\text{delta}} \tag{3.10}$$

$$\vec{X}(\text{t+1}) = \frac{\vec{X}1 + \vec{X}2 + \vec{X}3}{3} \tag{3.11}$$

3.2.2 Whale Optimization Algorithm (WOA)

Swarm intelligence algorithm i.e. WOA is proposed for problems with continuous development. This algorithm proves that, it has better performance compared to other well-known algorithmic techniques [53]. WOA gets its inspiration from hunting pattern and behavior of humpback whales as shown in fig. 3.3. In WOA, whale is considered in the form of solution i.e. each solution depicts a whale. The whale attempts to fill a new area in the search area refer to the best feature of the group. Method used by whales to hunt location and attack is as follows.

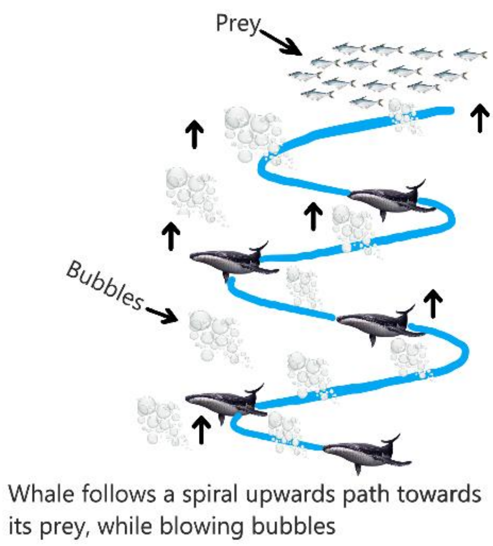


Fig. 3.3: Hunting pattern of humpback whales

- **Inspiration**

Whales are largely well-thought-out to be marauders. Whales breathe from the surface of oceans coz of which they never sleep. The interesting thing about whales is that they are considered to be very smart and emotional animals. Hof and Gucht [211] believe that certain areas of the human-like whale's brain have common cells known as spindle cells. The main reason for cetacean intelligence is that these cells are higher in number in comparison to the number in adults. Their hunting behavior is called bubble net feeding [212]. However, in [213], it was studied that this behaviour uses label sensing.

- **Encircling prey**

Humpback whales can make out their prey and also envelop their location. Since it is not conceivable to deduce from the search space the position of optimal design, WOA undertakes the best candidate solution to be the target prey or the one closest in proximity to the optimal solution. Once the best search agent has been developed and determined, the next stage may be for other search agents to attempt to update their location to the best search agent. This behaviour is expressed in the following equation (3.12) and equation (3.13) i.e. with respect to humpback whales is as follows:

$$\vec{D} = |\vec{C} \cdot x^{*}(t) - \vec{x}(t)| \tag{3.12}$$

$$\vec{x}(t+1) = \vec{x}^{*}(t) - \vec{A} . \vec{D} \tag{3.13}$$

$$\vec{A} = 2\vec{a}\vec{r} - \vec{a} \tag{3.14}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{3.15}$$

- **Bubble-net attacking method (exploitation phase)**

The bubble behavior of humpback whales was modeled mathematically using the following two approaches:

1) Narrow enclosure: the value obtained of Equation (3.14) and Equation (3.15) is downgraded to achieve this behavior. The point to be reflected here is that the amplitude of oscillation is also reduced. That is, $\vec{A}$ represents a random value in the range [-a, a] which is lowered from 2 to 0 over iterations. By setting random values for $\vec{A}$ [-1, 1].

2) Spiral update position: This method tends to compute first the distance between the particular position of a whale and its prey. Equation (3.16) is a spiral-centered equation that is positioned between the position of the whale and its prey to mimic the helix-like motion associated with humpback whales is as follows:

$$\vec{X}(t+1) = \vec{D}'' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}' \tag{3.16}$$

- **Search for prey (exploration phase)**

The vector-based approach can be used in the process of finding prey (exploration). Indeed, depending on each other's positions, humpback whales search for each other at random. Therefore, was used with random values greater than 1 or less than 1 to maneuver the searcher away from a reference whale. Unlike Mining Phase, the agent position upgraded in Discovery Phase is based on a random method.

3.3 Methodology

3.3.1 *Methodology Used*

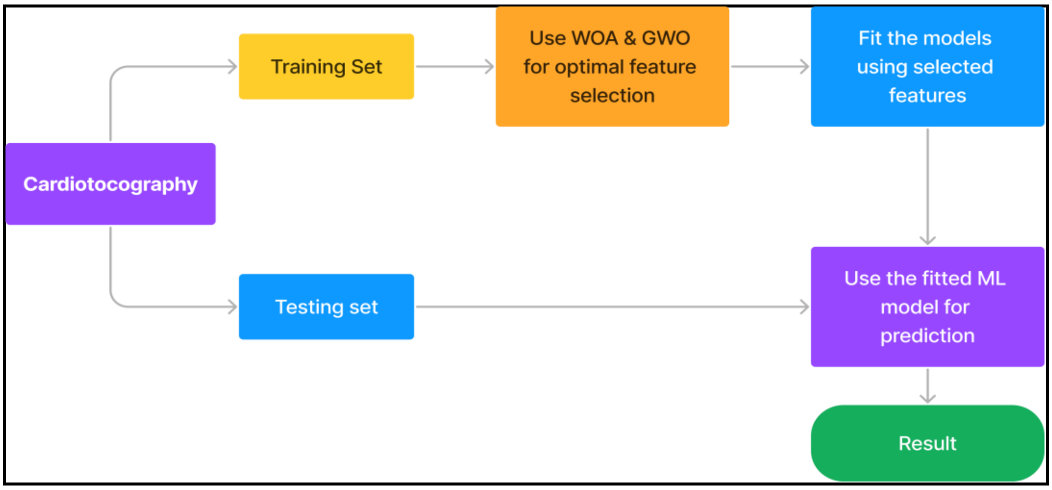


Fig. 3.4: Flowgraph of the Methodology

The cardiotocograph dataset is divided in training and testing sets in the ratio of 70:30, it comprises of features of 2126 fetal cardiotocograms. GWO and WOA selects the optimal features when exercised with the training set.Machine learning classifiers are employed to obtain accuracy in percentage of the features selected by GWO and WOA. The prediction and classification of fetus under various pathalogic states is the final result. Fig. 3.4 explains the methodology used by dividing it into various stages.

3.3.2 Dataset Description

The dataset comprises of values of 2126 fetal cardiotocograms(CTGs) classified by expert obstetricians for diagnostic features Fetal Heart Rate (FHR) and uterine contraction (UC). For assigning the label of classification, three expert obstetricians were involved in the process. Classification was done for morphologic pattern (A, B, C. ...) and fetal state (N, S, P).

The dataset comprises of 21 significant features of each cardiotocogram. Out of these 21 features;11 are obtained with the sensors of Electronic Fetal Monitor (EFM) machine and further these 11 features helps in extracting the remaining 10 features. Table 3.1 exhibits mapping of different problem types and their associated values in multi-classification problem.

Table 3.1: Mapping of problem Type and their values in Cardiotocography Dataset

Problem Type	**Multi-Classification Problem**
Types of features	Real
Total attributes	23
Total samples	2125
Total missing values	0

3.4 Results

In this section, number of optimal features selected by nature inspired algorithms WOA and GWO on cardiotocography dataset has been analyzed and depicted. Out of total of 21 features, 7 features are selected by WOA, whereas GWO tends to select even more reduced set of features i.e. 4 as shown in Table 3.2. The comparison of optimal subset of features selected by WOA and GWO is depicted in Fig. 3.5. Table 3.3 exhibits accuracy of different machine learning classifiers like Decision Tree, Random Forest and KNN when applied to all the features of CTG dataset and the optimal ones selected by GWO. On performing comparative analysis there is an average increase of 2.5% and above in terms of accuracy when using GWO. GWO performs better then WOA in terms of reduced feature set selected and also accuracy computed in percentage. Table 3.4 shows comparison of GWO and WOA on the basis of features selected by different nature-inspired algorithms, i.e. well depicted in Fig. 3.6.

Table 3.2: No. of features selected using different nature-inspired algorithms.

Algorithm Used	Number of features selected
WOA	7
GWO	4

Table 3.3: Accuracy of different ML classifiers on all features and features selected by GWO

Machine learning Classifier	Accuracy with feature selection (%)	Accuracy without feature selection (%)
Decision tree	98.22	96
Random forest	98.50	95.69
KNN	98.43	86.33

Table 3.4: Comparison of accuracy obtained by ML classifiers on the basis of features selected by different nature-inspired algorithms

Algorithm	K- Nearest Neighbour (Accuracy %)	Random Forest (Accuracy %)	Decision Tree (Accuracy %)
WOA	97.64	98.11	98.11
GWO	98.43	98.74	98.43

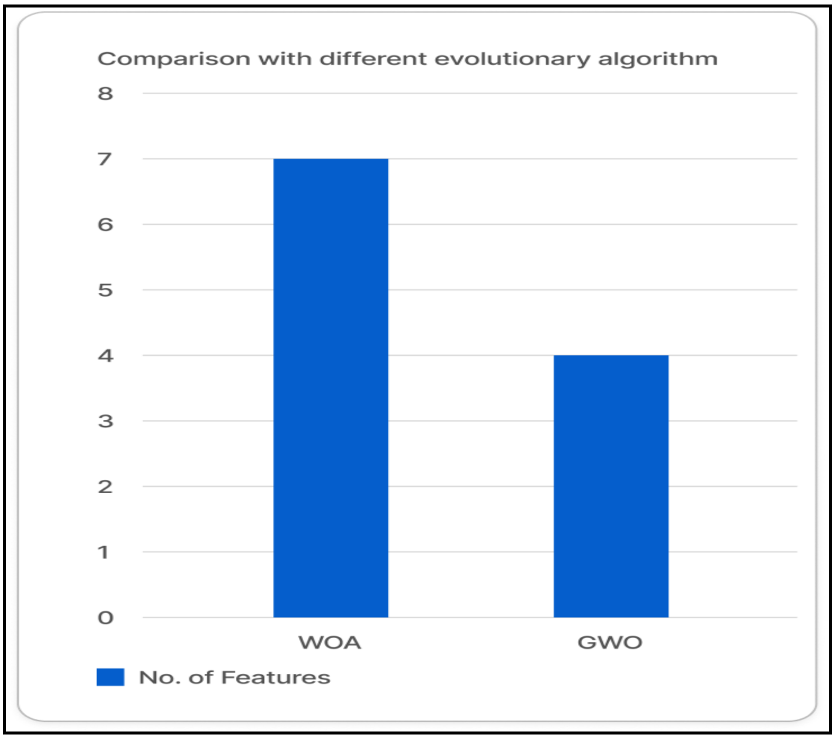


Fig. 3.5: Number of features selected using different Nature-Inspired Algorithms

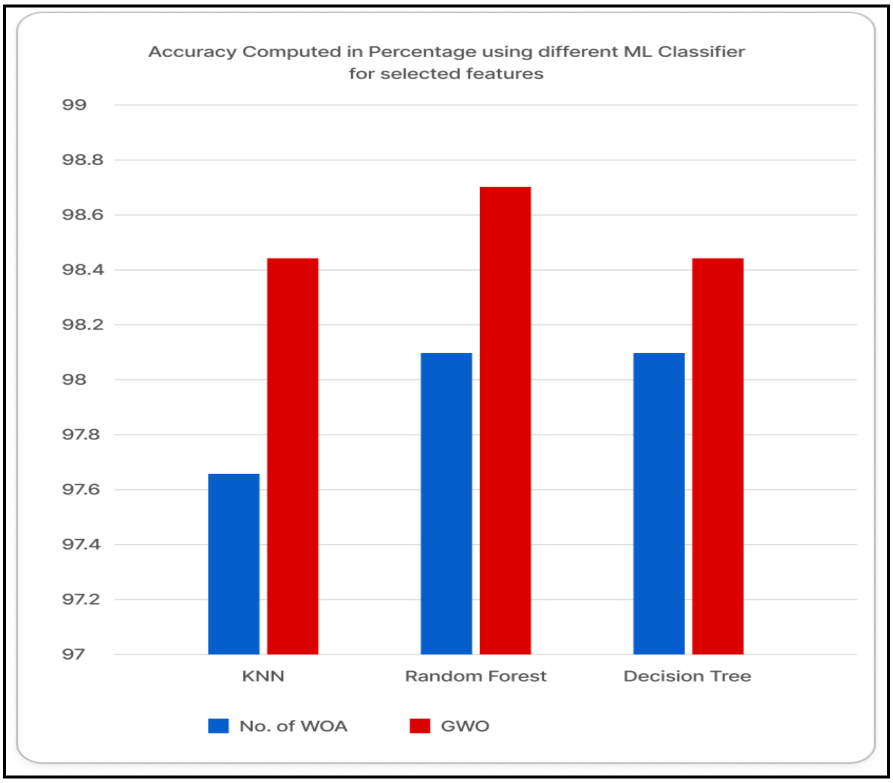


Fig. 3.6: Accuracy computed in percentage using different ML classifiers for features selected using WOA and GWO

3.5 Conclusion

This research work exhibits feature selection properties of nature-inspired algorithms Grey Wolf optimization and Whale Optimization Algorithms. These algorithms drop extraneous and irrelevant features thereby performing dimensionality reduction which greatly reduces the cost of computation. GWO and WOA have made a selection of reduced set of optimal features 4 and 7 respectively. The optimality of features selected are computed in terms of average accuracy of 98.74% and 98.11% for GWO and WOA using machine learning classifiers like KNN, decision tree and random forest.

CHAPTER 4

QUANTUM BINARY BAT ALGORITHM (QBBA) FOR LEUKOCYTE CLASSIFICATION

This chapter presents a systematized solution for the classification of leukocytes in blood smear. The proposed model incorporates the optimistic aspects of nature inspired and quantum inspired algorithms; this model tends to be perfect blend of both the techniques for reducing the dimensionality i.e. irrelevant features.

4.1 Introduction

Leukocytes (or White Blood Cells) frame the considerable element of the immune system. They are found in blood and lymph like body fluids. The prime role of leukocytes is to defend the body against the foreign substances that are threat to it. A Leukocyte is an amoeboid cell that has nucleus and other organelles. It can be classified under two heads i.e. granulocytes and agranulocytes. In the case of granulocytes granules are held inside cytoplasm and are further categorized as neutrophils, eosinophils, and basophils. On the other hand, agranulocytes can be categorized into lymphocytes and monocytes. The three granulocytes are accountable for fighting against the encountered infection and the abovementioned two agranulocytes is answerable for directing full body responses against germs. Deficiency of leukocytes can lead to disorders related to bone marrow or the one affecting the immune system. Thus, the significant concentration of various types of leukocytes is essential for effective functioning of the immune system.

Differential analysis of White Blood Cells (or leukocytes) takes account of number and relative percentage of aforementioned leukocytes. In medical science this process is carried out time and again for the identification of the disease.

Differential analysis of White Blood Cells (or leukocytes) is basically done to get an approximate count of total number of cells in the complete blood sample or a blood smear which is observed manually under a microscope and the count of cells in a particular region is taken into consideration.

The major findings of this chapter have been accepted and published in Journal of Expert Systems, Volume 31 September 2021, Wiley (SCIE Indexed IF: 2.812).

Since the process is carried out manually it becomes susceptible to human errors; for this reason, various machine learning models have been proposed over the time to enhance the efficacy of this process by reducing the errors.

Two broad classes of leukocytes tend to depend on the presence of granules in the cytoplasm which are identified with the aid of segmentation of microscopic images of blood smears as it is able to differentiate between cytoplasm and the nucleus of the cell; hence it is the earliest move of a machine learning model for differential analysis. Afaf Tareef et al. [199] put forth a technique for fully automated image segmentation of leukocytes based on the color and textures of the microscopic cell images. Der-Chen Huang et al. [200] proposes segmentation by enhancing the nucleus of the cell and suppressing the other regions like blood smear. A variety of other techniques such as Geometric Active Contours and Level Set Method have also been implemented by Khamael AL Dulaimi et al. [201], which can easily segment nuclei from cell walls and cytoplasm.

After the segmentation is done a set of features are extracted and they are trained with the help of a classifier. Out of the features that are extracted few of the features may be irrelevant and not so effective to the classification model which can further make the training process complex. In addition, handling fewer features is cost effective and greatly reduces the computation time. Thus, if it is observed from the stand points of economics and efficiency then identification of appropriate subset of features is of utmost importance. In this research, evolutionary algorithm is used and its work is further refined and enhanced by incorporating quantum principles. Quantum-inspired Binary Bat Algorithm, is used for feature selection and the selected features are tested using a number of different classifiers.

Quantum Computing or Quantum Computers are in the forefront of innovation. Quantum Computers are still in their infancy as they operate on a disparate hardware. The traditional bit in a quantum computer is replaced by a quantum-bit or q-bit, which stays in a quantum state, allowing it to store both 0 and 1 at the same time, following the superposition of states, a basic principle of quantum physics. Akama S. in his book [214] discusses the fundamentals, algorithms and future scope of quantum computing. P. W. Shor [215] [216] proposed a quantum algorithm for computation of discrete logarithms and factoring, he further uses this approach for prime factorization of large numbers, something which isn't possible using traditional computers. Quantum Computers are a powerful tool and will one day replace the traditional computers entirely. However, due to lack of resources and technologies, quantum-inspired computing has emerged which uses the concepts and fundamentals of a quantum computer to develop algorithms that can run on a conventional computer with ease. Kuk-Hyun Han et al. [217] proposed a class of quantum-inspired evolutionary algorithms and discovered that, quantum-inspired algorithms can deliver powerful results even with a smaller population without impetuous convergence. Abdesslem Layeb [218] [219] proposed a quantum inspired cuckoo search algorithm as well as a harmony search algorithm for 0-1 knap sack problems. Both of these algorithms delivered effective results and performed better than their traditional counterparts. S. Zhao [220] also proposed a novel quantum-inspired algorithm for training of fuzzy neural networks, and demonstrates its effectiveness by training non-linear functions using the proposed methods. It is safe to say, that a quantum-inspired optimization algorithm performs better in most of the cases and increases the efficiency by reducing the risks of premature convergence.

D. Gupta et al. [33] proposed the application of the Binary Bat Algorithm for the classification of Leukocytes and observed that the Binary Bat Algorithm (BBA) performed better than any other heuristic algorithm for the problem at hand i.e. the classification of leukocytes in blood smear to perform automated differential analysis for medical personnel. The following article introduces a quantum-inspired variant of the conventional Binary Bat Algorithm and compares the performance of the QBBA alongside its traditional variant in order to quantise the effects of quantum optimization on the Binary Bat Algorithm. The proposed algorithm is trained against the data-set of microscopic images of blood smear containing various leukocytes accumulated by the Imam Khomeini hospital's Haematology-Oncology and BMT Research Centre in Tehran, Iran [18]. The efficiency and accuracy of the algorithm is tested against well-established classifiers which is then compared with the results obtained by its traditional counterpart. It was observed that the QBBA performed better than BBA for a comparable population size and produced an accuracy of 98.31% as opposed to the 97.56% produced by BBA, which is in consistency with the expected results. It can also be concluded that QBBA is one of the most powerful algorithms for classification of leukocytes and can find its applications in the diagnosis of haematological diseases.

4.2 Background

4.2.1 Bat Algorithm

Bats are a group of mammals from the order or Chiroptera, which means that their forelimbs have adapted into wings over time, giving them the ability to fly. Approximately 70% of all bats are micro-bats, which is a species of bats that is characterised by their small size and the use of echolocation for navigation and hunting of prey. Contrary to popular belief, not all bats use echolocation. Micro-bats emit a high frequency sound wave while flying and listen to the reflecting sound waves for the reconstruction of the neighbouring environment as a sonic layout. Different species of the micro-bats emit sound waves of varied frequencies and bandwidths, which is a function of their hunting strategies.

Bats adjust pulse rate (or frequency) and loudness (or amplitude) of the emitted waves depending on their distance from the prey, emitting louder sound waves at a lower frequency when in search of prey and reducing the amplitudes, with an increased frequency when heeding towards them. Waves with a higher amplitude and lower frequency are used to detect the presence of prey within the proximity of the bat, whereas a lower amplitude and higher frequency allows the bat to pin point the exact location of the prey.

Fig. 4.1 depicts the use of echolocation for hunting as used by micro bats.

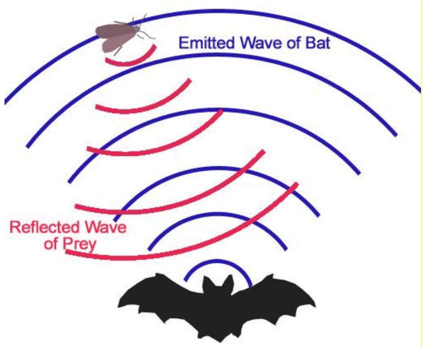


Fig. 4.1: Echolocation in Micro Bats

Yang [28] proposed a novel meta-heuristic algorithm based on the echolocation of micro bats with an underlying assumption that the micro bat identifies the difference between an obstacle and its prey and only changes its behavior when it is in proximity of a prey and not an obstacle. The said algorithm imitates the behavior of a colony of micro bats hounding their prey.

It is assumed that the bats reside in an n-dimensional vector space, where n is the number of optimization points of the problem. The current position of the i^{th} bat flying with a velocity V_i is represented as X_i. The waves emitted are assumed to be within the frequency range [f_{min}, f_{max}], with an initial loudness A_0 , and the rate of pulse emission as r. The position and velocity of the virtual micro-bat at a (t+1) is realized as a function of its frequency, position and velocity using equation (4.1), equation 4.2 and equation 4.3.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{4.1}$$

$$V_i = V_i + (X_i - X_{best})f_i \tag{4.2}$$

$$X_i = X_i + V_i \tag{4.3}$$

Here, β is a random number between (0,1) and X_{best} is the position of the global best solution up to the present time in the algorithm.

It is assumed that the loudness (A_i) varies from A_0 to A_{min} as the bat progresses towards its prey and the rate of pulse emission i.e. r_i, increases simultaneously. For ease of calculations, A_0 is set as 1 and A_{min} as 0. The initial rate of pulse emission is set at r_0. A_i and r_i, progress using equation (4.4) and equation (4.5).

$$A_i = \alpha A_i \quad (4.4)$$

$$r_i = (r_0)_i(1 - e^{-\lambda t}) \quad (4.5)$$

Here, α and λ are constants within the range of (0,1). For the ease of our calculations $\alpha = \lambda = 0.9$. From equations (4.4) and (4.5), we can infer that as$t \rightarrow \infty, A_i \rightarrow 0 and r_i \rightarrow (r_0)_i$, which is in coherence with the behavior of micro bats, assuming that the bat reaches its prey as $t \rightarrow \infty$. Fig.4.2 depicts the variations in position of a bat after every iteration of the bat algorithm.

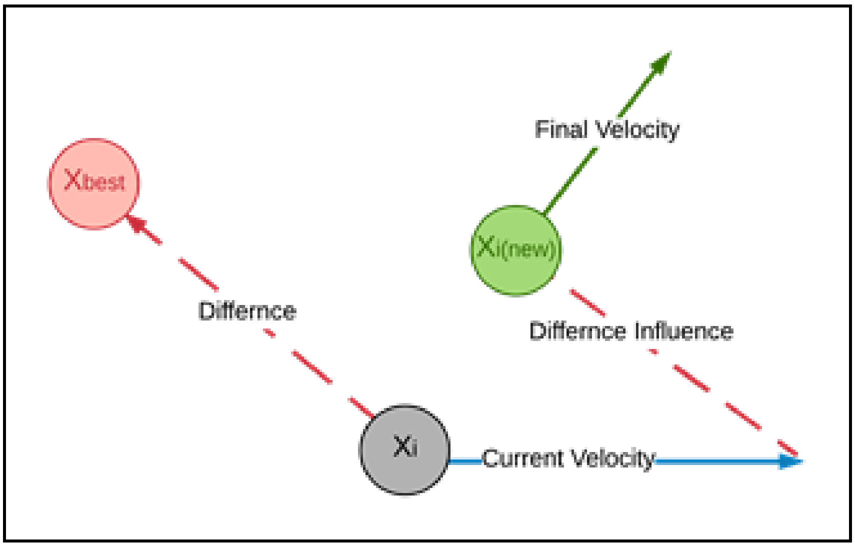


Fig.4.2: Position Variations of a Bat Algorithm

The bat algorithm is run for itr iterations and after every iteration. The bats are rated based on their distance from the prey, which is determined using the fitness function of the optimization problem. The bat with the most optimum solution is chosen as the global best solution for the problem, at the end of that particular iteration. The global best after itr iterations is chosen as the optimum solution for the problem.

The following Algorithm 1, depicts the fore-mentioned bat algorithm in detail.

Algorithm 1: Bat Algorithm

1. Initialise the population of the virtual micro-bats with the frequency ($\boldsymbol{f_i}$), position ($\boldsymbol{X_i}$), and flight velocity ($\boldsymbol{V_i}$).
2. Initialise the maximum loudness ($\boldsymbol{A_0}$) and maximum pulse emission rate ($\boldsymbol{r_0}$) for each bat ($\boldsymbol{b_i}$).
3. Set the maximum number of iterations as ***itr***.
4. While ***t < itr*** repeat steps 5 to 18.
5. For each virtual bat (bi), repeat the steps 6 to 16.
6. Use equations (4.1), (4.2) and (4.3) to adjust the frequencies, and vary the velocities as well as positions to generate a new solution.
7. Generate a random number k within the range (0,1).
8. if $k > r_i$
9. Select a random solution among the best solutions.
10. Generate a new solution local solution neighboring the randomly selected solution.
11. End if.
12. Generate a new solution y flying randomly.
13. if $k < A_i$ && $Cost(X_i) < Cost(X_{best})$
14. Accept the new solution.
15. Modify A_i and r_i according to equations (4.4) and (4.5).
16. End if.
17. End For.
18. Rank the bats based on their values of the Cost Function and select the globally best solution.
19. End While.

4.2.2 Feature Selection & Binary Bat Algorithm

Feature Selection or variable selection is a dimensionality reduction technique that is often used in machine learning to derive an optimum subset of features that can be used for model training. Feature Selection is necessary for the elimination of irrelevant features, in order to improve the performance of the system. Training a data-set with a large number of features requires a computer system with higher computational power, and significantly increases the cost of computations. Hence, feature selection not only eliminates irrelevant features but also reduces the cost of computation to a great extent. Isabelle Guyon et al. [221] introduces feature selection and a variety of techniques for its implementation. A variety of evolutionary algorithms find their implementation in feature selection and Bat Algorithm is no stranger to them.

The Bat Algorithm proposed by Yang [28] works best for continuous values, whereas, feature selection is a binary problem. R. Y. M. Nakamura et al. [30] proposed a Binary Bat Algorithm for 0-1 (or Binary) optimization problems and cited that BBA outperformed a various other evolution-based feature selection algorithm. The Binary Bat Algorithm reduces the n-dimensional vector space for the movements of the bats to a Boolean Hypercube of n dimensions, wherein a bat is only allowed to move across certain fixe points in the lattice structure, the nodes and corners. The BBA uses a sigmoid function to restrict the movement of the bat as per the specified parameters. The following equation (4.6) and equation (4.7) represent the transaction functions for transformation of the bat's position in terms of 0 or 1.

$$S(V_i) = \frac{1}{1+e^{-V_i}} \tag{4.6}$$

$$X_i = \{1, S(V_i) > \rho 0, Otherwise\} \tag{4.7}$$

Here, ρ is a random variable from the uniform distribution of (0,1). For feature selection, the n dimensions of the Boolean Hypercube are set to the number of features in the optimization problem, wherein the value 0 represents the absence of the feature from the extracted subset and 1 represents its presence.

The following Algorithm 2 explains the fore-mentioned Binary-Bat Algorithm in detail.

Algorithm 2: Binary Bat Algorithm

1. Initialise the population of the virtual micro-bats with the frequency ($\boldsymbol{f_i}$), position ($\boldsymbol{X_i}$), and flight velocity ($\boldsymbol{V_i}$).
2. Initialise the maximum loudness ($\boldsymbol{A_0}$) and maximum pulse emission rate ($\boldsymbol{r_0}$) for each bat ($\boldsymbol{b_i}$).
3. Set the maximum number of iterations as ***itr***.
4. While $\boldsymbol{t < itr}$ repeat steps 5 to 18.
5. For each virtual bat (bi), repeat the steps 6 to 16.
6. Use equations (4.1), (4.2), (4.6) and (4.7) to adjust the frequencies, and vary the velocities as well as positions to generate the solution.
7. Generate a random number k within the range (0,1).
8. if $k > r_i$
9. Select a random solution among the best solutions.
10. Generate a new solution local solution neighbouring the randomly selected solution.
11. End if.
12. Generate a new solution y flying randomly.
13. if $k < A_i$ && $Cost(X_i) < Cost(X_{best})$
14. Accept the new solution.
15. Modify A_i and r_i according to equations (4.4) and (4.5).
16. End if.
17. End For.
18. Rank the bats based on their values of the Cost Function and select the globally best solution.
19. End While.

The following Fig. 4.3 depicts the flowchart for the Binary Bat Algorithm. Here, F, V and X represent Frequency, Velocity and Position respectively.

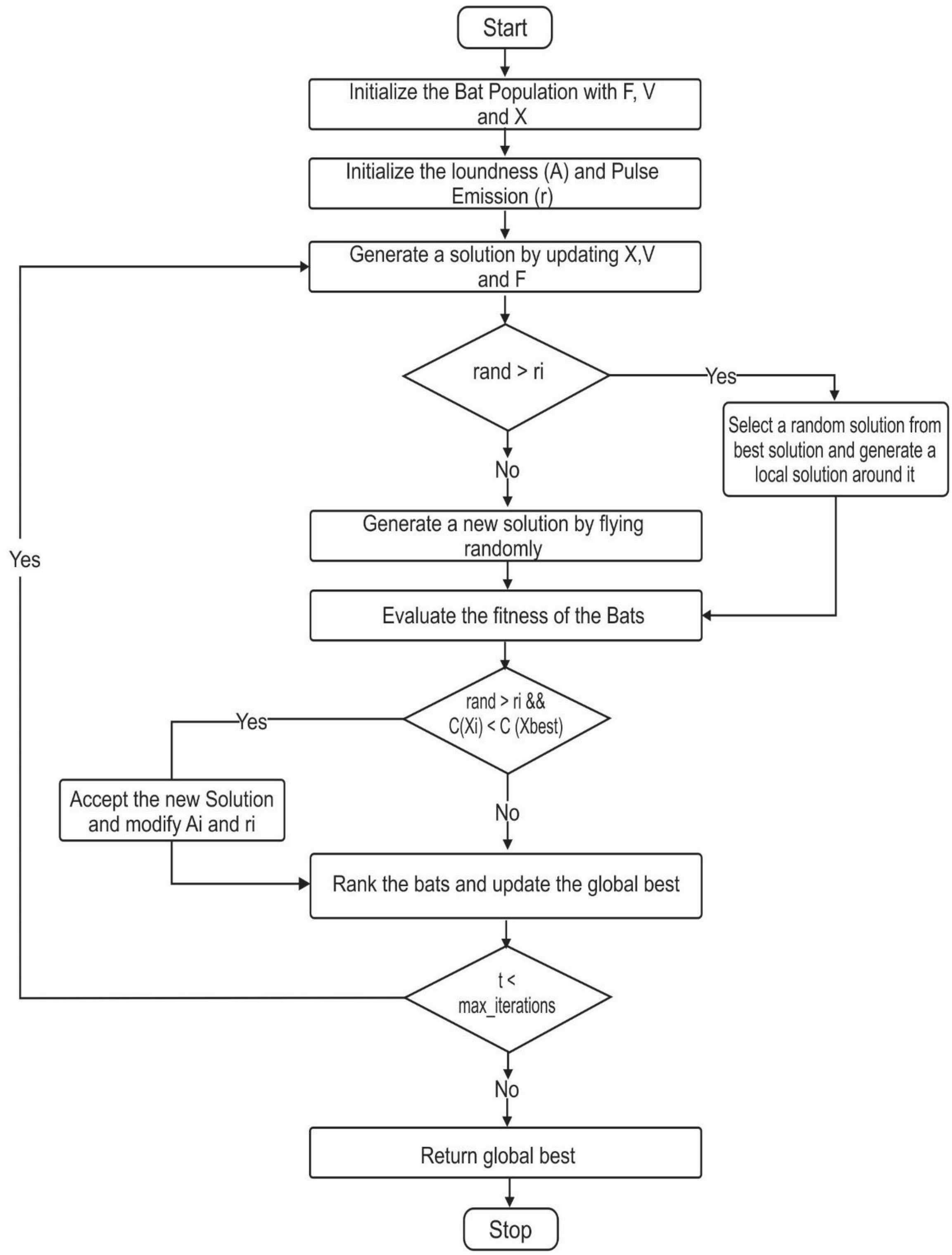


Fig. 4.3: Binary Bat Algorithm

4.2.3 Other Variations of the Binary Algorithm

Bat Algorithm is a powerful meta-heuristic algorithm which has been modified in-numerous times. Xin-She Yang [222] reviewed and compared the variations of the bat algorithm put forth up until 2013, which include the Fuzzy Logic Bat Algorithm proposed by Khan et al. [223], a Multi-Objective Bat Algorithm proposed by Yang [224], K-Means Bat Algorithm Proposed by Komarasamy et al. [225], and numerous others. Deepak Gupta et al. [32] proposed a variation of the bat algorithm for usability features of a software and concluded that the proposed algorithm performed better than the BA or any other evolutionary algorithm with a performance comparable to that of the Bat Algorithm.

4.3 Quantum-Inspired Binary Bat Algorithm

4.3.1 Quantum Optimization

A Quantum Computer is a computing device that replaces the building block of a traditional computer i.e., a bit by a quantum bit or a q-bit, which exhibits the properties of a quantum particle. A quantum bit, unlike any quantum particle possess two properties, the super-position of states and the entanglement of particles. As a result, it is neither 0 nor 1, but rather a combination of the two states, and is represented in terms of the probabilities of both 0 and 1. As a result, a single q-bit stores two continuous values ad opposed to one binary value.

The following equation (4.8) and equation (4.9) are the quantum representation and the necessary conditions for a q-bit respectively, with αbeing the probability of state 1 and βbeing the probability of state 0.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{4.8}$$

$$|\alpha|^2 + |\beta|^2 = 1 \tag{4.9}$$

The values of $\alpha$and $\beta$are always within the range of (0,1). The proposed research uses equations 4.8 and 4.9 to derive a quantum-inspired algorithm with normalised values such that each particle of the evolutionary algorithm can be thought of as a q-bit. A quantum-inspired evolutionary algorithm uses a complex number to represent a q-bit [6, 13] such that, $\alpha = cos\ \theta$ and $\beta = sin\theta$ for the complex number $e^{i\theta}$. The system does not store $\alpha$and β separately but rather stores θ and uses it to derive the subsequent real and imaginary values for the complex number.

A quantum particle is usually represented as a vector of probabilities of its possible states $[\alpha\beta]$, which eases the implementation of rotation and not gates, the two fundamental gates of a quantum computer. The following equation (4.10) and equation (4.11) represent the rotation and not gates respectively.

$$R(\Delta\theta) = [cos(\Delta\theta) - sin(\Delta\theta) sin(\Delta\theta) cos(\Delta\theta)] \tag{4.10}$$

$$Not - gate = [0110] \tag{4.11}$$

The rotation gate can be implemented in a complex representation by simply varying the magnitude of theta i.e. $e^{i(\theta+\Delta\theta)}$, the not-gate, simply interchanges the magnitudes of the probabilities of the two states, namely 0 and 1. Hence, the not-gate can easily be implemented by modifying the argument from θ to $\frac{\pi}{2} - \theta$ i.e. $e^{\frac{\pi}{2}-\theta}$.

4.3.2 Quantum –Inspired Binary Bat Algorithm

The quantum binary bat algorithm is assumed to be at two locations simultaneously which are determined using the modified equations of the system. The quantum bat's position is denoted as ϕ as apposed to X in the bat as well as binary bat algorithms, which is within the range of (0, 2π). The position is then modified to derive two values for the two positions of the binary bat.

The fitness function is evaluated at both of these locations and the optimum among the two is selected for further computations. The following equations (4.12), (4.13), (4.14), (4.15), (4.16) and (4.17) illustrate the transformation from 4.2, 4.6 and 4.7 of the binary bat, into the quantum-inspired binary bat algorithm.

$$V_i = V_i + (\Delta\phi_i)_{best} f_i \tag{4.12}$$

$(\Delta\Phi_i)_g = \big[2\pi + (\Phi_i)_g - \Phi_i, \big((\Phi_i)_g - \Phi_i < -\pi\big)(\Phi_i)_g - \Phi_i, \big(-\pi < (\Phi_i)_g - \Phi_i <$

$$\pi)(\Phi_i)_g - \Phi_i - \quad 2\pi, ((\Phi_i)_g - \Phi_i > \pi)] \tag{4.13}$$

$$\phi_i = \frac{\pi}{2} - (\phi_i + V_i) \tag{4.14}$$

$$(Y_i)_a = Real\left[\frac{1}{2}\left(\left(1 + e^{\Phi_i i}\right) - \left(1 - e^{\Phi_i i}\right)\right)\right]$$

$$(Y_i)_b = Img\left[\frac{1}{2}\left(\left(1 + e^{\Phi_i i}\right) - \left(1 - e^{\Phi_i i}\right)\right)\right] \tag{4.15}$$

$$S(Y_i)_a = \frac{1}{1+e^{-(Y_i)_a}}$$

$$S(Y_i)_b = \frac{1}{1+e^{-(Y_i)_b}} \tag{4.16}$$

$$(X_i)_a = \{1, S(Y_i)_a > \rho 0, Otherwise\}$$

$$(X_i)_b = \{1, S(Y_i)_b > \rho 0, Otherwise\} \tag{4.17}$$

The equation (4.12), (4.13) and (4.14) are used for transformation from a previous position of the quantum bat, to a new location. The equation (4.15) derives two simultaneous solutions for the two positions of the bat and transforms them within the range (-1,1). These two transformed solutions are then used for the derivation of two parallel feature matrices X_a and $X_{b,}$ using the equations (4.16) and (4.17). These solutions are then evaluated against the fitness or cost functions of the optimization problems and the better performing solution is used for the ranking of the bat. The following Algorithm 3 explains the proposed Quantum-Inspired Binary Bat Algorithm in detail.

Algorithm 3: Quantum-inspired Binary Bat Algorithm

---

1. Initialise the population of the virtual micro-bats with the frequency ($\boldsymbol{f_i}$), position ($\Phi_{\boldsymbol{i}}$), and flight velocity ($\boldsymbol{V_i}$).
2. Initialise the maximum loudness ($\boldsymbol{A_0}$) and maximum pulse emission rate ($\boldsymbol{r_0}$) for each bat ($\boldsymbol{b_i}$).
3. Set the maximum number of iterations as ***itr***.
4. While ***t*** < ***itr*** repeat steps 5 to 19.
5. For each virtual bat (b_i), repeat the steps 6 to 17.

6. Use equations (4.1), (4.12), (4.13) and (4.14) to adjust the frequencies, and vary the velocities as well as positions to generate the solution.
7. Generate a random number k within the range (0,1).
8. if $k > r_i$
9. Select a random solution among the best solutions.
10. Generate a new solution local solution neighbouring the randomly selected solution.
11. End if.
12. Generate a new solution y flying randomly.
13. if k < Ai && $Cost(X_i) < Cost(X_{best})$
14. Accept the new solution.
15. Modify A_i and r_i according to equations (4.4) and (4.5).
16. End if.
17. Use equations (4.15), (4.16) and (4.17) to generate two parallel solutions and compare the two to find the more optimum position of the bat.
18. End For.
19. Rank the bats based on their values of the Cost Function and select the globally best solution.
20. End While.

The following Fig. 4.4 depicts the flowchart for the Quantum-inspired Binary Bat Algorithm. Here, F, V and X represent Frequency, Velocity and Position respectively.

4.4 Methodology

4.4.1 Dataset Description

The Leukocytes Images for Segmentation and Classification (LISC) data-set was authored by Rezatofighi et al. [226], and consists of haematological images of the peripheral blood smears of healthy subjects. Humans are the perfect species for the study of leukocytes as it is easier to distinct their types within humans.

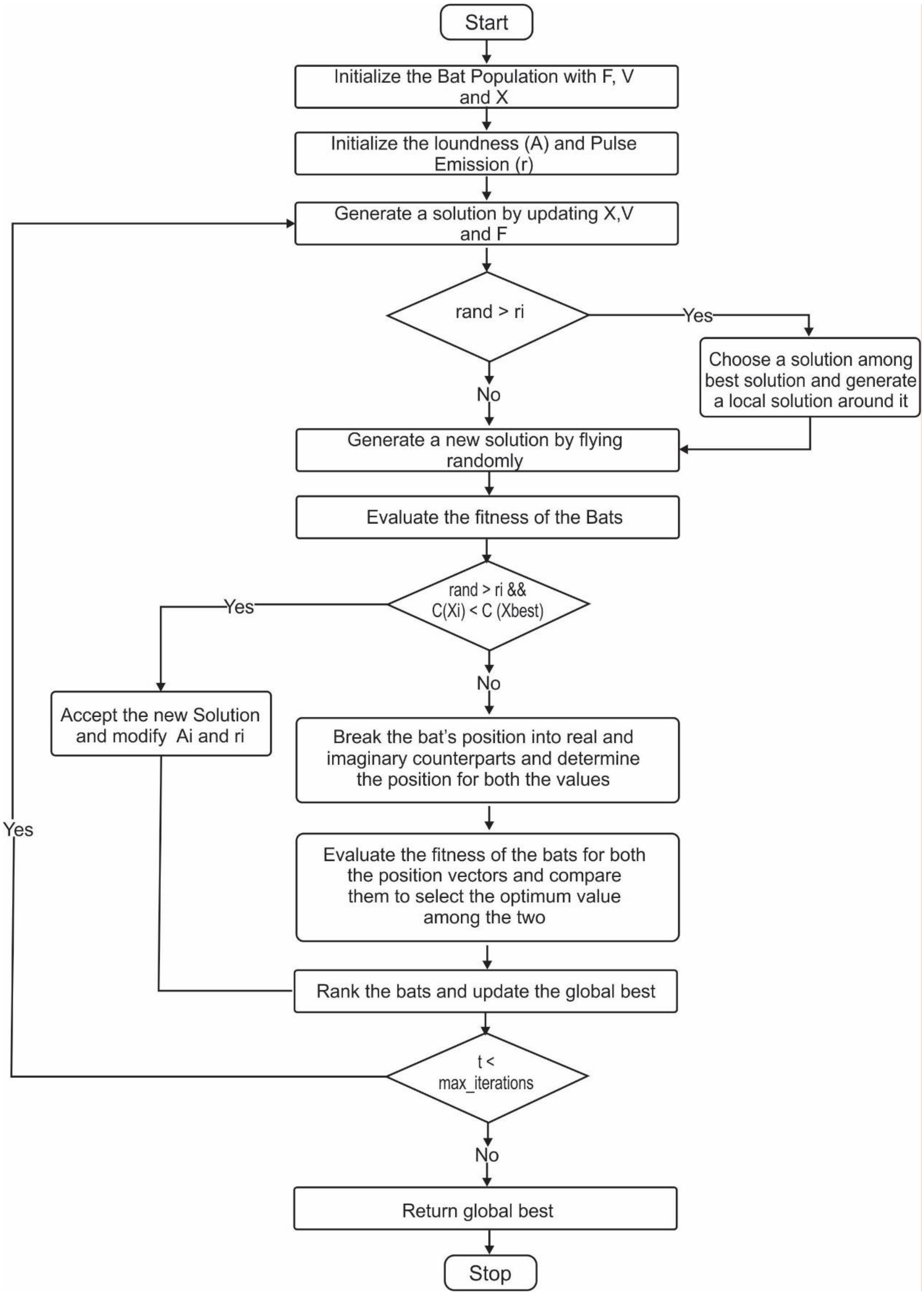


Fig. 4.4: Quantum-inspired Binary Bat Algorithm

The LISC dataset consists of 400 peripheral blood samples obtained from 8 healthy individuals. The blood samples were stained and smeared using the Gismo-Right technique, and were observed under a light microscope with an achromatic lens having a magnification power of 100. The images were recorded using a Sony SSCDC50AP digital camera. The images had a pixel density of 720×576 and were stored in the BMP format. A hematologist was then brought in for the classification of leukocytes into, basophil, eosinophil, lymphocyte, monocyte, and neutrophil.

The model uses the LISC data-set with 235 images of blood smears for classification of Leukocytes. The Fig. 4.5 depicts the data-sample with one microscopic image for each class of leukocytes.

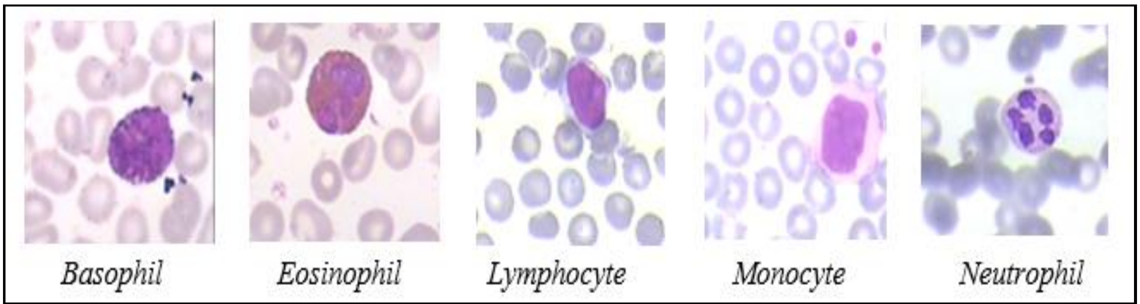


Fig. 4.5: LISC Dataset sample images

4.4.2 Feature Extraction

A classification algorithm cannot process images directly; hence, features are extracted from the above-mentioned data-set to allow classification of leukocytes. Herbert Ramoser et al. [227] extract features by segmentation of the LISC data-set using Pairwise SVM classification, with an accuracy of 95%. Tobias Bergen et al. [228], Erik Cuevas et al. [229], and Congcong Zhang [230] also propose various methods of feature extraction from blood smears for identification and classification of leukocytes, highlighting texture based [228], color based [229] as well as circle detection techniques [230] for the same.

The above approaches were used for the detection of leukocytes in the blood smear and a data-set with 35 features was realized, which was then trained as per the proposed model for the classification of leukocytes using the Binary Bat Algorithm.

The "outcome" of the derived data-set is set 1 for Basophils, 2 for Eosinophil, 3 for Lymphocytes, 4 for Monocytes and 5 for Neutrophils. The following Fig. 4.6, depicts a scatter plot displaying the Principal Component Analysis (PCA) for the spacial distribution of the feature extracted dataset.

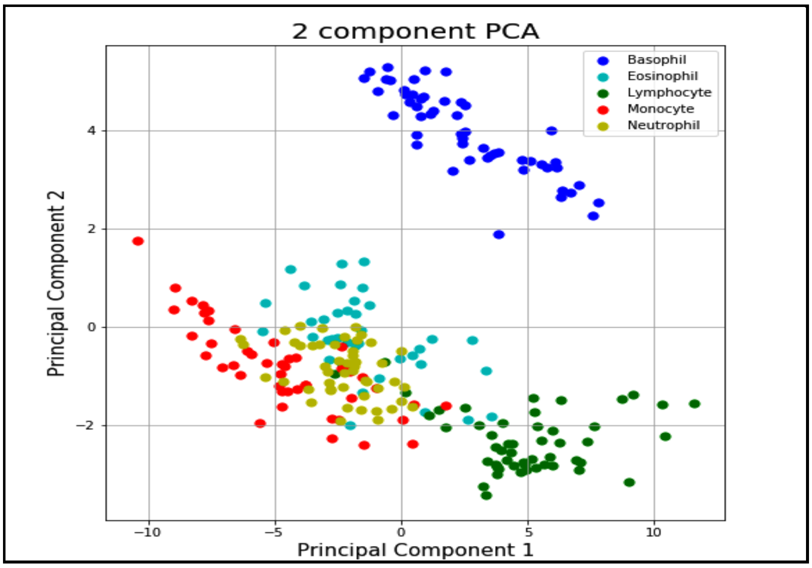


Fig. 4.6: PCA scatter plot of the dataset

The following Table 4.1 highlights the characteristics of the feature extracted data-set.

Table 4.1: Characteristics of LISC Dataset

Problem Description	Classification
Data-set Characteristics	Multivariate
Extracted Features	35
Feature Characteristics	Real
Total Data-points	237
Missing Values	0

4.4.3 Implementation of the proposed model

The proposed model uses the Quantum-inspired Binary Bat Algorithm for feature selection and tests the performance of the selected features over four classifiers, Logistic Regression, Decision Tree, KNN, and Random Forest. The system follows the following steps for each of these classification models

Division of the data-set into training and test sets. Implementation of QBBA for feature extraction using the evaluation function of the classification model as the Cost Function for the QBBA. Estimating the class of data-points in the test set using the selected features and comparing the results to obtain the accuracy of the system. The following Fig. 4.7 highlights the implementation approach.

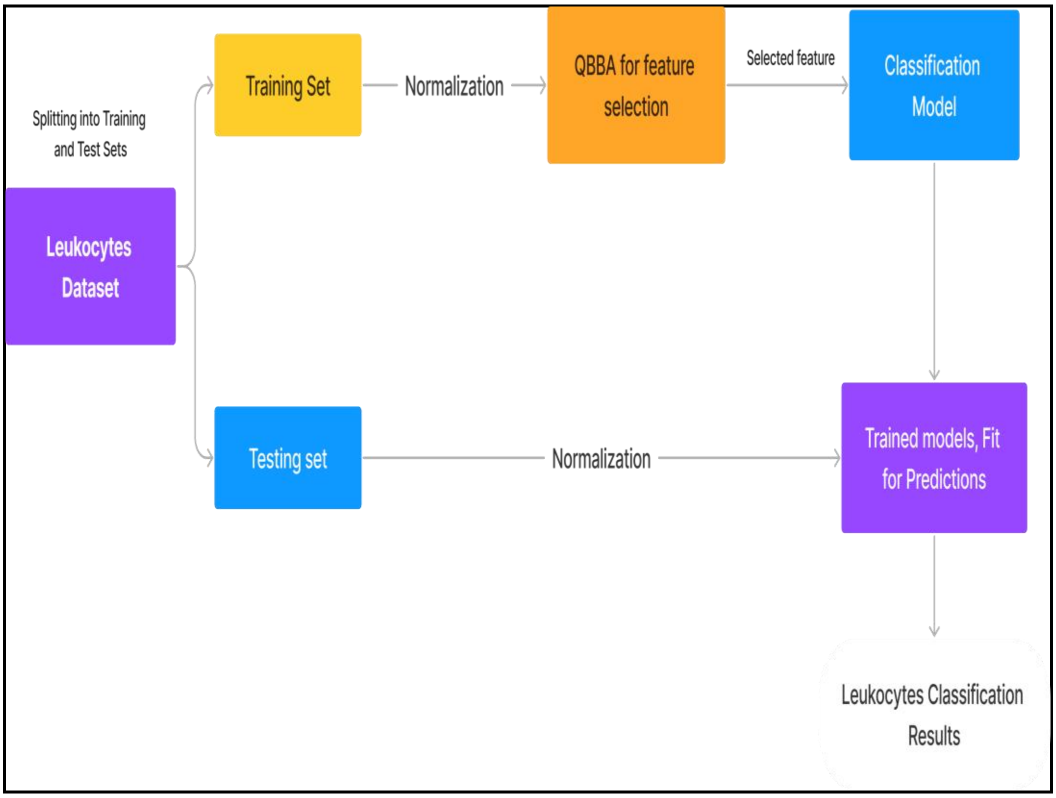


Fig. 4.7: The Implementation strategy for QBBA based classification

The following Algorithm explains the process of classification and optimization in detail:

Algorithm 4: Implementation of Quantum-Inspired Binary Bat Algorithm

1. Set the number of bats in the Bat Population as well as the number of iterations
2. For each division ratio 80:20, 75:25, 70:30, and 65:35 repeat steps 3 to 9
3. Split the data-set into training and test sets
4. For each Classification Strategy Logistic Regression, Decision Tree, KNN, and Random Forest repeat the steps 5 to 7

5. Calculate the accuracy of the model for the complete set of features, as the ratio of number of accurate predictions to the total number of predictions made by the chosen algorithm.
6. Select an optimum subset of features using QBBA and record the accuracy at the end of each iteration
7. Compare the results for each Classification Strategy
8. End For
9. Record the results
10. End For
11. Compare the results for different division ratios.

4.5 Results & Discussions

This section discusses the results obtained by the QBBA algorithm and compares the obtained results against its tradition counterpart. The results are obtained by the use of four classifiers for four different splitting ratios, and the comparisons are made thereof, on the basis of accuracy and the number of features selected by each of the classifiers.

A total of 35 features were extracted from the microscopic images of blood smear. The fitness of QBBA was evaluated on each of the following classifiers separately for a population of 35 bats over 20 iterations. The QBBA, and BBA algorithms were run on the above mentioned four classifiers for four different division ratios, 80:20, 75:25, 70:30, and 65:35. It was observed that the QBBA performed better in all of those situations by either reducing the features to yield the same level of accuracy, or by increasing the accuracy substantially. The following Table 4.2 illustrates the observed results in the following scenarios.

Table 4.2: Accuracy Comparisons of QBBA for different fitness functions

Fitness Function	Decision Tree	KNN	Random Forest	Logistic Regression
Accuracy (35 Features)	87.5%	93.33%	93.33%	97.91%
Accuracy (BBA)	97.91%	95.8%	98.61%	97.91%
Accuracy (QBBA)	97.91%	97.91%	98.61%	98.79%

The following Table 4.3 highlights the number of features selected by both Quantum-Inspired Binary Bat Algorithm and Binary Bat Algorithm for the above-mentioned fitness functions.

Table 4.3: Features selected by QBBA for different fitness functions

Fitness Function	Decision Tree	KNN	Random Forest	Logistic Regression
Features (BBA)	11	15	16	16
Features (QBBA)	8	6	12	14

The following figures, Fig. 4. 8 and Fig. 4.9 highlight the same data in a pictorial format.

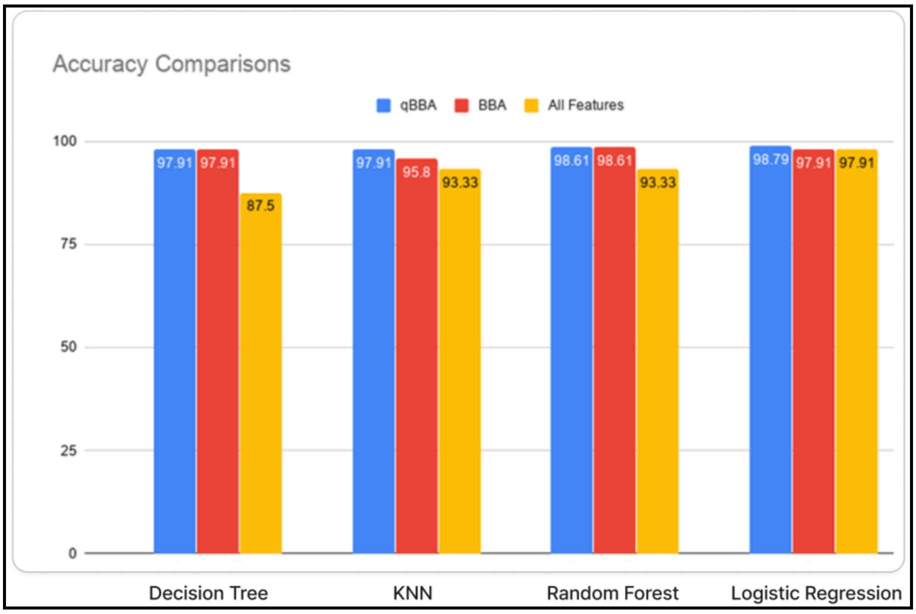


Fig. 4. 8: Accuracy Comparisons of QBBA, BBA and All Feature Selections over various fitness functions

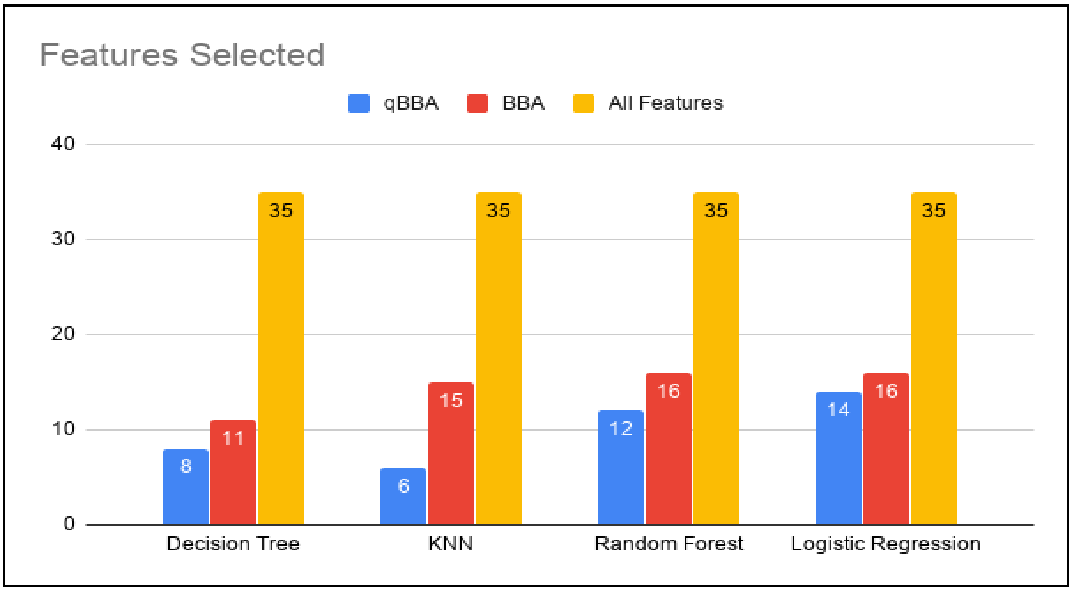


Fig. 4.9: Features Selected by qBBA, and BBA for various fitness functions

It can be easily observed that both QBBA and BBA perform better than an implementation without feature selection. The accuracy of both QBBA and BBA are comparable in the cases of Decision Trees and Random Forests, whereas QBBA performs slightly better in the case of KNN and Logistic Regression.

It can also be observed that QBBA selects a smaller subset of features as compared to its traditional counterpart. It is safe to conclude that QBBA delivers a comparable, and in some cases even better, accuracy than BBA for a smaller subset of features, making it a more powerful and cost-efficient algorithm. This is achievable as the population of a QBBA is as efficient as twice the population of BBA due to the application of the quantum approach. The performance of the QBBA algorithm was examined over a range of different bat population. The time consumed by each population size was then recorded. The following images represent the time consumed by the bat algorithm with respect to its population size.

It can be observed that the time function of the QBBA algorithm is less than O(n), where n is the population of the group of micro-bats. The Bat Algorithm depicts the same complexity but there is a slight increase in the per unit time for the QBBA as it requires parallel computing to deliver the reduced time results as expected by a Quantum Inspired Algorithm depicted in Fig. 4.10 and Fig. 4.11. In an ideal system, i.e., a parallel computing enabled system, the per unit time required by a QBBA will always be better than its conventional counterpart.

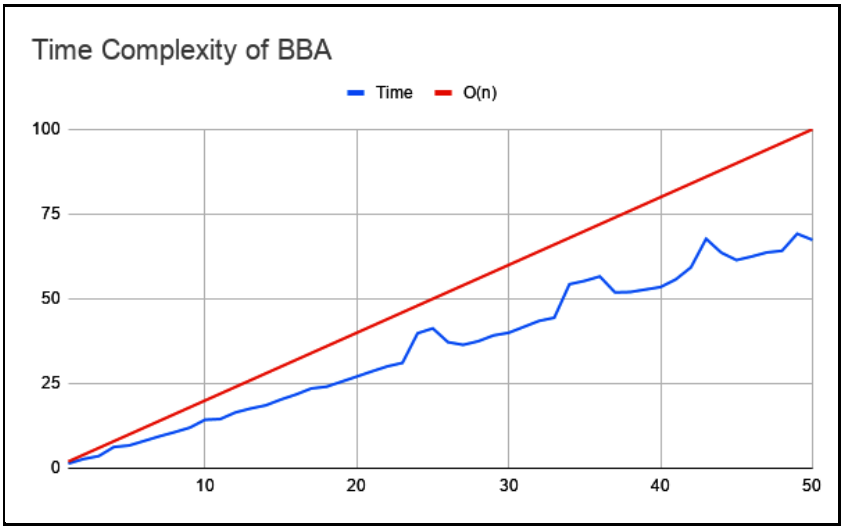


Fig. 4.10: Time complexity of BBA

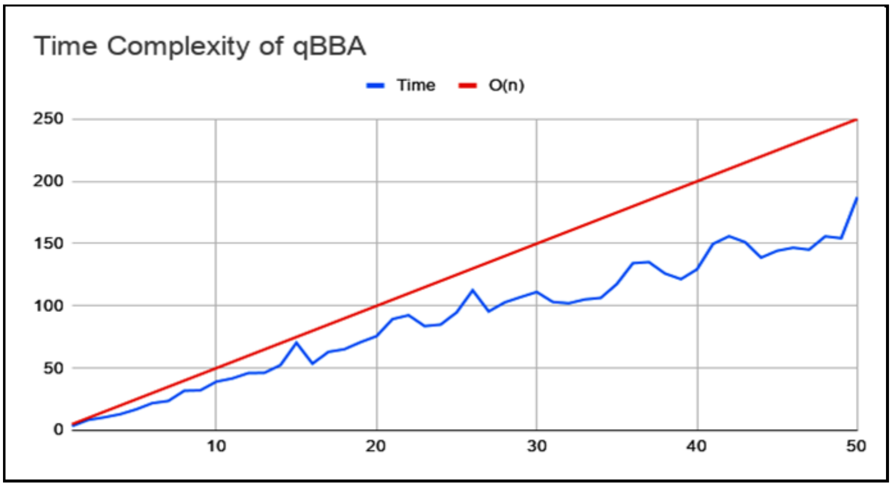


Fig. 4.11: Time Complexity of qBBA

4.6 Conclusion & Future Scope

The research presents a novel approach of integration of quantum principles and evolutionary algorithm i.e., Quantum-Inspired Binary Bat Algorithm (QBBA) This approach is applied to optimization problems which are of 0-1 type. The proposed algorithm is used to select the suitable subset of features from the extracted features for the classification of leukocytes which improves the efficiency and greatly reduces the computational costs. In comparison to traditional counterpart who was able to select minimum of 11 features QBBA outperforms as it selected minimum of 6 features. In terms of average accuracy, QBBA exhibited an average accuracy of 98.31% which depicts an evident improvement from 97.59% as displayed by BBA and 93.02% as displayed by the classification algorithm pre-feature selection. The algorithm proposed is effective in terms of feature selection as well as it tends to enhance its traditional counter-parts performance too. The proposed Quantum-Inspired Binary Bat Algorithm can be applied to the optimization of 0-1 problems and various other problems for feature selection.

Further studies with QBBA at its mainstay for leukocytes classifications can be carried out to open out the scope of the research to advance the diagnosis of haematological diseases. Various other nature inspired algorithms can be combined with proposed technique in pursuit of a more reliable and accurate system. QBBA can also be applied to various other classification or optimization problems.

CHAPTER 5

ENHANCED BINARY BAT ALGORITHM (EBBA) FOR FETAL HEALTH MONITORING

This chapter presents an Enhanced Binary Bat algorithm, the altered form of Binary Bat Algorithm for the multi-classification problem of Cardiotocography. Subset of optimal and relevant features is selected using the optimized and Enhanced BBA algorithm from cardiotocography dataset. Features selected by various evolutionary algorithms EBBA, qGWO, Genetic Algorithm; EBBA efficiently selects most reduced set of features. The proposed EBBA can be used in feature selection and classification of cardiotocography dataset under different fetus state i.e. normal, suspect and pathologic.

5.1 Introduction

These days' uncertainty and prevalent conditions of pandemic has engulfed and affected the world to an immense extent. Health has become top priority worldwide. Pregnant ladies were considered to be at higher risk to this pandemic situation. Otherwise also pregnancy may be complicated due to various medical problems associated with mother's health that could affect fetal well-being in turn. Risk to the fetus tends to increase with mother facing medical conditions like hypertension, preeclampsia, maternal diabetes and thyroid disease. There are many other circumstances that escalate level of risk to fetus during pregnancy such as vaginal bleeding, intra uterine growth restriction, extended pregnancy, placenta insufficiency, lessened fetal movements, multiples pregnancy etc. These complicated condition leads to the need of an additional means to gauge the fetal well-being. Cardiotocography is a process in which fetal heart rate and uterine contractions are monitored incessantly during pregnancy by employing ultrasound transducer upon the abdomen of the mother [207].

The major finding of this chapter has been accepted and published in Journal of Computers and Electrical Engineering, Volume 101, July 2022 Elsevier (SCIE Indexed IF: 4.152).

At times, the process of cardiotocography is well known as Electronic Fetal Monitoring (EFM). The purpose is to figure out the babies to whom supply of oxygen may possibly fall short; such a condition suffered by a fetus is called fetal hypoxia, which can have severe and prolonged effect as there is a risk associated with the baby of still birth, born with disabilities.

It is an indicative tool that is prevalently used to ascertain the fetal distress in the course of antepartum and intra-partum phase. With the help of readings and evaluations, one can get hold of Anabolic Acidosis and Hypoxic injury [231]. Cardiotocograph or more generally known as Electronic Fetal Monitor (EFM) is a machine used to keep record of heartbeat of fetus and the uterine contractions at the time of pregnancy. This process is popularly and formally known as Cardiotocography (CTG). The cardiotocographic data aids in classification of pathological state of the fetus as per normal indicating healthy condition, suspect with appropriate action required determining the extent of fetal distress or pathologic with speedy action required as fetal distress can affect immensely. It helps the obstetrician to foresee problems associated and obstruct the harm that can be caused to the fetus. During delivery, Hypoxic fetus is vulnerable to temporary impairment or death. Unsuitable treatments and incorrect diagnosis of Fetal Heart Rate pattern recordings results in more than half of the caused fatalities [232] [233] [195]. There is reduction in number of neonatal seizures with incessant cardiotocography during labor [208].

In terms of reliability External FHR outpaces Internal FHR as it also takes other parameters into account like recording maternal heart rate, loss of signal etc. likely during second stage of labour [234]. Before the CTG recording starts, it is advised to check the maternal heart rate separately in order to avoid the confusion of maternal heart rate and fetus heart rate [235]. During labour if there is a doubt of recording maternal heart rate instead of fetus, immediate internal monitoring is required with the help of fetal scalp electrode (FSE), if required. Obstetricians can make use of supplementary decision support system based on machine learning during Cardiotocography trace interpretation as an impartial gauge tool to enhance predictive competence and greatly reduces negative outcomes [209]. Generally, growth restricted or a preterm fetus has low reserves and compensatory mechanisms in comparison to full term grown fetus [236].

This research work proposes an optimized form of customary Binary Bat algorithm namely Enhanced Binary Bat Algorithm (EBBA). The algorithm proficiently combines the feature selection with the echolocation principle observed by bats for the cardiotocograph dataset. Further the algorithm makes use of several machine learning classifiers to report the accuracies of various nature inspired algorithms. The EBBA algorithm has been applied on the CTG (Cardiotocography) dataset is accessible at UCI Machine Learning Repository (https://archive.ics.uci. edu/ml/ machine-learning-databases/00193/). EBBA efficiently reduces the number of features i.e., selects the most distinguishing features with an average accuracy of 96.21%. Hence, it proves to be effective in the classification of fetal state under normal, suspect and pathologic label.

5.2 Background

5.2.1 Bat Algorithm

This algorithm is inspired by captivating group of mammals called bats. They possess the peculiar ability of echolocation as shown in Fig. 5.1 which they extensively use to detect an obstacle in their way during movement or track prey/food because of which they have drawn the attention of researchers from numerous fields. The phenomenon used by them is akin to sonar. Micro bats play a fundamental part in developing the significant attributes of this algorithm. They emit short pulse of sound which is also loud in nature, an echo returns as this wave knocks an object after a slice of time [33]. This in turn helps the bats to learn their farness from the object. This loud ultrasound generated tends to fluctuate around a fixed frequency which helps them to catch their prey and escape the obstacle as they are able to discriminate amongst them. A novel and interesting metaheuristics optimization technique has been developed by Yang [37] on the basis of behaviour of group of bats approaching obstacles, food/prey.

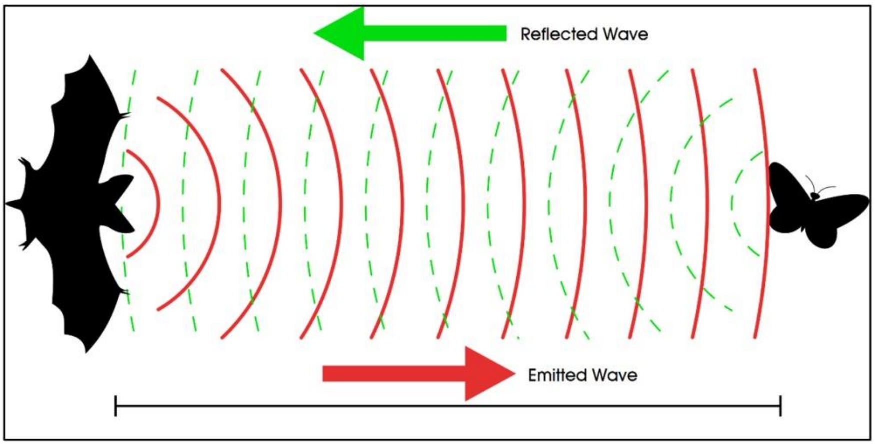


Fig. 5.1: Echolocation in Micro Bats

For each bat b_i, its initial position x_i, velocity v_i and frequency f_i are initialized with varying wavelength and loudness i.e., λ *and* A_0 to search for prey. T depicts the maximum number of iterations whereas t depicts each time step. As they approach the target, they tend to emit pulse whose wavelength (or frequency) is habitually adjusted, the rate of pulse emission ranges between values of 0 and 1 which greatly depends on its closeness to the object. The value of loudness varies from A_0 to A_{min} i.e. from a large positive value to a minimum constant value. Equations (5.1), (5.2) and (5.3) mentioned below keep record of velocity and position changes by the travelling of virtual bats.

$$f_i = f_{\min} + (f_{\min} - f_{\max}) \tag{5.1}$$

$$v_i^j\ (t) = v_i^j\ (t-1) + [\,\hat{x}^{\,j} - x_j^i\ (t-1)]f_i \tag{5.2}$$

$$x^j{}_i\,(t) = x^j{}_i\,(t-1) + v^j{}_i\,(t) \tag{5.3}$$

Here,

β-> randomly generated number $\beta \in [0, 1]$.

$x^j{}_i$ -> value of decision variable j for bat i at time step t.

Bat movements speed and reach is regulated by the value of f_i obtained by equation 1. The value of $\hat{x}^j$ variable depicts global best location solution for the decision variable j at that moment by comparing all the solutions of m bats available. Yang has recommended taking up random walks to get better variability of the likely solutions. Selection of one appropriate solution is made out of the current best solutions available and then to obtain a new solution for each and every bat that satisfies line 8 of algorithm 1 random walk is applied as given in equation (5.4).

$$x_{new} = x_{old} + \epsilon\bar{A}(t) \tag{5.4}$$

(t) stands for the average value of loudness of all the bats at time t, ϵ *varies in the range of [*$-1, 1$] attempts to the direction and strength of the random walk. The values are brought up-to-date for loudness A_i *and* emission pulse rate r_i *for* each and every iteration as mentioned below in equation (5.5) and equation (5.6):

$$(t + 1) = (t) \tag{5.5}$$

$$(t + 1) = (0)[1 - exp(-\gamma t)] \tag{5.6}$$

where α, γ are adhoc constants. Initially in the algorithm the values for emission rate (0) and the loudness (0) are chosen randomly most of the time. Generally, (0) $\in$ [1, 2] and (0) $\in$ [0, 1] [12].

The following Algorithm 5 depicts the aforesaid Bat algorithm in detail.

Algorithm 5: Bat Algorithm

---

1. Initially set the Bat population as b_i, where [i= 1 to m]
2. For each bat b_i there is a position x_i, velocity v_i and pulse frequency f_i where [i= 1 to m]
3. Initialize pulse rates as pulse_rate r_i, the loudness as A_i
4. Initialize each iteration as iter in T where T is maximum number of iterations
5. While iter < T, Repeat steps 6–18
6. For bat b_i, Repeat steps 7–15
7. Generate new solutions using Equations. (5.1), (5.2) and (5.3)
8. If (rand > r_i)
9. Choose a solution from the best solutions

10. Produce a local solution around the best solution
11. End if
12. If (rand < A_i) & (f (x_i) < f (CurrentBest))
13. Increase pulse rate r_i and reduce loudness of bat A_i
14. End if
15. End for
16. Rank the bats and find CurrentBest
17. End if
18. End while

5.2.2 Feature Selection & Binary Bat Algorithm

Bat algorithm application to continuous valued problems was exhibited by Yang [21]. In this the search space contained continuous valued positions that a bat can acquire. However, for combinatorial and discrete problems Nakamura et al. came up with the new variant of bat algorithm called Binary Bat Algorithm. On the other hand, for feature selection the search space is exhibited in the form of Boolean lattice of n dimension where the movement of bat tends to happen across the corners and nodes of the lattice. Since the problem addresses whether a feature is selected represented by value 1 or not by value 0. Equation (5.7) depicts bat's position which is controlled by a sigmoid function and the position of the bat is restricted using Equation (5.8). The sigmoid function is used to restrict the bats position to only 0 and 1 i.e., binary values in the binary version of the bat algorithm [25].

$$S(V_i) = \frac{1}{1+e^{-V_i}} \tag{5.7}$$

$$x_i^j = \begin{cases} 1 \text{ if } S\left(v_j^i\right) > \sigma \\ 0 \text{ Otherwise} \end{cases} \tag{5.8}$$

Here, $\sigma \sim (0, 1)$. Hence equation (5.8) makes 0, 1 i.e., binary values available to the Boolean lattice for bat's coordinates which signifies whether the feature is selected or not. Equation (5.3) can be replaced by equation (5.8).

5.3 Methodology

5.3.1 Enhanced Binary Bat Algorithm for feature selection and classification

The EBBA algorithm is modified version of Binary Bat Algorithm. It accepts all the attributes of the given IoT dataset and returns back the optimal features. It has been applied on Cardiotocography dataset. It is a multi-classification problem in which the diagnosis is given by an obstetrician about the fetal state i.e., Normal, suspect and pathologic. The EBBA was used with three classification machine learning models KNN, DecisionTree, Random Forest. The following algorithm 6 explains each step of EBBA precisely.

Algorithm 6: Enhanced Binary Bat Algorithm

1: **Set** population size size_of_pop, pulse emission rate constant pulse_rate_constant number of iterations iter , loudness constant loud_constant , total features n , bat position pos_of_bat , bat velocity vel_of_bat , bat loudness loud_of_bat , pulse emission rate pulse_rate , fitness fitness , global fitness global_fitness

2: **for each model do :**

3: **test_train_split(ratio = 70:30) . Sets should have only those features where** pos_of_bat[i] = 1

4: Fitness function is defined as the accuracy of the machine learning model. Accuracy is updated to acc[i]

5: Set **random = [0,1]**

6: **if** ((loud_of_bat[i] >= random) and (acc >= fitness[i]) :

7: fitness[i] = acc[i]

8: **Update** pulse_rate[i] & loud_of_bat[i] using Equations

9: **end if**

10: **end for**

11: **Set** index_pos and max_fitness

12: **if** (max_fitness > global_fitness) :

13: **global_fitness = max_fitness**

```
global_pos = index_pos
end if
Attributes Selected = Numeber of 1's in global_pos
for each model do :
Beta, random , e > [0,1]
if ( random > r[i] ):
find pos_of_bat[i] using Equations
end if
if ( ( loud_of_bat[i] > random ) and (global_fitness > fitness[i] ) ) :
Update freq_of_bat[i] , vel_of_bat[i] , pos_of_bat[i] using Equations
Get new pos_of_bat[i] using Equations
end if
end for
```

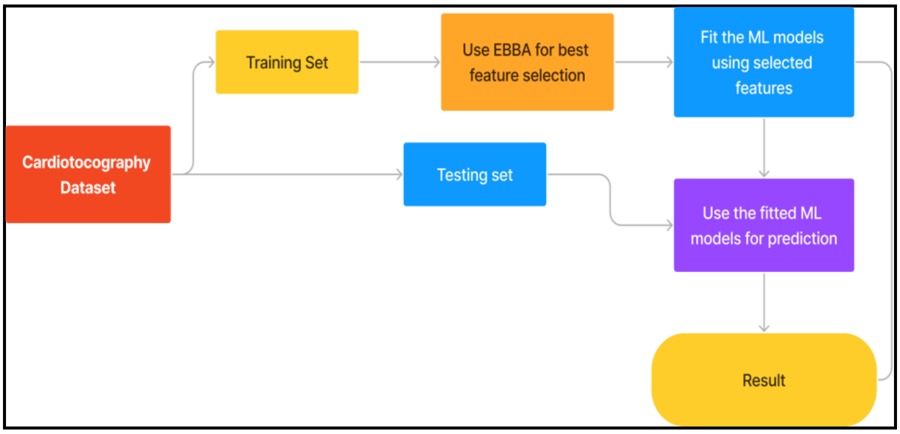


Fig. 5.2: Flow graph of EBBA

In Fig. 5.2 Cardiotocography dataset of 2126 fetal cardiotocograms (CTGs) features is divided into training set and testing set with the ratio of 70:30. EBBA is employed to the training set for the selection of optimal features. Machine classifiers are then applied to find out the accuracy of solutions obtained with the optimal features selected by EBBA. Result is the dataset classification with the determination of fetus state accurately. Fig. 5.3 comprehensively explains the working of Enhanced Binary Bat Algorithm (EBBA).

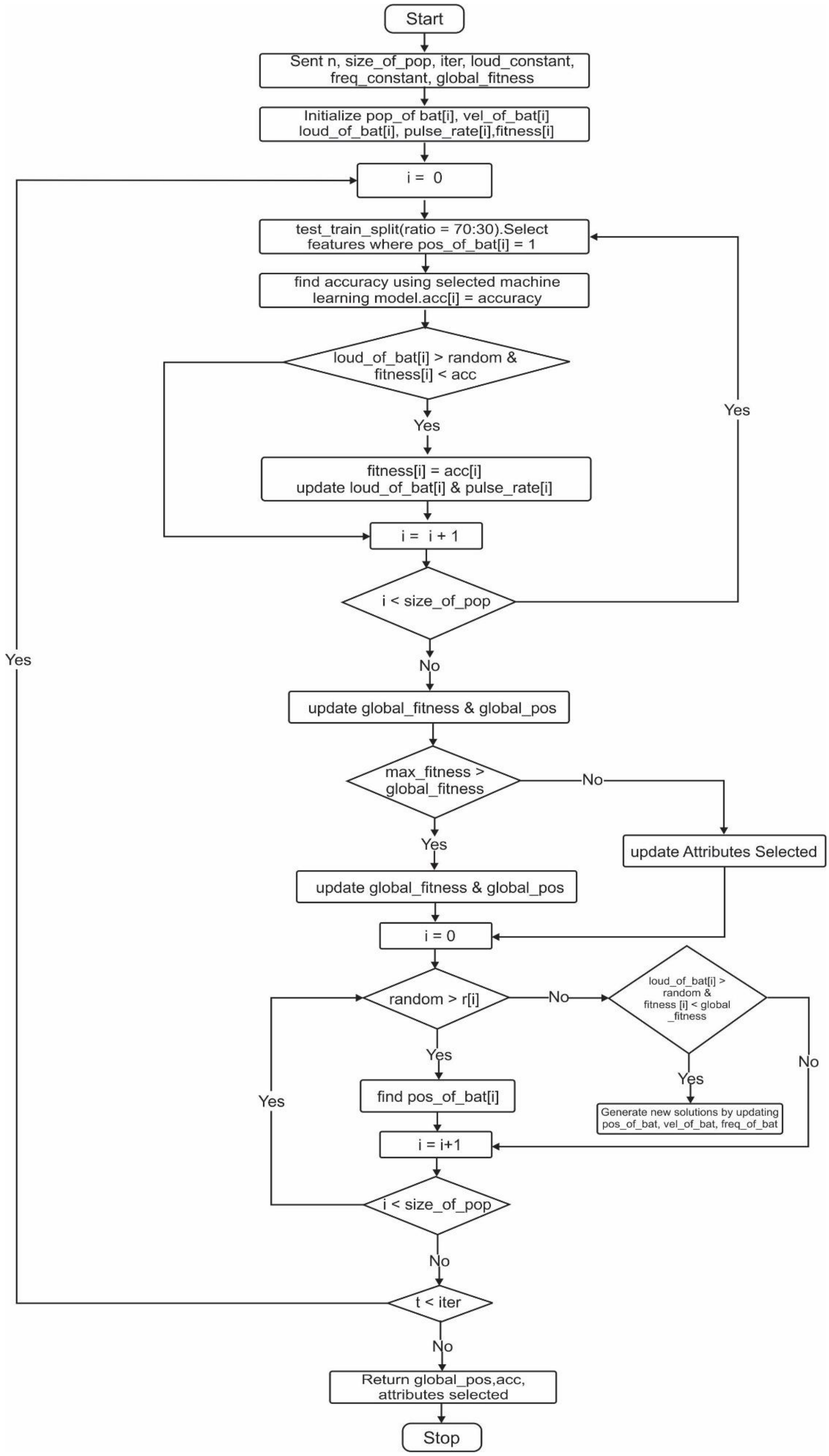


Fig. 5. 3: Flowchart of EBBA

5.3.2 Dataset

The dataset contains measurements of Fetal Heart Rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. The corresponding diagnostic features were evaluated by treating 2126 fetal cardiotocograms (CTGs). Three expert obstetricians classified CTGs with a classification label. Classification is primarily performed under two heads; one is morphologic pattern (A, B, C. ...) and the other is fetal state (N, S, P). The dataset is used for 3-class classification. Table 5.1 comprises of description of various Cardiotocograms attributes.

Table 5.1: Description of CTG Dataset attributes

Attribute	**Description**
LB	FHR baseline (beats per minute)
AC	No. of accelerations per second
FM	No. of fetal movements per second
UC	No. of uterine contractions per second
DL	No. of light decelerations per second
DS	No. of severe decelerations per second
DP	No. of prolonged decelerations per second
ASTV	Percentage of time with abnormal short-term variability
MSTV	mean value of short-term variability
Width	the width of FHR histogram
Min	minimum of FHR histogram
Max	Maximum of FHR histogram
Nmax	No. of histogram peaks
Nzeros	No. of histogram zeros
Mode	histogram mode
Median	histogram median
Variance	histogram variance
Tendency	histogram tendency
CLASS	FHR pattern class code (1 to 10)
NSP	fetal state class code (N = Normal; S = Suspect; P = Pathologic)

Each cardiogram is represented in the dataset by 21 features out of which 11 are obtained using electronic fetal monitoring sensors and the rest 10 are extracted using those 11 features. Table 5.2 exhibits various parameters of problem type and their values in cardiotocography dataset in terms of number of attributes, sample, features and also types of features in using multi-classification problem.

Table 5.2: Characteristics of Cardiotocography Dataset

Problem Type	Multi–Classification Problem
Types of features	Real
Total attributes	23
Total samples	2125
Total missing values	0

5.3.3 Implementation of EBBA

Implementation of EBBA has been performed. This section gives an insight to various implementation related information in terms of set up required to perform the experiment, parameters taken as inputs, various tuning parameters of machine learning algorithms.

5.3.3.1 Experimental Setup

The system on which all the testing was carried out has the following configuration:

- Windows 10 version 1903
- 2.30 GHz Intel Core i5 6th generation
- 8 GB 2401 MHz DDR4

Python 3.7 and it's libraries have been used.

5.3.3.2 Input Parameters

This section talks about various input parameters taken into consideration for EBBA. The input variables used in the algorithm are mentioned in Table 5.3.

Table 5.3: Initialized value and description of input parameters

Parameter	**Initialized Value**	**Description**
N	**21**	Total features
iter	**20**	Total iterations
size_of_pop	**30**	Size of population
pos_of_bat	**[0,1]**	Position of bats
vel_of_bat	**0**	Velocity of bats
loud_constant	**0.8**	Loudness constant
freq_constant	**[0,1]**	Frequency constant
pulse_rate_constant	**0.9**	Pulse rate constant
loud_of_bat	**[1,2]**	Loudness
pulse_rate	**[0,1]**	Pulse rate
fitness	**-1**	Fitness
global_fitness	**-1**	Global Fitness

5.3.3.3 Machine Learning Models

Different machine learning models (DecisionTree, K-Nearest Neighbours, Random Forest) are used for classification. Table 5.4 gives insight to various tuning parameters associated with machine learning classifiers.

Table 5.4: Parameters of ML classifiers

Model	**Tuning Parameters**
Decision Tree	max_depth = 16
KNN	p = 1, n_neighbors =1
Random Forest	n_estimators = 105

5.4 Results

In this section results of the EBBA algorithm on Cardiotocography dataset have been discussed. The number of optimal features selected using different evolutionary algorithms is shown in Table 5.5 which shows EBBA tends to select the reduced set of 11 features out of 21 whereas qGWO selects 15 and Genetic Algorithm selects 12 features. Table 5.6 exhibits accuracy in percentage obtained using different ML models (Decision Tree, KNN, Random Forest) when applied to all features of the dataset and optimal features selected by proposed EBBA. Comparative study has been done with the nature inspired algorithms qGWO and Genetic Algorithm in terms of accuracy. Accuracy of different algorithms on the basis of features selected by EBBA is presented in Table 5.7. There is an average increase of 2.09 % in accuracy when using EBBA. The proposed algorithm outperforms the two evolutionary algorithms in terms of accuracy and minimal feature subset obtained. Average number of features selected by different evolutionary algorithms EBBA, qGWO and GA is depicted in Fig. 5.4. Fig. 5.5 presents accuracy computed in terms of percentage using different ML classifiers on the set of all features and set of features selected by EBBA. Comparative analysis of accuracy of different evolutionary algorithms and machine learning classifiers is represented in Fig. 5.6.

Table 5.5: No. of features selected using different nature-inspired algorithms

Algorithm	Features selected (out of 21)
EBBA	11
qGWO	15
Genetic	12

Table 5.6: Accuracy of different ML classifiers on all features and features selected by EBBA

Model	Accuracy using all features (%)	Accuracy using selected features(%)
Decision Tree	92.63	94.36
KNN	92.78	94.67
Random Forest	93.57	96.21

Table 5.7: Comparison of accuracy obtained by ML classifiers on the basis of features selected by different Nature-Inspired Algorithms

Algorithm/ML Classifier	Decision Tree	KNN	Random Forest
EBBA	94.36	94.67	96.21
gGWO	93.41	92.47	94.82
Genetic	94.04	92.06	94.66

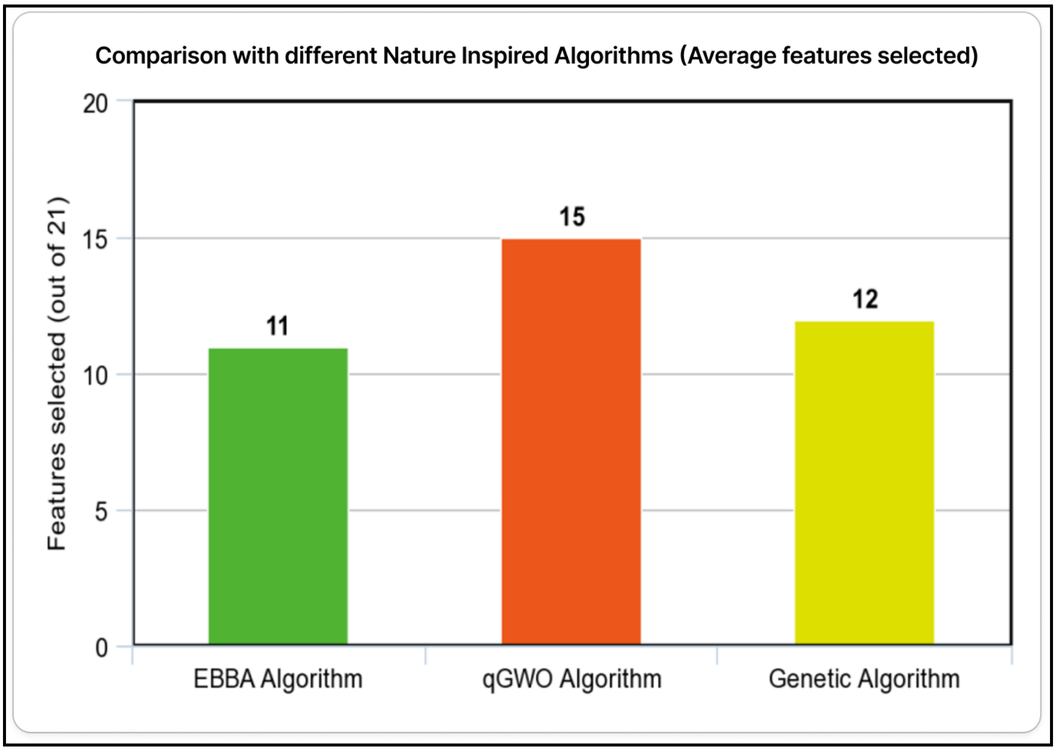


Fig. 5.4: Number of features selected using different Nature-Inspired Algorithms

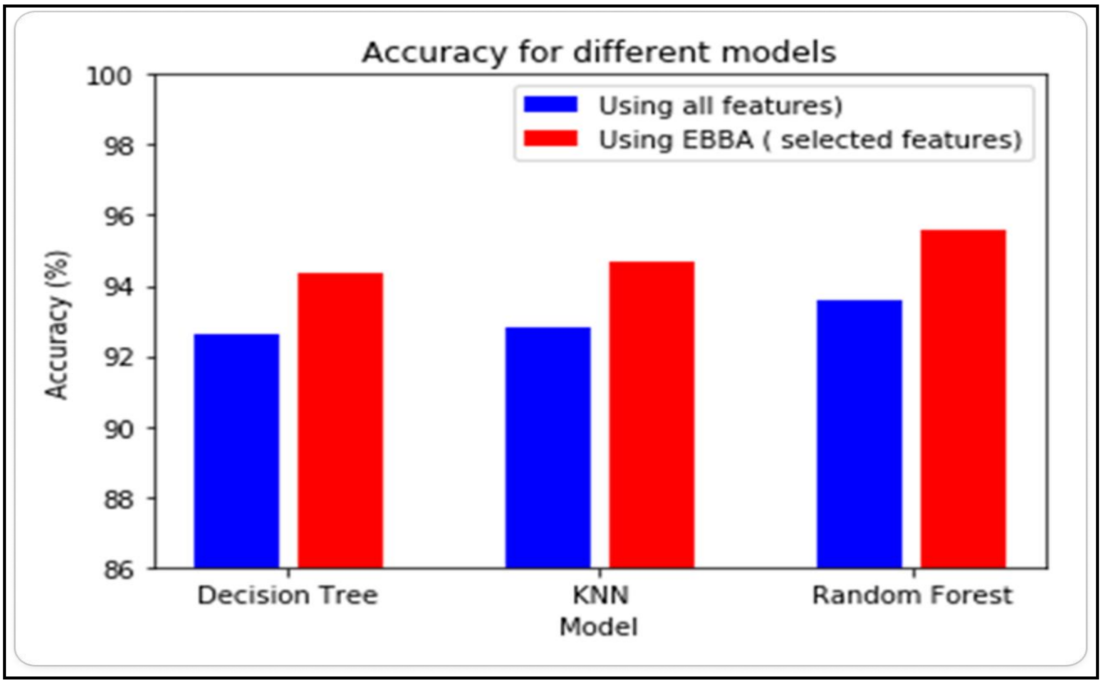


Fig. 5.5: Accuracy computed in percentage using different ML classifier using all features and the features selected using EBBA

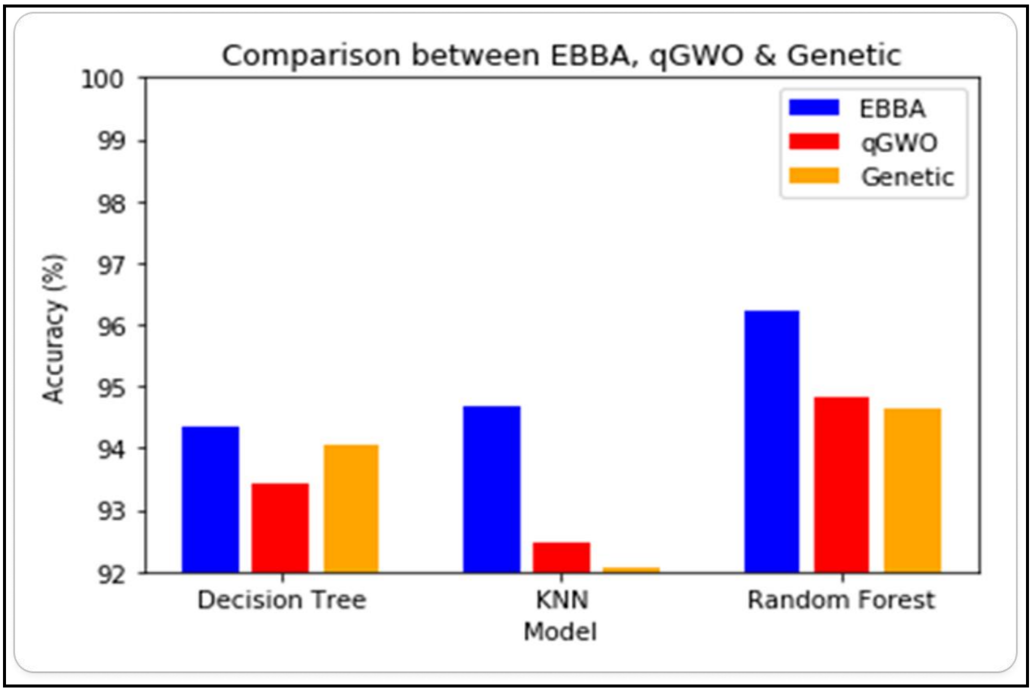


Fig. 5.6: Comparison between different nature-inspired algorithms & ML classifiers in terms of Accuracy

5.5 Conclusion & Future Scope

Enhanced Binary Bat Algorithm (EBBA) outperforms other metaheuristics for optimized feature selection. This novel bio-inspired optimization algorithm greatly reduces computational cost by performing dimensionality reduction with high accuracy for implementing feature selection. Section 3.1 discusses the suggested algorithm in detail. EBBA has made a selection of least possible 11 features, with the utmost average accuracy of 96.21% in comparison to average accuracies of 94.82% and 94.66% computed by customary algorithms qGWO and GA respectively. The results depict that this novel proposed algorithm outperforms qGWO and GA with respect to accuracy measure.

Further studies can be conducted in the discipline of fetal health monitoring and diagnosing of fetal distress could be done by combining various nature inspired algorithms and various machine learning models. The hybridization of various nature inspired algorithms can give results with improved accuracies. Other techniques of fetal distress determination can also be evaluated and examined with the proposed algorithm for timely and accurate detection. Other than the cardiotocography data set EBBA can moreover be useful to arena of optimization problems applications.

CHAPTER 6

EFFICIENT DIAGNOSIS OF THYROID DISEASE USING IMPROVISED MOTH FLAME OPTIMIZATION ALGORITHM

This chapter presents an intelligent bio-inspired feature selection algorithm named Improvised Moth Flame Optimization (IMFO) to select the most significant attributes required for early and accurate diagnosis of thyroid disease. The proposed IMFO is modeled as a filter-based feature selection method and is an improvised variant of Moth Flame Optimization algorithm. IMFO successfully selects the optimal subset of thyroid attributes to reduce the computationally expensive learning time. The selected attributes significantly enhance classification accuracy, improving the overall diagnostic performance.

6.1 Introduction

Thyroid disease is primarily affecting masses throughout the globe. According to recent studies and surveys on thyroid disease, it is estimated that about 82 million people across the globe are suffering from thyroid disease, of which 42 million people and around 20 million people belong to India and America, respectively. Recent statistics show that there has been a significant increase in thyroid cases of about 64% in females and 48% in males. Thyroid is the sixth most common disease in women and mainly occurs between 20-34. Even though the thyroid is seen mainly in women, both men and women die at similar rates once diagnosed. According to a recent study in America, around 2170 patients died this year, out of which 1020 were men and 1150 were women. Researchers suggest that this similar death ratio is due to a worse prognosis in men.

In our body, metabolism and protein synthesis are regulated by the thyroid hormone released by the endocrine gland. It is present below Adam's apple, which has its frontage on the neck, and its two lobes are connected with the help of an isthmus.

The major finding of this chapter has been accepted in Journal of Computer Modeling in Engineering & Sciences (SCIE Indexed IF: 2.027).

The thyroid gland produces several hormones, T3 (triiodothyronine) and T4 (thyroxine). Another particular hormone the thyroid gland produces is calcitonin, which performs calcium homeostasis, i.e., it regulates calcium ions. Thyroid Stimulating Hormone (TSH) signals the secretion of thyroid hormones by thyroid glands.

Thyroid Releasing Hormone (TRH) signals the secretion of the pituitary gland produced by a small organ in the brain, namely the hypothalamus. Lately, the active hormones T3 and T4 enter the bloodstream. Further, the thyroid hormone T4 tends to convert to T3 in the liver and other body parts. The presented exposition is mainly focused on 2 types of thyroid disorders: hypothyroidism and hyperthyroidism. Hypothyroidism is caused due to underactive thyroid gland. Several reasons cause hypothyroidism, including iodine deficiency, hypothalamic disease, or Hashimoto's thyroiditis (autoimmune disease).

On the other hand, hyperthyroidism is characterized by the overactive thyroid gland, i.e., excessive production of thyroid hormones. Graves' disease, an excessive amount of iodine in the body, etc., are the leading causes of hyperthyroidism. Both are severe thyroid disorders and can cause serious health problems like heart failure, lung diseases, weak and brittle bones, etc. Therefore, early and accurate thyroid disease diagnosis is the primary and foremost objective.

Before providing appropriate treatment to an affected person, the thyroid needs to be diagnosed as early as possible. Doctors have been using various traditional methods to diagnose thyroid in the last few years. Some of them include physical examination [237], Computed tomography scan [238], X-ray, Radionuclide scanning [239], and much more. There are several risk factors associated with these traditional methods. The work done in [202] clearly explains the risk of cancer due to radiation. According to the study presented in [203], x-ray exposure leads to neoplastic disease, due to which the cells of the body show abnormal growth, which may gradually lead to cancer.

Highlights of the research are as mentioned below:

- An Improved Moth Flame Optimization Algorithm (IMFO) for classifying thyroid states under normal, hyperthyroidism, and hypothyroidism have been discussed.
- Most discriminating and significant features are selected using Proposed IMFO
- The proposed algorithm has been implemented on the Thyroid dataset with an average accuracy of 99.61%.
- Different machine learning classifiers: (i) DecisionTree (ii) K-Nearest Neighbor (KNN) (iii) Random Forest (iv) SVM (v) Quadratic Discriminant Analysis (QDA) (vi) Gaussian Naive Bayes (GaussianNB) (vi) AdaBoost have been used to assess the optimality of features selected by IMFO.
- Comparative analysis in terms of accuracies and selected features of several evolutionary algorithms like Optimized Crow Search Algorithm (OCSA), Modified Grey Wolf Optimization Algorithm (MGWO), Optimized Cuttlefish Algorithm (OCFA), and Modified Ant-Lion Optimization Algorithm (MALO) has been discussed.

6.2 Background

6.2.1 Traditional Moth Flame Optimization Algorithm (MFO)

In the traditional MFO algorithm, the candidate solutions to the problem are the moths present in the search space. The moths fly in the 1-D, 2-D, 3-D, or hyper-dimensional space while changing their current position. Another prominent aspect of this algorithm is that both moths and flames are considered the optimal solution to the problem. The distinction between moths and flames can be observed by updating their position in search space. Flames are considered the best solution and act as flags or pins for the moths. Moths try to update their position around a particular flame in the search space and update their position in case of a better solution.

The above-explained mechanism is performed with the help of a logarithmic spiral, constrained by the following factors:

1. The initial point of the spiral must start from the position of the moth in the search space
2. The final point of the spiral must be the position of the flame around which the moth is flying.
3. The fluctuation of the range of the spiral should not exceed the search space.

The mathematical equation for updating the position is as follows:

$$S(M_i, F_i) = D_i e^{bt} cos(2\pi t) + F_j \tag{6.1}$$

where d_i represents the distance between the i^{th} moth and j^{th} flame. The constant b defines the shape of the logarithmic spiral, and t denotes a random number whose value lies in [-1,1].

D_i is calculated as follows:

$$D_i = |F_j - M_i| \tag{6.2}$$

where M_i and F_j represent the i^{th} moth and j^{th} flame in equation (6.2), respectively, and D_i denotes the distance between them.

From equation (6.1), it has been observed that the next position of a moth is updated concerning a flame in the search space. The variable 't' in the equation decides how close the moth will be to the flame's next position. The value of t lies in [-1,1], which indicates that at t = -1, the position is the closest, and at t=1, the position is the farthest, assuming a hyper ellipse in all directions. Therefore, with the help of equation (6.1), the movement of the moths is restricted on this elliptical path and not in between the search space, which guarantees both exploration and exploitation. The graphical representation of this mechanism is illustrated in fig. 6.1.

Flames are inferred to be the best solutions obtained so far to increase the possibility of finding better solutions, and their position is stored in the matrix F. The moths must update their positions concerning this matrix during optimization. Further improvement in exploitation is made by assuming the value of t in [r,1], where r is the convergence constant linearly decreased from -1 to -2 over the successive iteration. This method enables the moths to exploit their corresponding flame more precisely in the search space.

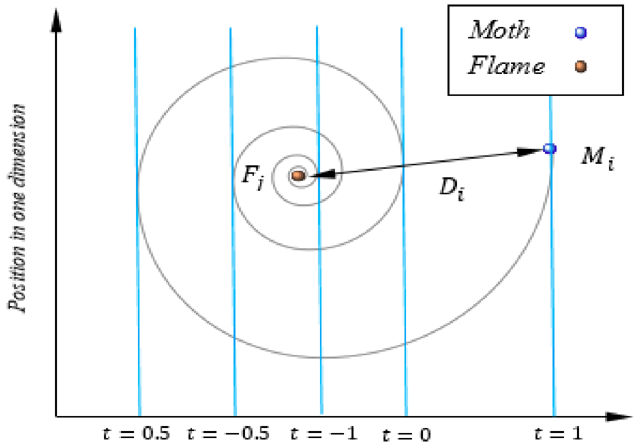


Fig. 6.1: Logarithmic spiral depicting the space around a flame

Since in MFO, the position updating requires a moth to fly towards a flame, there may be a possibility that the moths may get trapped in local optima. To avoid this situation, each moth is restricted to flying towards a particular flame from the list of all flames. After each successive iteration, the list of flames is sorted according to their fitness using a fitness evaluation function. The moths then updated their position concerning their flames in the updated list.

There may be a possibility that due to n number of flames in the search space, the exploitation by moths may degrade. To prevent this issue, the number of flames in the search space is decided by an adaptive decreasing mechanism which is represented as follows:

$$FlameNumber = round\left(N - \frac{l * N - 1}{T}\right) \tag{6.3}$$

where, l and T denotes the current and maximum number of iterations, respectively, and N represents the maximum number of flames in equation (6.3). Fig. 6.2 explains the linear decrement of the number of flames following T. The flowchart of traditional Moth Flame Optimization is shown in fig. 6.3.

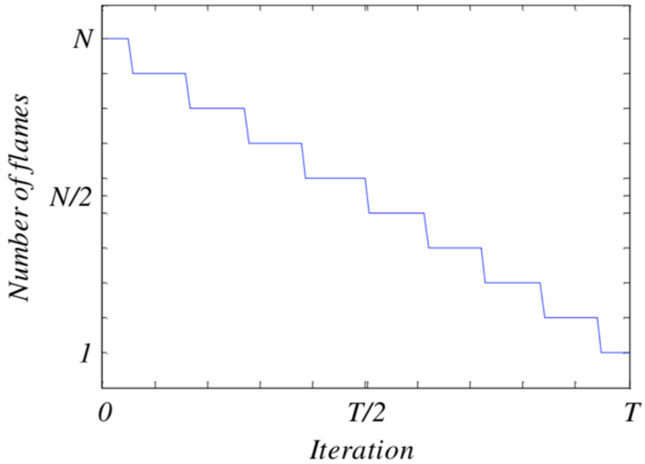


Fig. 6.2: Number of flames at each iteration

The algorithm of MFO is described below in Algorithm 7:

Algorithm 7: Moth Flame Optimization Algorithm

Initialize each moths position randomly

Define all the initial parameters

if iteration == 1

    calculate fitness for each moth;

    sort the moths;

else

    double the population and calculate fitness;

    sort population;

end

for i in range (1, n + 1)

    for j in range (1, dim + 1)

      calculate r and t

      Evaluate D using equation (6.2)

      Update each moth's position by using equation (6.1)

    end

end

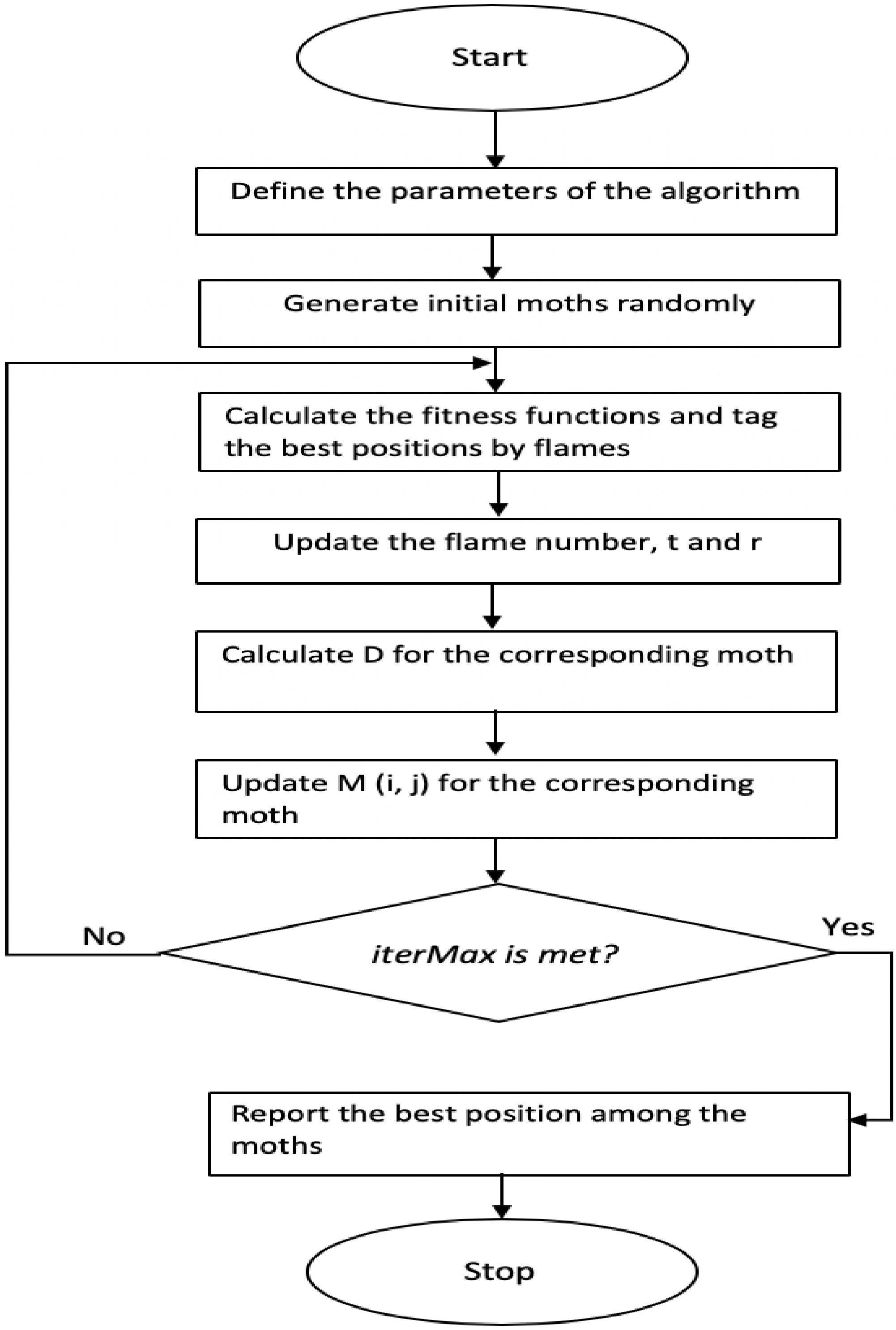


Fig. 6.3: Flowchart: Traditional Moth Flame Optimization (MFO) Algorithm

6.3 Proposed Methodology and Implementation

In this chapter an efficient model is proposed to diagnose thyroid diseases, as shown in fig. 6.4. The proposed system model consists of 3 main processing steps: data preprocessing to normalize the data so that all features are on a common scale, optimal feature subset selection using the proposed bio-inspired IMFO algorithm, and classification of possible diseases. All the experiments were conducted using a Lenovo Ideapad 720S having Intel Core i7 8th generation processor with 8 GB RAM and 512 GB SSD. All the algorithms were written in python language and were executed using Python version 3.6 and libraries: Scikit-learn, Pandas, Matplotlib, and NumPy.

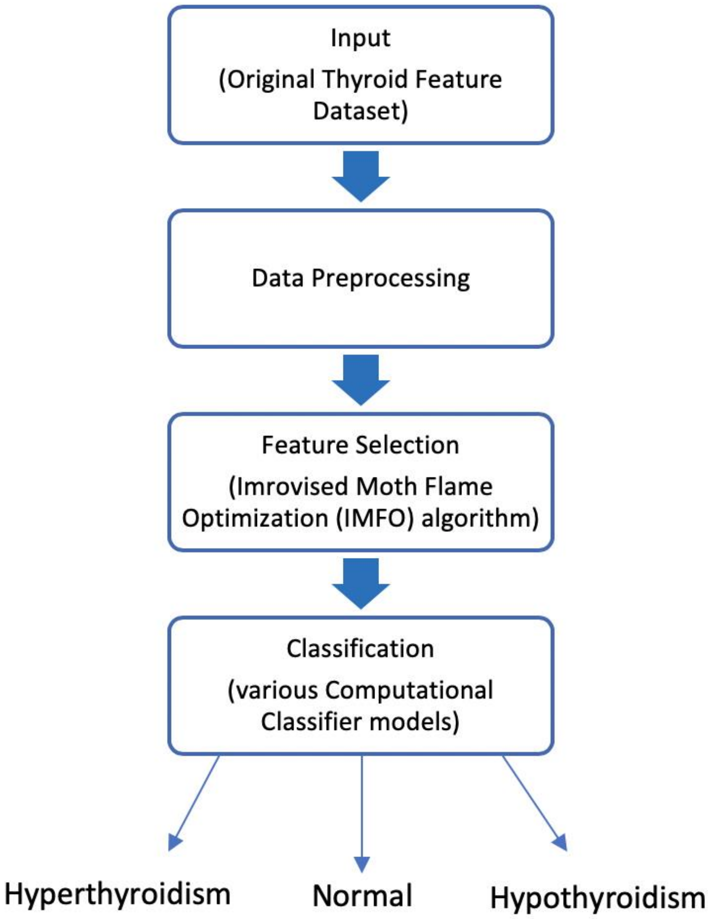


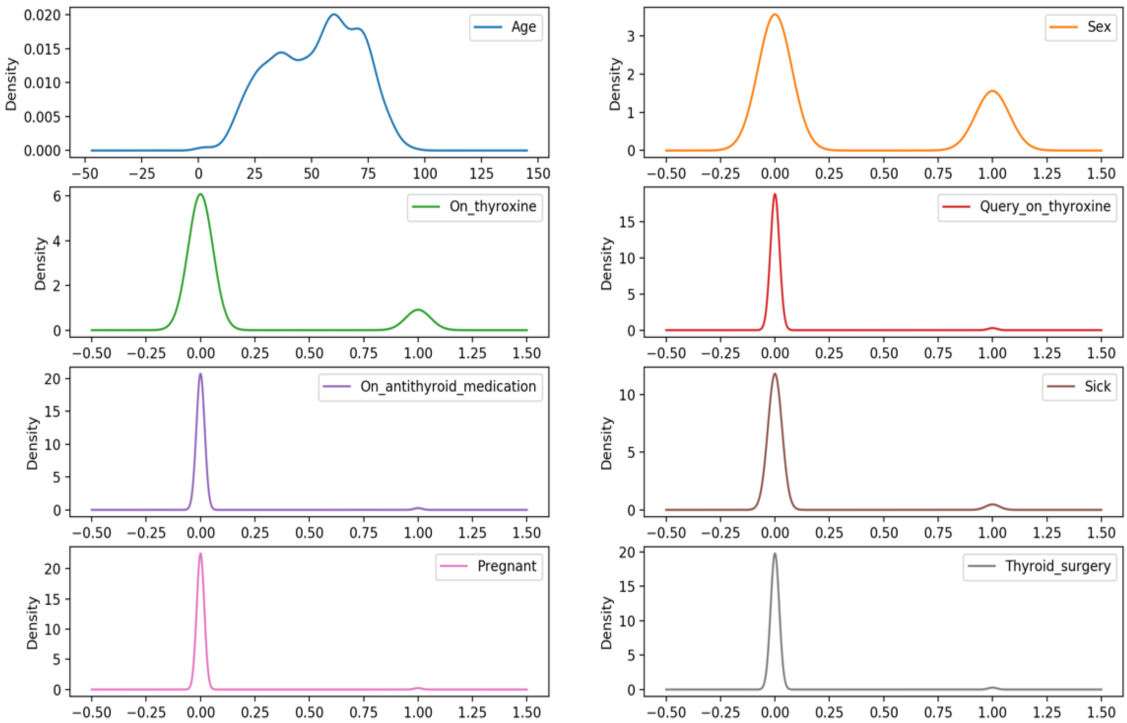
Fig. 6.4: Proposed system model using IMFO

6.3.1 Dataset

The Garavan Institute in Sydney, Australia, provided the dataset and is available on the University of California at Irvine (UCI) storehouse website. The dataset consists of 7200 patients, out of which 368 were suffering from hyperthyroidism, 6666 patients were diagnosed with hypothyroidism, and rest of the subjects have regular (healthy) conditions as described in table 6.1. In total, 21 attributes of each subject are described in the dataset. The density distribution of each attribute is shown in figure 6.5. The distribution of the dataset using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) is shown in fig. 6.6 and fig. 6.7.

Table 6.1: Thyroid dataset class description

Class	Instances
Normal	66
Hyperthyroidism	368
Hypothyroidism	6666



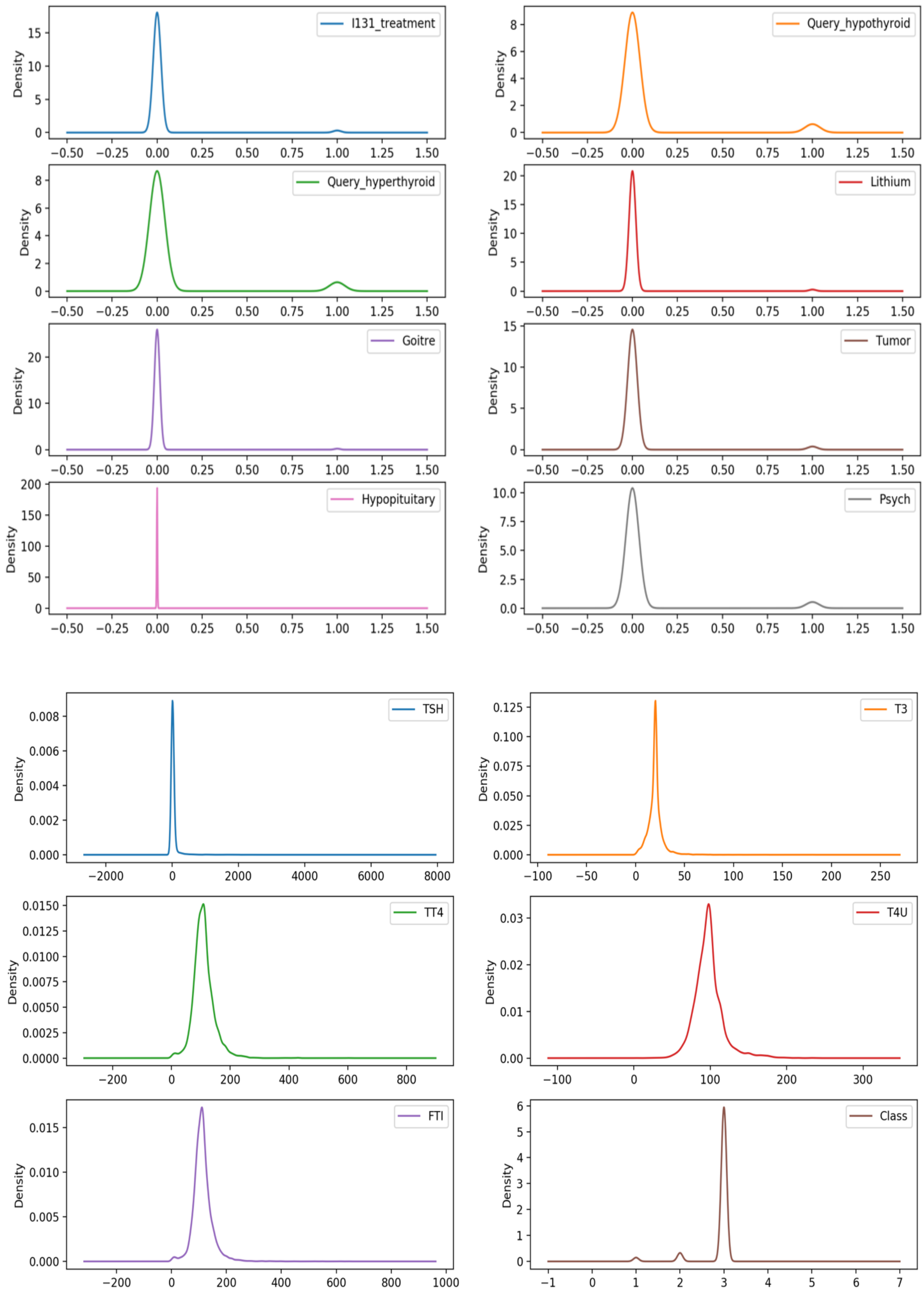


Fig. 6.5: Density distribution of each attribute of the thyroid dataset

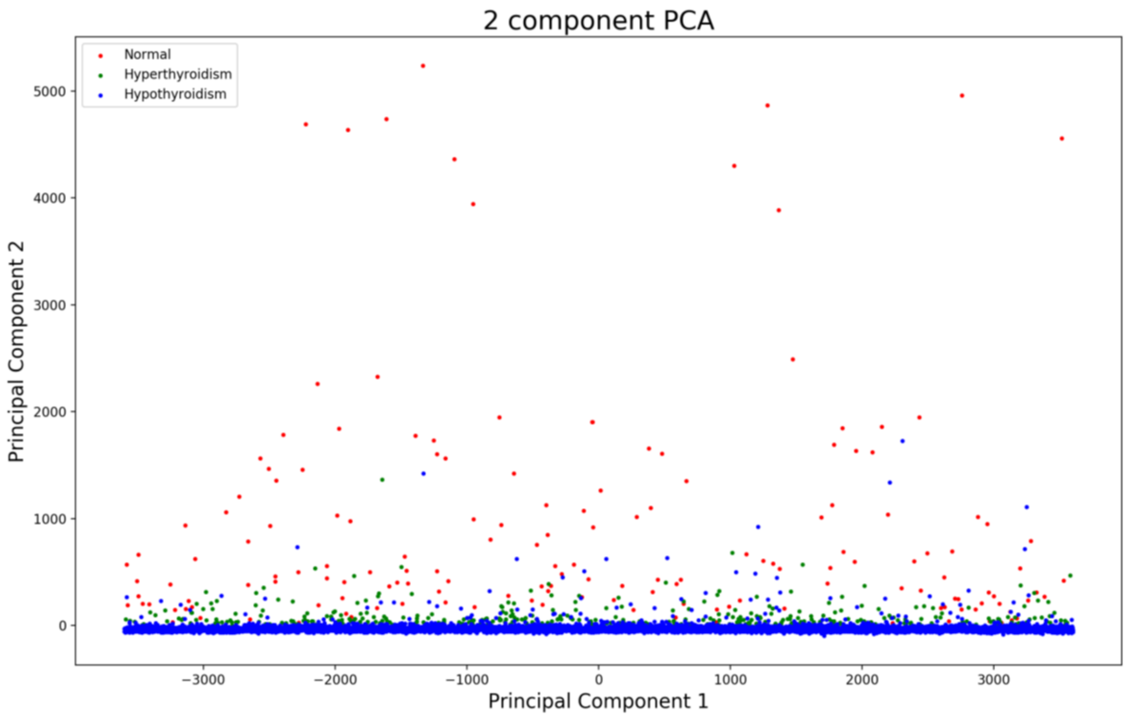


Fig. 6. 6: PCA plot of the dataset

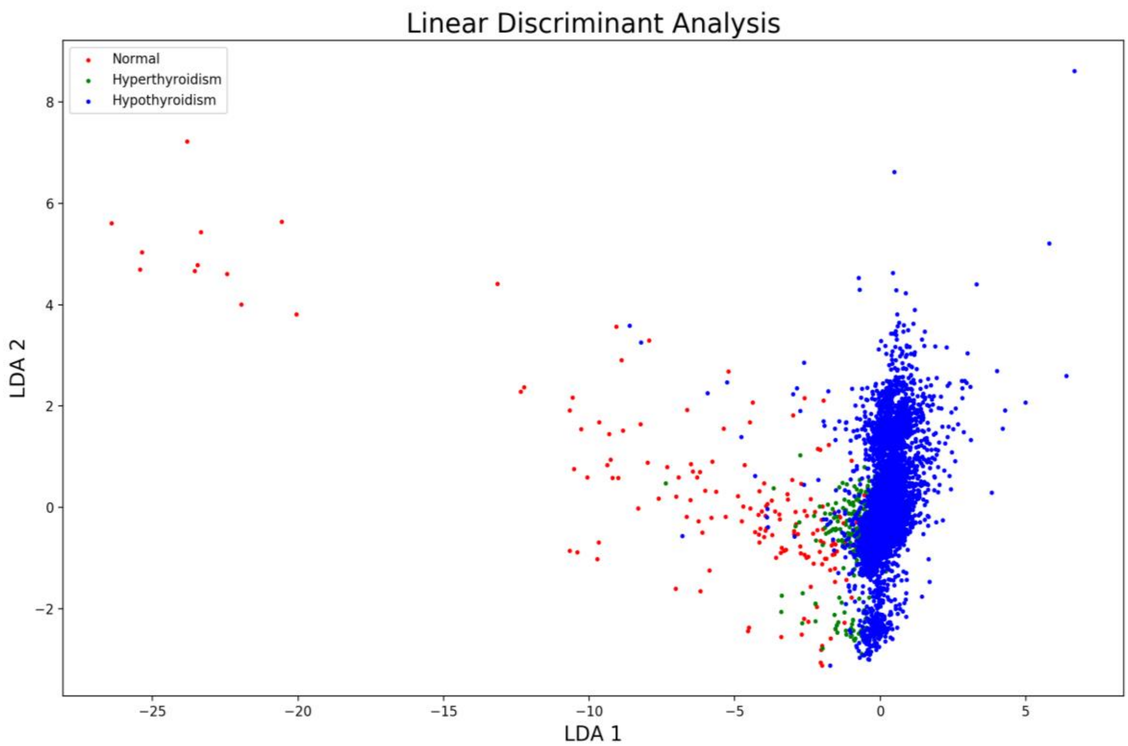


Fig. 6. 7: LDA plot of the dataset

6.3.2 Proposed Improvised Moth Flame Optimization Algorithm

The proposed IMFO algorithm works filter-based [221] for optimal feature selection. The main motive of using filter-based techniques along with MOF is to reduce computational time and cost, as filter-based methods do not require the training of classifiers while evaluating the features. In the proposed IMFO, a fitness function has been introduced, called after each iteration, to evaluate and sort the features according to their respective fitness. All the standard equations of the MGO algorithm are used along with the fitness function to rank each feature's importance. Other than the fitness function, we have applied a constraint according to which the fitness value of each feature is calculated at the end of each iteration and is compared with the best fitness value obtained so far. If the fitness value of the current feature is equal to or greater than the best value, its index is recorded. After completing all the iterations, the feature whose indexes were being recorded is selected, and the rest are dropped from the dataset. The proposed algorithm is explained in the following steps.

Step 1. The first step is the initialization step in which the total number of moths present in the population (m), the number of iterations for which the algorithm will run (max_iter), the size of the position vector of moths (dim), and the logarithmic parameter b, are initialized.

Step 2. Take transpose of the dataset to align features row-wise. The total number of moths will be the total number of rows now. Next, initialize each moth's position as the values present in each row. Restrict the size of the position vector according to dim.

Step 3. In this step, the fitness of each moth is calculated using the fitness function defined in algorithm 3.

Step 4. Case 1: For the first iteration, sort the population according to their fitness value in descending order. Store the best fitness value.

Step 5. Case 2: For the rest of the iterations, double the population and sort them according to their fitness value in descending order. Store the best fitness value

Step 6. Update position of moths for step 4 and step 5 using equation 6.1

Step 7. Now calculate the fitness of the updated population.

Step 8. If the fitness value comes greater or equal to the best fitness value for each moth, return its index.

The input parameters used in the Improvised Moth Flame Optimization Algorithm (IMFO) are shown in table 6.2.

Table 6.2: Input variables and parameters

Parameter	Value	Description
M	21	Total number of moths (search agents)
max_iter	40	Maximum no. of iterations
B	1	Decides shape of logarithmic spiral

The pseudo-code of IMFO and fitness function are presented in algorithms 8 and 9, respectively, and the flowchart is shown in fig. 6.8.

Algorithm 8: Pseudo code of Improvised Moth Flame Optimization Algorithm

Initialize the population of *n* features(moths) from the dataset

Set the dimension

Restrict the size of the position vector according to dimension

while (Iteration <Max_iteration + 1):

    Update N_{flames} *using* equation 6.3

    Call fitness function to evaluate each moth (feature).

    if current iteration = 1

        Sort the moths according to their fitness in descending order

        Append the sorted population in matrix F

    else

        Double the population of moths and flames.

Calculate fitness using a fitness function

Sort the double population in descending order.

Select the N-best flames having the best fitness.

end

Calculate the value of r.

for each $Moth_i$ *with* $i \leq n$ do

Calculate t as $t = (r - 1) * random + 1$. {r lies in [0,1]}

if $i \leq N$, **then**

Update $Moth_i$ position acc. to F_i using eq. 6.1

else

Update $Moth_i$ position acc. to F_N using eq. 6.1

end

end

for each $Moth_i$ *with* $i \leq n$ **do**

if Fitness($Moth_i$) >= best flame score

store index of $Moth_i$

Return index of best features selected

end

Algorithm 9: Pseudo code of Fitness functions

s ← calculate the number of non-zero terms for each x^j_i

return

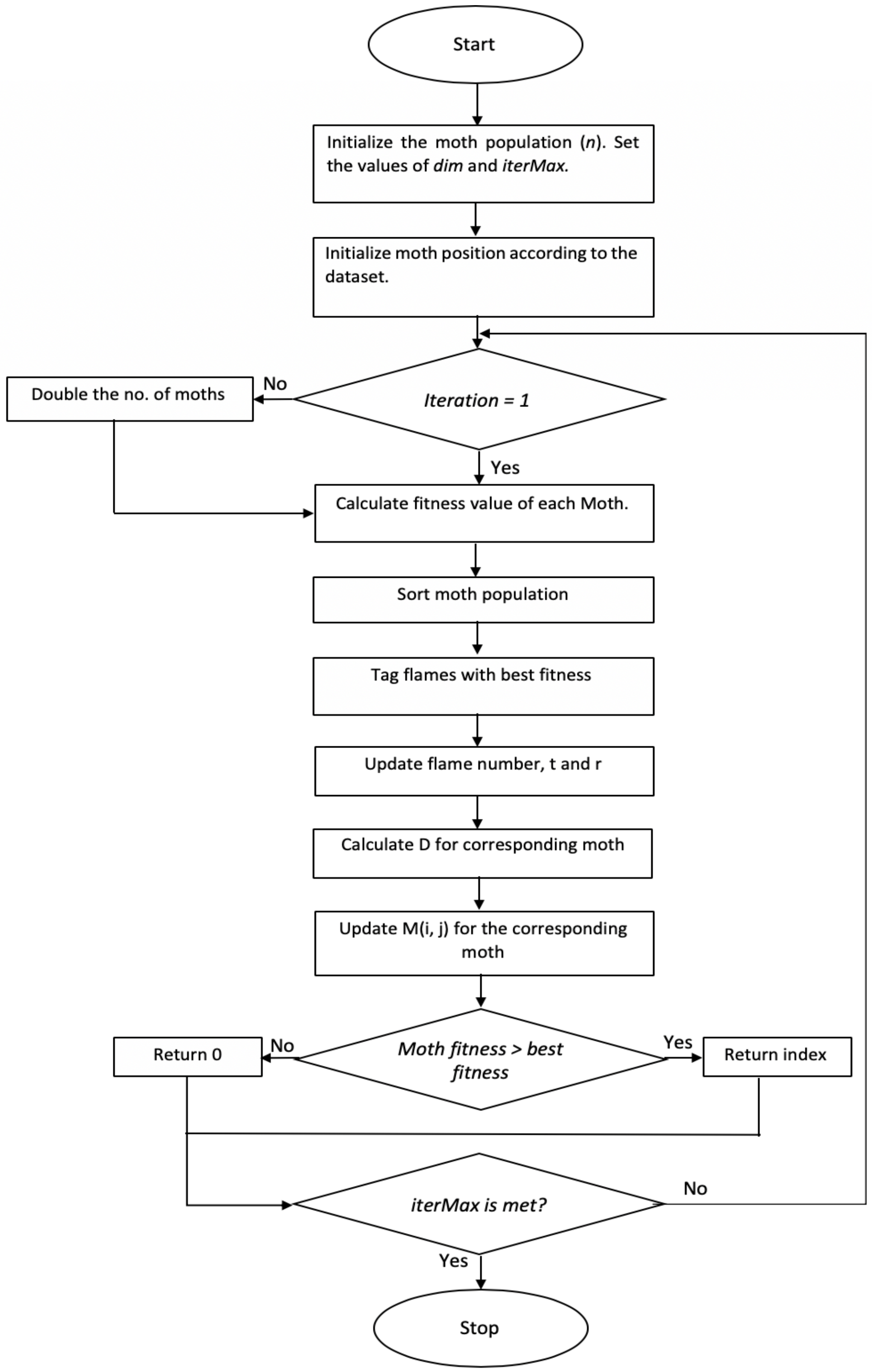


Fig. 6.8: Flowchart: Improvised Moth Flame Optimization (IMFO) Algorithm

6.3.3 Machine Learning Models

This study employed seven computational machine learning classifiers to classify thyroid diseases. Each classifier was set to different tuned parameter settings, as shown in Table 6.3. The classification process was validated by adopting a 10-fold cross-validation method, reducing the bias error.

Table 6.3: ML Classifiers and their variable input parameters

Classifiers	Tuning parameters
KNN	n_neighbors = 10
Random Forest	n_estimators = 220
Support Vector Machine (SVM)	gamma='auto'
Gaussian Naive Bayes (GaussianNB)	Default parameters
Quadratic Discriminant Analysis (QDA)	priors = None, reg_param = 0.0, store_covariance = False, store_covariances = None, to = 0.0001
AdaBoost	base_estimator = None, n_estimators = 50, learning_rate = 1.0, random_state = None
Decision Tree	max_depth=30, min_samples_split=20

6.3.4 Evaluation Criteria

In this subsection, we briefly discuss the methods used to validate the classification performance of the selected feature subset obtained from IMFO. Three indicators are used to evaluate the performance of a classifier, namely, precision (P), recall (R), and F-measure (F), which are calculated using the confusion matrix of the classification result. The P, R, and F are calculated as described in equations (6.4), (6.5) and (6.6).

$$P = \frac{TP}{TP+TN} \tag{6.4}$$

$$R = \frac{TP}{TP+FP} \tag{6.5}$$

$$F = \frac{2PR}{P+R} = \frac{2TP}{2TP+FP+FN} \tag{6.6}$$

6.4 Result Analysis

This section discusses the results acquired when the methodology described in the previous section was applied to the thyroid dataset. The IMFO selected a subset of 5 features out of 21 features. In other words, IMFO eliminated 76.1% of insignificant features. The number of prominent thyroid attributes selected at each iteration is shown in fig. 6.9. Fig. 6.10 displays the fitness value at each iteration to describe the convergence rate of IMFO. The selected optimal feature subset given by IMFO was then input to various machine learning classifiers. The dataset was split into 70% train and 30% test set for classification purposes. Table 6.4 outlines the accuracies achieved when the complete feature set was considered and the optimal feature subset obtained from IMFO over different classifiers. We can highlight that all the classifiers used in this study achieved better accuracy with features selected by IMFO than the original complete feature set. In other words, bio-inspired feature selection is a practical approach to classification problems. Random forest gave the maximum accuracy of 99.61%, followed by 99.55% with Decision Tree and 97.55% with SVM. The accuracy plot for IMFO is shown in fig. 6.11.

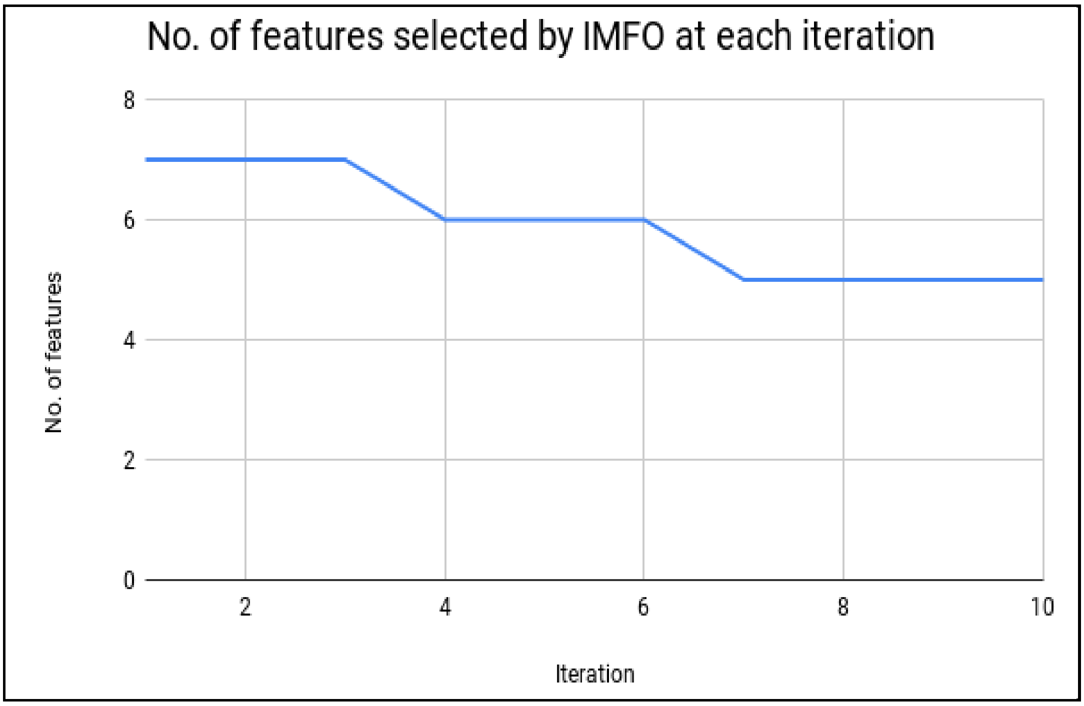


Fig. 6.9: Plot of the number of features selected by IMFO at each iteration

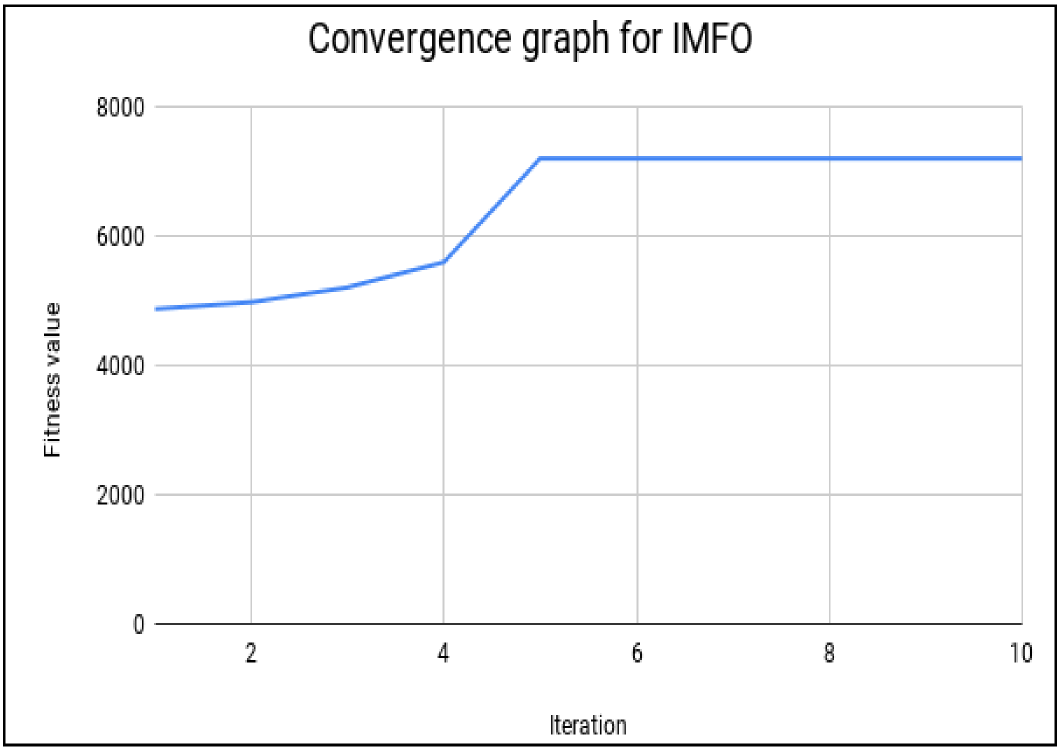


Fig. 6.10: Convergence graph of IMFO calculated on thyroid disease dataset

Table 6.4: Classification Accuracy achieved with and without feature selection by proposed IMFO

ML Classifier	**Accuracy (%)**	
	Without Feature selection	**Improvised Moth Flame Optimization (IMFO)**
KNN	87.44	96.94
Random Forest	90.33	99.61
Decision Tree	89.22	99.55
Support Vector Machine (SVM)	91.94	97.55
Quadratic Discriminant Analysis (QDA)	18.00	93.50
Gaussian Naive Bayes (GaussianNB)	58.50	95.05
AdaBoost	91.05	96.83

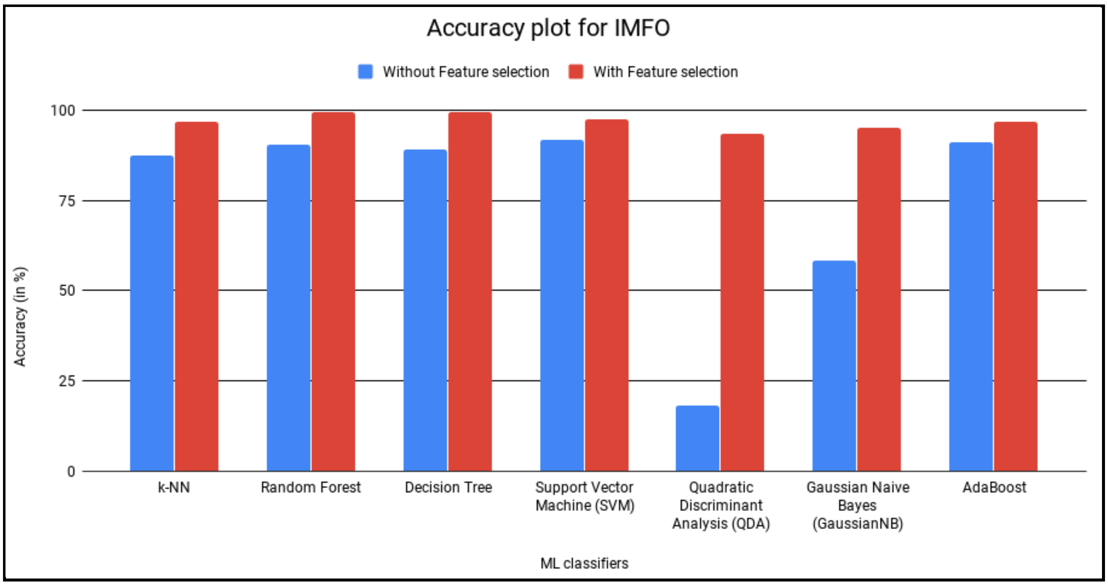


Fig. 6.11: Comparison of classification accuracy without feature selection and with IMFO selected features

6.5 Comparison of proposed IMFO with OCSA, MGWO, OCFA and MALO

This section provides a comparison between the proposed IMFO and four contemporary intelligent bio-inspired algorithms, namely Optimized Crow Search Algorithm (OCSA) [240], Modified Grey Wolf Optimization Algorithm (MGWO) [241], Optimized Cuttlefish Algorithm (OCFA) [242] and Modified Ant-Lion Optimization Algorithm (MALO) [243], which are applied as feature selection methods for the diagnosis of thyroid disease. The algorithms above are carefully chosen from the literature because of their feasibility in being modeled as feature selection methods and their high diagnostic accuracy in the biomedical field of study.

All four algorithms are implemented separately on the same machine and applied to the thyroid dataset used in this study. Table 6.5 and fig. 6.12 outline the number of prominent features selected by the algorithms concerning the total number of features. It is observed that IMFO selected the least number of features, i.e., 23.8%, whereas OCFA selected maximum features of about 42.8%. Also, MALO, MGWO, and OCSA eliminated 71.4%, 66.6%, and 61.9% features, respectively.

Table 6. 5: Number of features selected by IMFO, MALO, OCSA, OCFA and MGWO

Feature Selection Algorithm	No. of features selected	Total no. of features
Improvised Moth Flame Optimization (IMFO)	5	21
Modified Ant Lion Optimization (MALO)	6	21
Optimized Crow Search Algorithm (OCSA)	8	21
Optimized Cuttle Fish Algorithm (OCFA)	9	21
Modified Grey Wolf Optimization (MGWO)	7	21

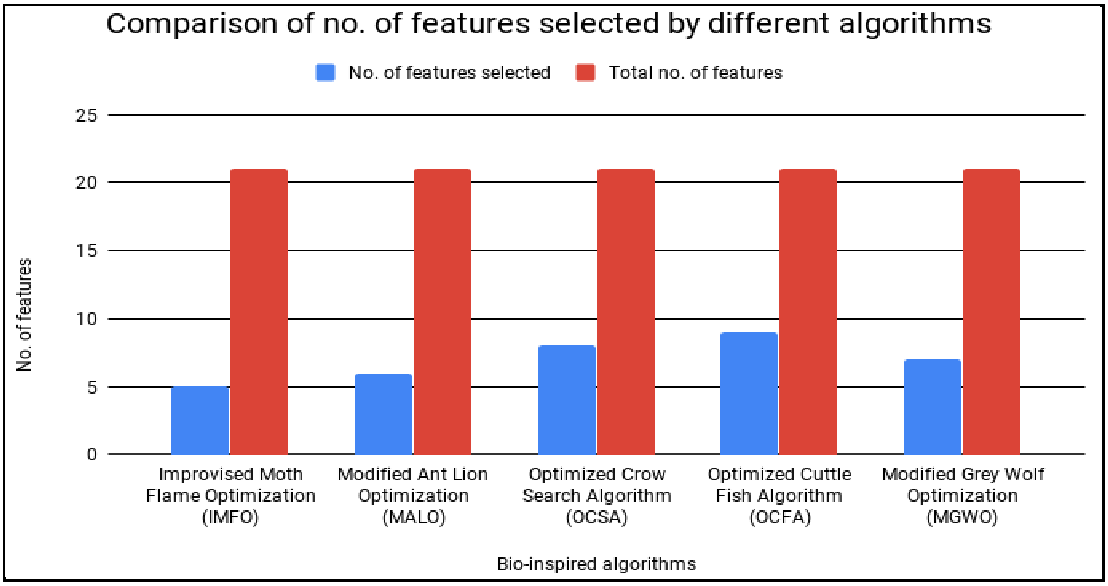


Fig. 6.12: Comparison of the number of features selected by IMFO, MALO, OCSA, OCFA, and MGWO

The classification performance of the significant subset of features selected by the compared algorithms is shown in table 6.6 and fig. 6.13. The comparison shows that the proposed IMFO algorithm achieves the highest accuracy of 99.61% with the Random Forest classifier. OCSA gave the second-best accuracy of 98.83 with the Ada Boost classifier, and MGWO achieved the third-best accuracy of 98.66% with the Random Forest classifier. OCFA and MALO performed comparatively poorly, having maximum accuracies of 95.15% and 98.83%, respectively, as shown in fig. 6.14. Since IMFO performed significantly better in comparison with its contemporary algorithms, it can be highlighted that IMFO can find an optimal combination of prominent features by adaptively searching the feature space. It can also reduce the possibility of premature convergence by avoiding the local minima efficiently. However, it is noteworthy to mention that not only does IMFO outperform the other methods in terms of recognition accuracy, but also filters out the maximum number of insignificant features, confirming the efficiency of IMFO as a feature selection method.

Table 6.6: Comparison of accuracy obtained in percentage by various ML classifiers by proposed IMFO and other algorithms in the literature applied on the same dataset.

ML Classifier	IMFO	MALO	OCSA	MGWO	OCFA
KNN	96.94	92.51	95.11	95.83	92.58
Random Forest	99.61	95.94	98.05	98.66	86.76
Decision Tree	99.55	95.66	97.61	98.44	92.55
Support Vector Machine (SVM)	97.55	93.35	95.77	95.16	90.68
Quadratic Discriminant Analysis (QDA)	93.50	83.74	6.66	8.27	57.32
Gaussian Naive Bayes (GaussianNB)	95.05	76.20	81.61	9.77	61.30
AdaBoost	96.83	96.13	98.83	97.94	95.15

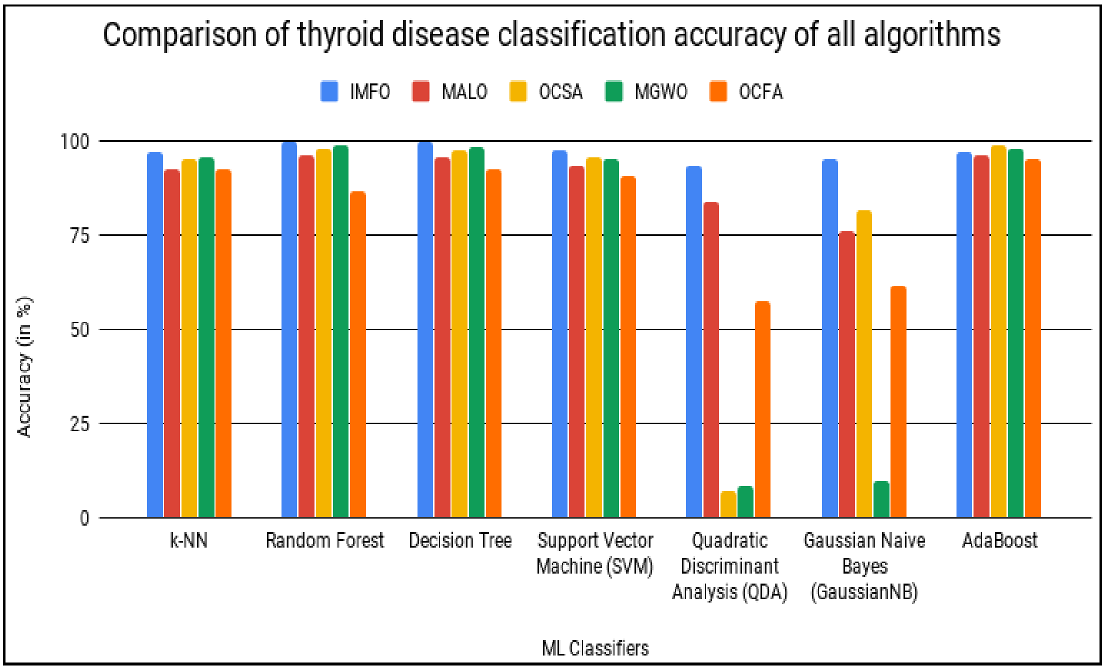


Fig. 6.13: Comparison of Accuracy between proposed IMFO and other bio-inspired feature selection methods present in the literature

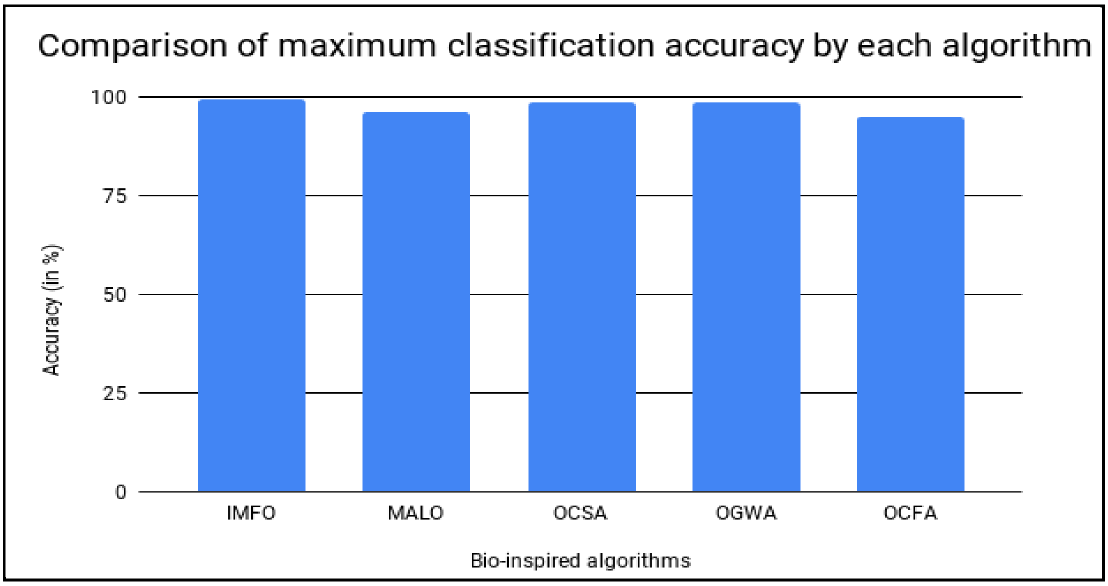


Fig. 6.14: Comparison of maximum classification accuracy achieved by IMFO, MALO, OCSA, OGWA, and OCFA

To validate the classification performance of the selected feature subset obtained by the different algorithms considered in this exposition, precision, recall, and f1 score are calculated for the classification confusion matrix, as shown in Tables 6.7, 6.8 and 6.9 respectively. The comparison of the fisher score calculated for each classifier for all the algorithms above is shown in fig. 6.15.

Table 6.7: Comparison of precision calculated using different ML classifiers for various variants of NIA algorithms

ML Classifier	IMFO	MALO	OCSA	MGWO	OCFA
KNN	0.97	0.91	0.94	0.95	0.91
Random Forest	0.99	0.95	0.98	0.99	0.89
Decision Tree	0.99	0.96	0.98	0.98	0.92
Support Vector Machine (SVM)	0.96	0.92	0.96	0.94	0.87
Quadratic Discriminant Analysis (QDA)	0.94	0.85	0.89	0.93	0.72
Gaussian Naive Bayes (GaussianNB)	0.96	0.73	0.92	0.93	0.84
AdaBoost	0.97	0.97	0.97	0.98	0.93

Table 6.8: Comparison of recall calculated using different ML classifiers for various variants of NIA algorithms

ML Classifier	IMFO	MALO	OCSA	MGWO	OCFA
KNN	0.95	0.91	0.95	0.96	0.91
Random Forest	0.99	0.95	0.98	0.99	0.85
Decision Tree	0.98	0.94	0.98	0.98	0.92
Support Vector Machine (SVM)	0.96	0.93	0.95	0.95	0.89
Quadratic Discriminant Analysis (QDA)	0.92	0.82	0.07	0.08	0.56
Gaussian Naive Bayes (GaussianNB)	0.95	0.75	0.82	0.10	0.59
AdaBoost	0.96	0.96	0.98	0.98	0.94

Table 6.9: Comparison of F1 score calculated using different ML classifiers for various variants of NIA algorithms

ML Classifier	IMFO	MALO	OCSA	OGWA	OCFA
KNN	0.96	0.91	0.94	0.95	0.91
Random Forest	0.99	0.95	0.98	0.99	0.86
Decision Tree	0.98	0.94	0.98	0.98	0.92
Support Vector Machine (SVM)	0.96	0.92	0.94	0.94	0.88
Quadratic Discriminant Analysis (QDA)	0.91	0.83	0.08	0.07	0.67
Gaussian Naive Bayes (GaussianNB)	0.95	0.73	0.84	0.09	0.63
AdaBoost	0.96	0.96	0.97	0.98	0.93

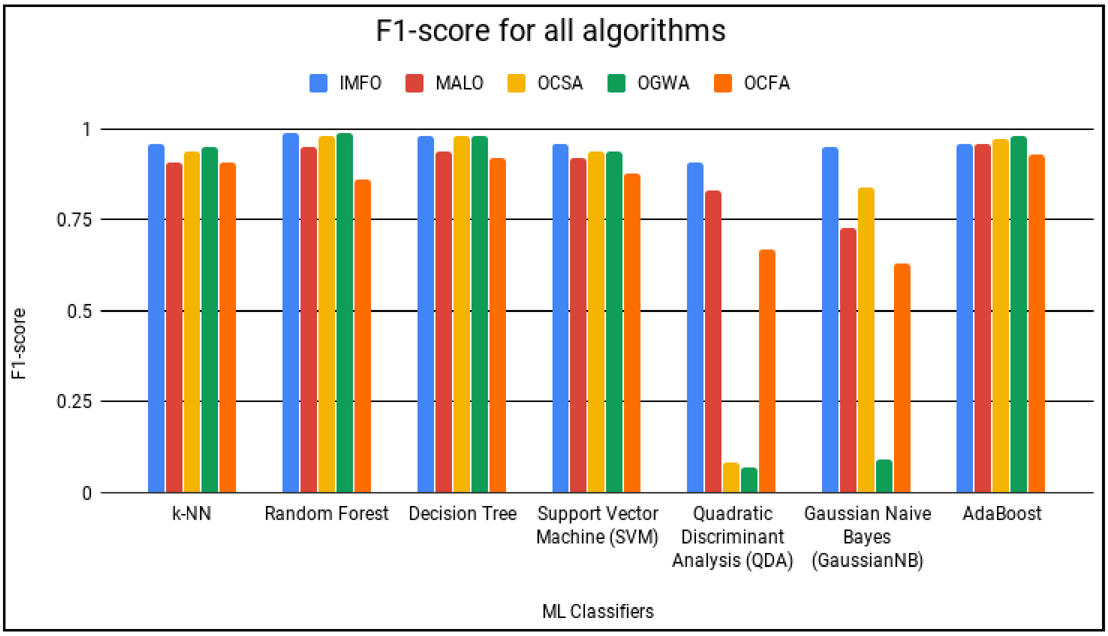


Fig. 6.15: Comparison of F1-score between proposed IMFO and other bio-inspired feature selection methods present in the literature

6.6 Conclusion and Future Scope

This exposition aimed to propose an improvised variant of the Moth Flame Optimization algorithm and to employ it as a filter-based feature selection method to select the minimal number of significant features essential for thyroid disease diagnosis. The optimal subset of prominent features given by the proposed Improvised Moth Flame Optimization algorithm provided better classification accuracy by utilizing all features and required less computational time. In addition, the results prove that the proposed method performs significantly better than the other bio-inspired feature selection method in the literature. After demonstrating the potential and effectiveness of the proposed bio-inspired feature selection method, we can explore its application in the areas of medical image diagnosis. Additionally, the proposed method can be applied to more complex datasets with high dimensionalities to enhance the diagnostic accuracy of thyroid disease. This work can be extended by combining the proposed method with other nature-inspired algorithms to develop a hybrid approach for feature selection optimization problems.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This chapter presents the primary findings and delves into potential avenues for further research, drawing upon the proposed optimization algorithm and methodology.

7.1 Conclusion

Some of the available nature-inspired algorithms including neural networks, particle swarm, genetic, ant colony and grey wolf have a wide range of perspectives on stability and convergence of these reviewed algorithms. The research provides primary insight into the nature and purpose of the novel algorithms developed. Various nature-inspired algorithms have been comprehensively reviewed and analyzed in our research work. Dominant contributors and published articles of selective NIA's have been obtained. Brief overview of various variants of selective nature inspired algorithms and their working phenomenon has been discussed. The potential of these selective algorithms has been summarized using their application scope and objectives that can be contented on the basis of literature.

Briefly this research work not only identifies the popular nature-inspired optimization algorithms but also certain swarm intelligence-based and other nature-based algorithms that have limited literature review and discusses their developments, principles and domain of application to address the gap of unawareness. This research explores the mentioned nature-inspired algorithms for their real-life application in the organizational, social, as well as management space, which depends on their business.

Various machine learning classification algorithms have been used to get a deeper understanding of the dataset and they are applied to various multiclassification problems of biomedical applications. Various feature selection techniques have been investigated. The process of feature selection optimization involves the utilization of both feature selection techniques and nature-inspired algorithms to optimize the chosen features. Previous studies have addressed this issue by employing an iterative approach to achieve convergence towards an ideal collection of features. Feature selection optimization is a methodology that is not limited to a certain domain. The method of feature selection optimization involves a hybrid approach that combines pure feature selection approaches with nature-inspired algorithms in order to optimize the selected features. The field of data science is not without its limitations, as it grapples with many challenges stemming from data-related issues, primarily arising from the unforeseen surge in data volume. The primary issue associated with data is the phenomenon known as the curse of dimensionality (CoD), Various Feature selection approaches like filter, wrapper and embedded are united together with the optimization abilities of nature-inspired algorithms in terms of dimensionality reduction to derive optimal reduced feature subset in the pre-processing stage.

Our research can be concluded as a stage wise process. The first stage, of this process basically includes conversion of input data; the biomedical dataset in any format(image/video/audio/text) is converted to a csv (comma separated values) format using various feature extraction techniques. If the dataset is already in csv format then it is processed as it is. The second stage takes biomedical dataset in csv format as an input and further various existing nature-inspired algorithms or designed novel nature-inspired algorithms which are basically the enhanced and improvised version of the existing nature-inspired algorithms using various feature selection approaches are applied to the biomedical dataset in order to obtain optimal features from the existing set of features in the dataset. These algorithms tend to drop extraneous and irrelevant features thereby performing dimensionality reduction which greatly reduces the cost of computation.

In order to realise the optimality of the reduced feature set obtained in the second stage, various machine learning classifiers are used to evaluate the reduced feature set on the basis of various performance metrics i.e. accuracy, f1 score, sensitivity and others in the third stage of the process. In the final stage of the process comparative analysis with existing algorithms on the particular biomedical dataset is performed in order to declare the supremacy of the designed novel nature-inspired algorithms.

The novel bio-inspired optimization algorithm greatly reduces computational cost by performing dimensionality reduction when implemented to a biomedical dataset; with high accuracy for implementing feature selection. The novel proposed algorithms are effective in terms of feature selection as well as they tend to enhance their traditional counter-parts performance.

7.2 Future Scope

The preliminary knowledge regarding these algorithms further would be convenient for researchers working under the data science and research field for applying them in the real-world NP-hard combinatorial problems.

Further studies can be conducted in the biomedical discipline of disease diagnosis could be performed by combining various nature inspired algorithms and various machine learning models. The hybridization of various nature inspired algorithms can give results with improved accuracies which can aid in timely and accurate detection.

The proposed novel nature-inspired algorithms can also be applied to various other classification or optimization problems i.e. after demonstrating the potential and effectiveness of the proposed nature-inspired feature selection method, we can explore its application in the areas of medical image diagnosis. Additionally, the proposed method can be applied to more complex datasets with high dimensionalities to enhance the diagnostic accuracy of diseases.

This work can be extended by combining the proposed method with other nature-inspired algorithms to develop a hybrid approach for feature selection optimization problems. The proposed quantum-based nature-inspired algorithms can be applied to the optimization of 0-1 problems and various other problems for feature selection.

REFERENCES

[1] H. Zang, S. Zhang and K. Hapeshi, "A Review of Nature-Inspired Algorithms," *Journal of Bionic Engineering,* vol. 7, pp. S232-S237, 2010.

[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, 2018.

[3] I. Jain, V. K. Jain and R. Jain, "Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification," *Applied Soft Computing,* vol. 62, p. 203–215, January 2018.

[4] H. M. Harb and A. S. Desuky, "Feature selection on classification of medical datasets based on particle swarm optimization," *International Journal of Computer Application,* vol. 104, no. 5, pp. 14-17, October 2014.

[5] I. Boussaïd, J. Lepagnot and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.,* vol. 237, pp. 82-117, 2013.

[6] A. Gogna and A. Tayal, "Metaheuristics: review and application," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 25, no. 4, p. 503–526, December 2013.

[7] X. S. Yang, Engineering optimization: an introduction with metaheuristic applications, John Wiley & Sons, 2010.

[8] A. L. Jaimes and C. Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Inf. Sci.,* vol. 277, pp. 1-20, 2014.

[9] R. Kapur, "Review of nature inspired algorithms in cloud computing," *International Conference on Computing, Communication & Automation,* pp. 589-594, 2015.

[10] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Computing and Informatics Journal,* vol. 3, pp. 231-246, 2018.

[11] I. Fister Jr., X. S. Yang, I. Fister, J. Brest and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," 2013.

[12] S. Das and P. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation,* vol. 15, pp. 4-31, 2011.

[13] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC),* pp. 210-214, 2009.

[14] S. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.,* vol. 69, pp. 46-61, 2014.

[15] S. Mirajalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications,* vol. 27, no. 4, p. 1053–1073, May 2015.

[16] X. Yang, Z. Cui, R. Xiao, A. Gandomi and M. Karamanoglu, "Swarm Intelligence and Bio-Inspired Computation: Theory and Applications," 2013.

[17] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Inf. Sci.,* vol. 179, pp. 2232-2248, 2009.

[18] S. Kirkpatrick, C. D. Gelatt and P. M. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, p. 671–680, 1983.

[19] A. Gogna and A. Tayal, "Metaheuristics: review and application," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 25, pp. 503-526, 2013.

[20] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* 1995.

[21] D. Karaboga and B. Basturk , "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization* , vol. 39, p. 459–471, 2007.

[22] M. Dorigo, "Ant colony optimization," IEEE Computational Intelligence Magazine," vol. 1, no. 4, p. 28–39, 2006.

[23] R. Arnay , F. Fumero and J. Sigut, "Ant Colony Optimization-based method for optic cup segmentation in retinal images," *Applied Soft Computing,,* vol. 52, pp. 409-417, March 2017.

[24] K. Z. Gao, P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai and Q. K. Pan, "A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion," *Expert Systems with Applications,* vol. 42, no. 21, pp. 7652-7663, 2015.

[25] X. S. Yang, A. H. Gandomi, S. Taltahari and A. H. Alavi, "Metaheuristics in water, geotechnical and transport engineering," Newnes, 2012.

[26] A. Gandomi, X. Yang, S. Talatahari and A. H. Alavi, "Metaheuristic Applications in Structures and Infrastructures," 2013.

[27] A. Noshadi, J. Shi, W. S. Lee, P. Shi and A. Kalam, "Optimal PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system," *Neural Computing and Applications,* vol. 27, pp. 2031-2046, 2015.

[28] X. Yang and A. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations,* vol. 29, pp. 464-483, 2012.

[29] S. Mirjalili, S. M. Mirjalili and X. S. Yang, "Binary bat algorithm," *Neural Computing and Applications,* vol. 25, no. 3-4, p. 663–681, December 2013.

[30] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa and X. S. Yang, "BBA: A Binary Bat Algorithm for Feature Selection," in *25th SIBGRAPI Conference on Graphics, Patterns and Images,* Ouro Preto, Brazil, 2012.

[31] B. Yang, Y. Lu, K. Zhu, G. Yang, J. Liu and H. Yin, "Feature Selection Based on Modified Bat Algorithm," *IEICE Transactions on Information and Systems,* vol. E100.D, no. 8, pp. 1860-1869, 2017.

[32] D. Gupta and A. K. Ahlawat, "Usability feature selection via MBBAT: A novel approach," *Journal of Computational Science,* vol. 23, pp. 195-203, November 2017.

[33] D. Gupta, J. Arora, U. Agrawal and A. Khanna, "Optimized Binary Bat algorithm for classification of white blood cells," *Measurement,* vol. 143, pp. 180-190, September 2019.

[34] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.

[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995.

[36] A. K. Tripathi, K. Sharma and M. Bala, "A novel clustering method using grey wolf optimizer and mapreduce," *Big Data Research,* vol. 14, pp. 93-100, 2018.

[37] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm,”," *Nature Inspired Cooperative Strategies for Optimization*, pp. 65-74, 2010.

[38] A. K. Tripathi, K. Sharma and M. Bala, "Dynamic frequency based parallel k-bat algorithm for massive data clustering(DFBPKBA)," *International Journal of System Assurance Engineering and Management,* vol. 9, pp. 866-874, September 2017.

[39] A. S. Eesa, A. M. A. Brifcani, and Z. Orman, "Cuttlefish Algorithm – A Novel Bio-Inspired Optimization Algorithm," *International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013 1978,* vol. 4, no. 9, p. 1978, September 2013.

[40] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,10.1016/j.knosys.2015.07.006.," *Knowledge-Based Systems,* vol. 89, pp. 228-249, 2015.

[41] X. B. Meng, Y. Liu, X. Z. Gao and H. Zhang, "A New Bio-inspired Algorithm: Chicken Swarm Optimization," *Advances in Swarm Intelligence,* p. 86–94, October 2014.

[42] M. A. A. Betar and . M. A. Awadallah, "Island bat algorithm for optimization," *Expert Systems with Applications,* vol. 107, p. 126–145, October 2018.

[43] G. I. Sayed, A. Darwish and A. E. Hassanien, "A New Chaotic Whale Optimization Algorithm for Features Selection 10.1007/s00357-018-9261-2," *Journal of Classification,* vol. 35, no. 2, pp. 300-344, July 2018.

[44] S. M. Lim and K. Y. Leong, "A Brief Survey on Intelligent Swarm-Based Algorithms for Solving Optimization Problems," in *Nature-inspired Methods for Stochastic, Robust and Dynamic Optimization*, 2018.

[45] J. Holland, "Adaptation in natural and artificial systems," 1975.

[46] C. R. Reeves and J. E. Rowe, Genetic Algorithms: Principles and Perspectives., Springer Science & Business Media, 2006.

[47] C. Reeves, "Genetic Algorithms," in *International Series in Operations Research & Management Science,* pp. 55-82.

[48] A. H. Gandomi, G. J. Yun and S. X. Yang, "Chaos-enhanced accelerated particle swarm optimization," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 2, p. 327–340, 2013.

[49] Z. H. Zhan, J. Zhang, Y. Li and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 15, no. 6, pp. 832-847, 2011.

[50] M. Dorigo, V. Maniezzo and A. Colorni, "Positive Feedback as a Search Strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.

[51] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical computer science,* vol. 344, no. 2, pp. 243-278, 2005.

[52] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, March 2014.

[53] S. Mirjalili and A. Lewis, "Whale Optimization Algorithms," *Advances in Engineering Software*,, vol. 95, pp. 51-67, 2016.

[54] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, 2019.

[55] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.

[56] S. Kumar and A. Kumar, "A brief review on antlion optimization algorithm," in *International Conference on Advances in Computing, Communication Control and Networking*, Greater Noida, 2018.

[57] J. C. Bansal, H. Sharma, S. S. Jadon and M. Clerc, "Spider Monkey Optimization algorithm for numerical optimization," in *Memetic Computing*, 2014.

[58] P. R. Singh, M. A. Alaziz and S. Xiong, "Modified Spider Monkey Optimization Based on Nelder-Mead method for Global Optimization," *Expert Systems With Applications*,, vol. 110, pp. 264-289, 2018.

[59] S. C. Chu, P. w. Tsai and J. S. Pan, "Cat Swarm optimization," in *Pacific Rim International Conference on Artificial Intelligence*, 2006.

[60] X. Nie, W. Wang and H. Nie, "Chaos Quantum-Behaved Cat Swarm Optimization Algorithm and Its Application in the PV MPPT," *Computational Intelligence and Neuroscience*, pp. 1-11, 2017.

[61] Y. Xu, H. Chen, A. A. Heidari, . J. Luo, Q. Zhang, X. Zhao and C. Li, "An Efficient Chaotic Mutative Moth-flame-inspired Optimizer for Global Optimization Tasks," *Expert Systems with Applications*, vol. 129, pp. 135-155, 2019.

[62] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper Optimization Algorithm: Theory and application," *Advances in Engineering Software,* vol. 105, pp. 30-47, 2017.

[63] G. Li and C. Li, "Hybrid Binary Bat Algorithm with Cross-Entropy Method for Feature Selection," in *4th International Conference on Control and Robotics Engineering* , 2019.

[64] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers & Structures,* vol. 169, pp. 1-12, 2016.

[65] X. Meng, Y. Liu, X. Gao and H. Zhang, "A New Bio-inspired Algorithm: Chicken Swarm Optimization," *Advances in Swarm Intelligence,* vol. 8794, 2014.

[66] K. K, "Micro-genetic algorithms for stationary and non-stationary function optimization," *Intelligent control and adaptive systems,* p. 289–296, 1990.

[67] B. Manderick and P. Spiessens, "Fine-grained parallel genetic algorithm," in *Third International Conference on genetic algorithms*, 1989.

[68] L. M. Rocha , "Contextual genetic algorithms: evolving developmental rules," in *Advances in Artificial Life* , 1995.

[69] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *IEEE international conference on evolutionary computation*, 1996.

[70] G. R. Hari, "Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms," Ph.D. thesis, University of Michigan, 1997.

[71] E. Falkenauer, "The grouping genetic algorithm," in *State of the Art in Global Optimization*, 1996, p. 249–265.

[72] D. Whitley, S. Rana and R. B. Heckendorn, "The island model genetic algorithm: on separability, population size and convergence," *Journal of Computing and Information Technology,* vol. 7, p. 33–47, 1998.

[73] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol Compution*, vol. 2, no. 3, p. 221–248, 1994.

[74] K. Deb, S. Agrawal, A. Pratap and T. Mey, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," International Conference on Parallel Problem Solving from Nature.

[75] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference- point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 4, p. 577–601, 2014.

[76] K. F. Man, T. M. Chan, K. S. Tang and S. Kwong, "Jumping-genes in evolutionary computing," in *IECON 2004 - 30th Annual Conference of IEEE Industrial Electronics Society*, 2004.

[77] S. Janson, E. Alba, B. Dorronsoro and M. Middendorf, "Hierarchical cellular genetic algorithm," in *European conference on evolutionary computation in combinatorial optimization,* 2006.

[78] Y. He and C. W. Huii, "Dynamic rule-based genetic algorithm for large-size single-stage batch schedulingt 21:," *Computer Aided Chemical Engineering,* vol. 21, p. 1911–1916, 2006.

[79] B. Ma and Y. Xia, "A tribe competition-based genetic algorithm for feature selection in pattern classification," *Applied Soft Computing,* vol. 58, p. 328–338, 2017.

[80] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," in *Proceedings of the IEEE*, 2001.

[81] M. B. Ghalia, "Particle Swarm Optimization with an Improved Exploration Balance," in *51st Midwest Symposium on Circuits and Systems*, Knoxville, TN, USA, 2008.

[82] F. v. d. Bergh and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization. In: Evolutionary Computation," in *IEEE Transactions on Evolutionary Computation*, 2004.

[83] Z. H. Zhan, J. Zhang, Y. Li and H. S.-H. Chung, "Adaptive Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 6, pp. 1362-1381, 2009.

[84] S. M. Elsayed, R. A. Sarker and E. M. Montes, "Particle swarm optimization method for constrained optimization Problems," in *IEEE Congress on Evolutionary Computation*, Cancun, Mexico, 2013.

[85] K. Parsopoulos, V. P. Plagianako, G. D. Magoulas and M. N. Vrahatis, "Stretching technique for obtaining global minimizers through Particle Swarm Optimization," in *Particle Swarm Optimization Workshop*, Indianapolis, IN, U.S.A, 2001.

[86] K. E. K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211-224, 2004.

[87] R. Brits, A. P. Engelbrecht and F. v. d. Bergh, "Solving systems of unconstrained equations using particle swarm optimization," in *IEEE International Conference on Systems, Man and Cybernetics,*, Yasmine Hammamet, Tunisia, 2002.

[88] F. v. d. Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," in *IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia, 2002.

[89] R. Brits, A. P. Engelbrecht and F. v. d. Bergh, "A niching particle swarm optimizer," in *4th Asia-Pacific conference on simulated evolution and learning*, 2002.

[90] A. P. Engelbrecht, "Asynchronous particle swarm optimization with discrete crossover," in *IEEE Symposium on Swarm Intelligence*, Orlando, FL, USA, 2014.

[91] L. M. Gambardella, E. D. Taillard and M. Dorigo, "Ant Colonies for the quadratic assignment problem," *Journal of the Operational Research Society,* vol. 20, pp. 167-176, 1999.

[92] M. Dorigo and L. M. Gambardella, "Ant-Q: A Reinforcement Learning Approach to Combinatorial Optimization," 1995.

[93] T. Stutzle and H. Hoos, "MAX-MIN Ant System and local search for the traveling salesman problem," in *IEEE International Conference on Evolutionary Computation*, Indianapolis, IN, USA, 1997.

[94] B. Bullnheimer and R. F. Hartl, "A new rank based version of the Ant System—a computational study," *Central European Journal of Operations Research,* vol. 7, no. 1, pp. 25-38, 1999.

[95] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," *INFORMS Journal on Computing,* vol. 11, no. 4, p. 358–369, 1999.

[96] O. Cordon, V. de, I. F. Herrera and L. Moreno, "A new ACO model integrating evolutionary computation concepts: the best-worst Ant System," in *ANTS 2000*.

[97] M. Guntsch and M. Middendorf, "A population based approach for ACO," in *Proceedings of the Applications of Evolutionary Computing on EvoWorkshops*, 2002.

[98] C. Blum, "Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling," *Computers & Operations Research,* vol. 32, no. 6, pp. 1565-1591, 2005.

[99] C. Blum, A. Roli and M. Dorigo, "HC-ACO: The Hyper-Cube Framework for Ant Colony Optimization.," *IEEE Trans Syst Man Cybern B Cybern,* vol. 34, no. 2, 2001.

[100] S. Tsutsui, "cAS:Ant Colony Optimization with Cunning Ants," in *International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Heidelberg.

[101] L. Wang, Q. Niu and M. Fei, "A novel quantum ant colony optimization algorithm," in *Bio-Inspired Computational Intelligence and Applications*, Berlin, Heidelberg, 2007.

[102] Q. Ding, X. Hu, L. Sun and Y. Wang, "An improved ant colony optimization and its application to vehicle routing problem with time windows," *Neurocomputing,* vol. 98, pp. 101-107, 2012.

[103] J. Xiao, X. T. Ao and Y. Tang, "Solving software project scheduling problems with ant colony optimization," in *Computers and Operations Research*, 2013.

[104] E. G. Talbi, O. Roux, C. fonlupt and D. Robillard, "Parallel ant colonies for the quadratic assignement problem," *Future Generation Computer Systems,* vol. 17, no. 4, p. 441–449, 2001.

[105] C. Ratanavilisagul, "Modified Ant Colony Optimization with pheromone mutation for travelling salesman problem," in *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Phuket, Thailand, 2017.

[106] A. Kishor and . P. K. Singh, "Empirical study of grey wolf optimizer," in *Fifth international conference on soft computing for problem solving*, 2016.

[107] M. Chandra, A. Agrawal, A. Kishor and R. Niyogi, "Web service selection with global constraints using modified gray wolf optimizer," in *International conference on advances in computing, communications and informatics*, Jaipur, India, 2016.

[108] S. Saremi, S. Z. Mirjalili and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications,* vol. 26, no. 5, p. 1257–1263.

[109] Q. Luo, S. Zhang, Z. Li and Y. Zhou, "A novel complex-valued encoding grey wolf optimization algorithm," *Algorithms,* vol. 9, no. 1, 2015.

[110] W. Long, X. Liang, S. Cai, J. Jiao and W. Zhang, "A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems," *Neural Computing Application,* vol. 28, p. 421–438, 2017.

[111] M. R. S. Malik, E. R. Mohideen and L. Ali, "Weighted distance grey wolf optimizer for global optimization problems," in *IEEE International Conference on Computational Intelligence and Computing Research*, Madurai, India, 2015.

[112] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing,* vol. 60, pp. 115-134, 2017.

[113] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of Computational Design and Engineering,* vol. 5, no. 4, pp. 458-472, 2018.

[114] R. A. Ibrahim, M. A. Elaziz and S. Lu, "Chaotic opposition-based grey- wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Systems with Applications,* vol. 108, pp. 1-27, 2018.

[115] S. Dhargupta, M. Ghosh, S. Mirjalili and R. Sarkar, "Selective Opposition based Grey Wolf Optimization," *Expert Systems with Applications,* vol. 151, 2020.

[116] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering,* vol. 5, no. 3, pp. 275-284, 2018.

[117] Y. Ling, Y. Zhou and Q. Luo, "Lévy Flight Trajectory-Based Whale Optimization Algorithm for Global Optimization," *IEEE Access,* vol. 5, pp. 6168-6186, 2017.

[118] H. Alamri, Y. A. Alsariera and K. Z. Zamli, "Opposition-Based Whale Optimization Algorithm," *Journal of Computational and Theoretical Nanoscience,* vol. 24, no. 10, pp. 7461-7464, 2018.

[119] I. N. Trivedi, J. Pradeep, J. Narottam, A. Kumar and L. Dilip, "A Novel Adaptive Whale Optimization Algorithm for Global Optimization," *Indian Journal of Science and Technology,* vol. 9, no. 38, pp. 1-6, 2016.

[120] H. Chen, W. Li and X. Yang , "A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems," *Expert Systems with Applications,* vol. 158, 2020.

[121] H. M. Zawbaa, E. Emary and C. Grosan, "Feature Selection via Chaotic Antlion Optimization," *PLoS ONE,* vol. 11, no. 3, 2016.

[122] K. R. Subhashini and J. K. Satapathy, "Development of an Enhanced Ant Lion Optimization Algorithm and its Application in Antenna Array Synthesis," *Applied Soft Computing,* vol. 59, pp. 153-173, 2017.

[123] S. Saha and V. Mukherjee, "A novel quasi-oppositional chaotic antlion optimizer for global optimization," *Applied Intelligence,* vol. 48, p. 2628–2660, 2017.

[124] N. Sihag, "A Novel Adaptive Ant Lion Optimizer for Global Numerical Optimization," *International Journal of Engineering Research & Technology,* vol. 7, no. 2.

[125] E. Emary and H. M. Zawbaa, "Feature selection via Lèvy Antlion optimization," *Pattern Analysis and Applications,* vol. 22, p. 857–876, 2018.

[126] S. K. Dinkar and K. Deep, "An efficient Opposition based Lévy Flight Antlion Optimizer for Optimization Problems," *Journal of Computational Science,* vol. 29, pp. 119-141, 2018.

[127] C. Chen and L. Yu, "A hybrid ant lion optimizer with improved Nelder–Mead algorithm for structural damage detection by improving weighted trace lasso regularization," *Advances in Structural Engineering,* vol. 23, no. 3, pp. 468-484, 2019.

[128] S. Kumar, V. Sharma and R. K. Poonia, "Self-Adaptive Spider Monkey Optimization Algorithm for Engineering Optimization Problems," *International Journal of Information, Communication and Computing Technology,* vol. 2, p. 96 – 107, 2015.

[129] S. Kumar, V. Sharma and R. K. Poonia, "Modified Position Update in Spider Monkey Optimization Algorithm," *International Journal of Emerging Technologies in Computational and Applied Sciences,* vol. 2, p. 198 – 204, 2014.

[130] G. Hazrati, H. Sharma and N. Sharma, "Adaptive step-size based Spider Monkey Optimization," in *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems,* Delhi, 2016.

[131] K. Gupta, K. Deep and J. C. Bansal, "Spider monkey optimization algorithm for constrained optimization problems," *Soft Computing*, vol. 21, p. 6933–6962, 2016.

[132] K. Gupta and K. Deep, "Tournament Selection Based Probability Scheme in Spider Monkey Optimization Algorithm," in *Advances in Intelligent Systems and Computing*, Springer, Berlin, Heidelberg, 2015.

[133] P. R. Singh, M. A. Elaziz and S. Xiong, "Modified Spider Monkey Optimization based on Nelder–Mead method for global optimization," *Expert Systems with Applications,* vol. 110, pp. 264-289, 2018.

[134] S. Bhattacharyya, V. Snasel, A. Dey, S. Dey and D. Konar, "Quantum Spider Monkey Optimization (QSMO) Algorithm for Automatic Gray-Scale Image Clustering," in *International Conference on Advances in Computing, Communications and Informatics*, Bangalore, 2018.

[135] A. Sharma, A. Sharma, B. K. Panigrahi, D. Kiran and R. Kumar, "Ageist Spider Monkey Optimization Algorithm," *Swarm and Evolutionary Computation,* vol. 28, pp. 58-77, 2016.

[136] A. Dey, S. Dey, S. Bhattacharyya, J. Platos and V. Snasel, "Novel quantum inspired approaches for automatic clustering of gray level images using Particle Swarm Optimization, Spider Monkey Optimization and Ageist Spider Monkey Optimization algorithms," *Applied Soft Computing,* vol. 88, 2020.

[137] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao and S. P. Hao, "Parallel Cat Swarm Optimization," in *International Conference on Machine Learning and Cybernetics*, Kunming, 2008.

[138] P. W. Tsai, J. S. Pan, S. M. Chen and B. Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications,* vol. 39, p. 6309–6319, 2012.

[139] M. Orouskhani, M. Mansouri and M. Teshnehlab, "Average-Inertia Weighted Cat Swarm Optimization," in *Advances in Swarm Intelligence*, 2011.

[140] M. Orouskhani, Y. Orouskhani, M. Mansouri and M. Teshnehlab, "A Novel Cat Swarm Optimization Algorithm for Unconstrained Optimization Problems," *International Journal of Information Technology and Computer Science*, vol. 5, pp. 32-41, 2013.

[141] Y. Kumar and G. Sahoo, "An improved cat swarm optimization algorithm for clustering," in *Computational Intelligence in Data Mining*, 2015.

[142] Y. Kumar and G. Sahoo, "An improved cat swarm optimization algorithm based on opposition-based learning and Cauchy operator for clustering," *Journal of Information Processing Systems*, vol. 13, no. 4, pp. 1000-1013, 2013.

[143] P. M. Pradhan and G. Panda, "Solving multi objective problems using cat swarm optimization," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2956-2964, 2012.

[144] X. Nie, W. Wang and H. Nie, "Chaos quantum-behaved cat swarm optimization algorithm and its application in the PV MPPT," *Computational Intelligence and Neuroscience,* pp. 1-11, 2017.

[145] L. Pappula and D. Ghosh, "Cat swarm optimization with normal mutation for fast convergence of multimodal functions," *Applied Soft Computing,* vol. 66, p. 473–491.

[146] M. Zhao, "A novel compact cat swarm optimization based on differential method," *Enterprise Information Systems,* vol. 14, pp. 196-220, 2018.

[147] V. Savsani and M. A. Tawhid , "Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems," *Engineering Applications of Artificial Intelligence,* vol. 63, pp. 20-32, 2017.

[148] A. A. Elsakaan, . R. A. El-Sehiemy, S. S. Kaddah and M. I. Elsaid, "An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions," *Energy,* vol. 157, pp. 1063-1078, 2018.

[149] X. Yueting , C. Huiling, L. Jie, Z. Qian, J. Shan and Z. Xiao, "Enhanced Moth-flame Optimizer with Mutation Strategy for Global Optimization," *Information Sciences,* vol. 492, pp. 181-203, 2018.

[150] P. Jain and A. Saxena, "An opposition theory enabled moth flame optimizer for strategic bidding in uniform spot energy market," *Engineering Science and Technology,* vol. 22, no. 4, pp. 1047-1067, 2019.

[151] Y. Xu, . H. Chen, A. A. Heidari, J. Luo, Q. Zhang, X. Zhao and C. Li, "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Systems with Applications,* vol. 129, pp. 135-155, 2019.

[152] Y. Caiyang, A. A. Heidari and H. Chen, "A Quantum-behaved Simulated Annealing Enhanced Moth-flame Optimization Method," *Applied Mathematical Modelling,* vol. 87, pp. 1-19, 2020.

[153] M. A. Elaziz, . A. A. Ewees, R. A. Ibrahim and S. Lu, "Opposition- based moth-flame optimization improved by differential evolution for feature selection," *Mathematics and Computers in Simulation,* vol. 168, pp. 48-75, 2020.

[154] A. A. Ewees, M. A. Elaziz and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Systems with Applications,* vol. 112, pp. 156-172, 2018.

[155] X. Chen, Y. Ye, C. Dong, Z. Chen, W. Guo and Y. Huang, "Grasshopper optimization algorithm combining Gaussian and chaos theory for optimization design," in *International Conference on Electronic Information Technology and Computer Engineering*, Xiamen, 2019.

[156] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Computing & Applications,* vol. 31, p. 4385–4405, 2018.

[157] S. Dwivedi, M. Vardhan and S. Tripathi, "An effect of chaos grasshopper optimization algorithm for protection of network infrastructure," *Computer Networks,* vol. 176, 2020.

[158] G. Li and C. Le, "Hybrid Binary Bat Algorithm with Cross-Entropy Method for Feature Selection," in *International Conference on Control and Robotics Engineering*, Nanjing, China, 2019.

[159] X. Huang, X. Zeng and R. Han, "Dynamic Inertia Weight Binary Bat Algorithm with Neighborhood Search," *Computational Intelligence and Neuroscience,* vol. 8, pp. 1-15, 2017.

[160] V. S and R. R, "An Integration of Binary Bat Algorithm and Naïve Bayes Classifier for Intrusion Detection in Distributed Environment," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 6, no. 2, 2017.

[161] G. I. Sayed, A. Darwish and A. E. Hassanien, "Chaotic crow search algorithm for engineering and constrained problems," in *12th International Conference on Computer Engineering and Systems,* Cairo, Egypt, 2017.

[162] S. Khalilpourazari and S. H. R. Pasandideh, "Sine Cosine Crow Search Algorithm: A powerful hybrid meta heuristic for global optimization," in *Third International Conference on Artificial Intelligence and Soft Computing*, 2018.

[163] M. Jain, A. Rani and V. Singh, "An improved Crow Search Algorithm for high-dimensional problems," *Journal of Intelligent & Fuzzy Systems,* vol. 33, pp. 3597-3614, 2017.

[164] D. Gupta, J. J. P. C. Rodrigues, S. Sundaram, A. Khanna, V. Korotaev and V. C. d. Albuquerque, "Usability feature extraction using modified crow search algorithm: a novel approach," *Neural Computing and Applications,* vol. 32, p. 10915–10925, 2020.

[165] S. Shekhawat and A. Saxena, "Development and applications of an intelligent crow search algorithm based on opposition based learning. In: 99:210:230," *ISA Transactions,,* vol. 99, pp. 210-230, 2020.

[166] K. Wang, Z. Li, H. Cheng and K. Zhang, "Mutation chicken swarm optimization based on nonlinear inertia weight," in *3rd IEEE International Conference on Computer and Communications*, Chengdu, China, 2017.

[167] K. Ahmed, A. E. Hassanien and S. Bhattacharyya, "A novel chaotic chicken swarm optimization algorithm for feature selection," in *Third International Conference on Research in Computational Intelligence and Communication Networks,* Kolkata, 2017.

[168] M. Han and S. Liu, "An Improved Binary Chicken Swarm Optimization Algorithm for Solving 0-1 Knapsack Problem," in *International Conference on Computational Intelligence and Security*, Hong Kong, China, 2017.

[169] C. Qu, S. Zhao, Y. Fu and W. He, "Chicken Swarm Optimization Based on Elite Opposition-Based Learning," *Mathematical Problems in Engineering,* vol. 11, pp. 1-20, 2017.

[170] K. Ahmed, A. E. Hassanien, E. Ezzat and P.-W. Tsai, "An Adaptive Approach for Community Detection Based on Chicken Swarm Optimization Algorithm," in *International conference on genetic and evolutionary computing*, Cham, 2016.

[171] A. S. Eesa and Z. Orman, "A new clustering method based on the bio-inspired cuttlefish optimization algorithm," *Expert Systems*, vol. 37, no. 11, 2019.

[172] M. E. Riffi and M. Bouzidi, "Discrete cuttlefish optimization algorithm to solve the travelling salesman problem," in *Third World Conference on Complex Systems*, Marrakech, Morocco, 2015.

[173] M. A. Sameh, M. A. Badr, M. A. L. Badr, M. I. Marei and M. A. Attia, "Optimized PIA Controller for Photovoltaic System Using Hybrid Particle Swarm optimization and Cuttlefish Algorithms," in *7th International Conference on Renewable Energy Research and Applications*, Paris, 2018.

[174] A. Jović, K. Brkić and N. Bogunović, "A review of feature selection methods with applications," in *38th International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, 2015.

[175] N. Hoque, D. Bhattacharyya and J. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Systems with Applications,* vol. 41, no. 14, p. 6371–6385, 2014.

[176] I. H. Witten, E. Frank and M. A. Hall, Data mining: Practical machine learning tools and techniques, San Francisco CA, USA: Morgan Kaufmann, 2011.

[177] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in *20th International Conference on Machine Learning*, Washington DC, USA, 2003.

[178] H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection-A Filter Solution," in *13th International Conference on Machine Learning*, Bary, Italy, 1996.

[179] J. Tang, S. Alelyani and H. Liu, "Feature Selection for Classification: A Review," in *Data Classification: Algorithms and Applications*, CRC Press, 2014, pp. 37-64.

[180] R. O. Duda, P. E. Hart and D. G. Stork, Pattern classification, Wiley, 2012.

[181] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and ReliefF," *Machine Learning,* vol. 53, pp. 23-69, 2003.

[182] S. Alelyani, J. Tang and H. Liu, "Feature Selection for Clustering," in *A Review Data Clustering: Algorithms and Applications*, CRC Press, 2013.

[183] D. M. Witten and R. Tibshirani, "A framework for feature selection in clustering," *Journal of the American Statistical Association,,* vol. 105, no. 490, pp. 713-726, 2010.

[184] Y. Li, M. Dong and J. Hua, "Localized feature selection for clustering," *Pattern Recognition Letters,* vol. 29, no. 1, pp. 10-18, 2008.

[185] D. S. Modha and W. S. Spangler, "Feature weighting in k-means clustering," *Machine Learning,* vol. 52, no. 3, pp. 217-237, 2003.

[186] M. Dash and Y.-S. Ong, "RELIEF-C: Efficient Feature Selection for Clustering over Noisy Data," in *23rd IEEE International Conference on Tools with Artificial Intelligence*, Roca Raton, Florida, USA, 2011.

[187] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research,* vol. 3, pp. 1289-1305, 2003.

[188] T. Liu, S. Liu, Z. Chen and W.-Y. Ma, "An evaluation on feature selection for text clustering," in *20th International Conference on Machine Learning*, Washington DC, USA, 2003.

[189] J. Bins and B. A. Draper, "Feature selection from huge feature sets," in *8th International Conference on Computer Vision*, Vancouver, British Columbia, Canada, 2001.

[190] M. Muštra, M. Grgić and K. Delač, "Breast density classification using multiple feature selection," *Automatika,* vol. 53, p. 1289– 1305, 2012.

[191] N. Dessì, E. Pascariello and B. Pes, "A Comparative Analysis of Biomarker Selection Techniques," *BioMed Research International,* 2013.

[192] H. Abusamra, "A comparative study of feature selection and classification methods for gene expression data of glioma," *Procedia Computer Science,* vol. 23, p. 5–14, 2013.

[193] C. Liu, D. Jiang and W. Yang, "Global geometric similarity scheme for feature selection in fault diagnosis," *Expert Systems with Applications,* vol. 41, no. 8, p. 3585–3595, 2014.

[194] S. Mazumdar, R. Choudhary and A. Swetapadma, "An innovative method for fetal health monitoring based on artificial neural network using cardiotocography measurements," in *Third International Conference on Research in Computational Intelligence and Communication Networks*, 2017.

[195] V. Nagendra, H. Gude, D. Sampath and S. Corns, "Evaluation of Support Vector Machines and Random Forest Classifiers in a Real time fetal monitoring system based on Cardiotocography data," in *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, 2017.

[196] M. L. Huang and Y. Y. Hsu, "Fetal distress prediction using discriminant analysis, decision tree, and artificial neural network," *Journal of Biomedical Science and Engineering,* vol. 5, pp. 526-533, 2012.

[197] A. Batra, A. Chandra and V. Matoria, "Cardiotocography Analysis Using Conjunction of Machine Learning Algorithms," in *IEEE International Conference on Machine Vision and Information Technology (CMVIT)*, Singapore, 2017.

[198] S. Chinnasamy, M. Chitradevi, M. Chitradevi and G. Geetharamani, "Classification of Cardiotocogram Data using Neural Network based Machine Learning Technique," *International Journal of Computer Applications,* vol. 47, no. 14, pp. 19-25, 2012.

[199] A. Tareef, Y. Song, W. Cai, Y. Wang, D. Feng and M. Chen, "Automatic nuclei and cytoplasm segmentation of leukocytes with color and texture-based image enhancement," *International Symposium on Biomedical Imaging,* pp. 935-938, 2016.

[200] D. C. Huang, K. D. Hung and Y. K. Chan, "A computer assisted method for leukocyte nucleus segmentation and recognition in blood smear images," *Journal of Systems and Software,* vol. 85, no. 9, pp. 2104-2118, 2012.

[201] K. A. Dulaimi, I. T. Ryes, J. Banks and V. Chandran, "White Blood Cell Nuclei Segmentation Using Level Set Methods and Geometric Active Contours," *Digital Image Computing: Techniques and Applications,* November 2016.

[202] D. J. Brenner and E. J. Hall, "Computed tomography—an increasing source of radiation exposure," *New England Journal of Medicine,* vol. 357, no. 22, pp. 2277-84, 2007.

[203] B. MacMohan, "Prenatal x-ray exposure and childhood cancer," *Journal of the National Cancer Institute,* vol. 28, no. 5, pp. 1173-1191, 1962.

[204] S. Razia, P. SwathiPrathyusha, N. V. Krishna and N. S. Sumana, "A Comparative study of machine learning algorithms on thyroid disease prediction," *International Journal of Engineering and Technology,* vol. 7, no. 2.8, pp. 315-319, 2018.

[205] K. Agrawal and M. Dhakar, "Thyroid Prediction System using Auto Associative Neural Network," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 6, no. 4, 2016.

[206] K. Shankar, S. Lakshmanaprabu, D. Gupta, A. Maseleno and V. Albuquerque, "Optimal feature-based multi-kernel SVM approach for thyroid disease classification," *The Journal of Supercomputing*, vol. 76, no. 28, pp. 1-16, 2018.

[207] J. H. Miao and K. H. Miao, "Cardiotocographic diagnosis of fetal health based on multiclass morphologic pattern predictions using deep learning classification," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 1-11, 2018.

[208] Z. Alfirevic, D. Devane, G. M. Gyte and A. Cuthbert, "Continuous cardiotocography (CTG) as a form of electronic fetal monitoring (EFM) for fetal assessment during labour," *Cochrane Database Syst. Rev.*, vol. 2, no. 5, 2017.

[209] P. Fergus, M. Selvaraj and C. Chalmers, "Machine learning ensemble modelling to classify caesarean section and vaginal delivery types using Cardiotocography traces," *Computers in Biology and Medicine*, vol. 93, p. 7–16, 2018.

[210] B. M. Kumar and R. B. Ashok, "Soft Computing Using GWO (Grey Wolf Optimization) for the Performance Improvement of High-Speed Brushless DC Motor," in *International Conferencon Emerging Trends and Innovations Engineering And Technological Research*, Ernakulam, India, 2018.

[211] P. R. Hof and E. V. D. Gucht, "Structure of the cerebral cortex of the humpback whale, Megaptera novaeangliae (Cetacea, Mysticeti, Balaenopteridae)," *The Anatomical Record*, vol. 290, no. 1, pp. 1-31, 2007.

[212] W. A. Watkins and W. E. Schevill, "Aerial observation of feeding behavior in four baleen whales: Eubalaena glacialis," *Journal of Mammalogy*, vol. 60, no. 1, p. 155–163, 1979.

[213] J. A. Goldbogen, A. S. Friedlaender, J. Calambokidis, M. F. McKenna, M. Simon and D. P. Nowacek, "Integrative approaches to the study of baleen whale diving behavior, feeding performance and foraging ecology," *BioScience*, vol. 63, pp. 90-100, 2013.

[214] S. Akama, Elements of quantum computing, Berlin: Springer, 2015.

[215] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, 1994.

[216] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing,* vol. 26, no. 5, p. 1484–1509, 1997.

[217] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 6, pp. 580-593, 2002.

[218] A. Layeb, "A novel quantum inspired cuckoo search for knapsack problems," *International Journal of Bio-Inspired Computation,* vol. 3, no. 5, p. 297–305, 2011.

[219] A. Layeb, "A hybrid quantum inspired harmony search algorithm for 0–1 optimization problems," *Journal of Computational and Applied Mathematics,* vol. 253, pp. 14-25, 2013.

[220] S. Zhao, G. Xu, T. Tao and L. Liang, "Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 2009-2015, 2009.

[221] I. Guyon and A. Elisseeff, "An introduction to variable and attribute selection," *The Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[222] X.-S. Yang, "Bat Algorithm: Literature Review and Applications," *International Journal of Bio-Inspired Computation,* vol. 5, 2013.

[223] K. Khan, A. Nikov and A. Sahai, "A fuzzy bat clustering method for ergonomic screening of office workplaces," in *Third International Conference on Software, Services and Semantic Technologies*, 2011.

[224] X.-S. Yang, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation,* vol. 3, no. 5, pp. 267-274, 2011.

[225] K. Ganesan and A. Wahi, "An optimized K-means clustering techniqueusing bat algorithm," *European Journal of Scientic Research,*, vol. 84, no. 2, pp. 263-273, 2012.

[226] S. H. Rezatofighi and H. Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," *Computerized Medical Imaging and Graphics,* vol. 35, no. 4, pp. 333-343, 2011.

[227] H. Ramoser, V. Laurain, H. Bischof and R. Ecker, "Leukocyte segmentation and classification in blood-smear images," in *IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, China, 2005.

[228] T. Bergen, D. Steckhan, T. Wittenberg and T. Zerfass, "Segmentation of leukocytes and erythrocytes in blood smear images," in *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, 2008.

[229] E. Cuevas, D. Oliva, M. Díaz, D. Zaldivar, M. Pérez-Cisneros and G. Pajares, "White Blood Cell Segmentation by Circle Detection Using Electromagnetism-Like Optimization," *Computational and mathematical methods in medicine,* vol. 5, 2013.

[230] C. Zhang, X. Xiao, X. Li, Y.-J. Chen, W. Zhen, J. Chang, C. Zheng and Z. Liu, "White Blood Cell Segmentation by Color-Space-Based K-Means Clustering, Switzerland).," *Sensors,* vol. 14, no. 9, pp. 16128-16147, 2014.

[231] M. Ramla, S. Sangeetha and S. Nickolas, "Fetal health state monitoring using decision tree classifier from cardiotocography measurements," in *Second International Conference on Intelligent Computing and Control Systems*, Madurai, India, 2018.

[232] F. Marzbanrad, L. Stroux and G. D. Clifford, "Cardiotocography and beyond: a review of onedimensional Doppler ultrasound application in fetal monitoring," *Physiological Measurement* , vol. 39, no. 8, 2018.

[233] A. Subasi, B. Kadasa and E. Kremic, "Classification of the cardiotocogram data for anticipation of fetal risks using bagging ensemble classifier," *Procedia Computer Science,* vol. 168, pp. 34-39, 2020.

[234] A. Pinas and E. Chandraharan, "Continuous cardiotocography during labour: Analysis, classification and management," *Best Practices & Research Clinical*

Obstetrics & Gynaecology, vol. 30, pp. 33-47, 2016.

[235] R. Nurani, E. Chandraharan, V. Lowe, A. Ugwumadu and S. Arulkumaran, "R. NuMisidentification of maternal heart rate as fetal on cardiotocography during the second stage of labor: The role of the fetal electrocardiograph," *Acta Obstetricia et Gynecologica Scandinavica,* vol. 91, no. 12, pp. 1428-32, 2012.

[236] K. Afors and E. Chandraharan, "Use of continuous electronic fetal monitoring in a preterm fetus: clinical dilemmas and recommendations for practice," *Journal of Pregnancy,* 2011.

[237] M. W. Ashcraft and A. J. V. Herle, "Management of thyroid nodules. I: History and physical examination, blood tests, x-ray, and ultrasonography," *Head and neck surgery,,* vol. 3, no. 3, pp. 216-227, 1981.

[238] J. Hsieh, Computed tomography: principles, design, artifacts, and recent advances, Bellingham WA: SPIE, 2009.

[239] J. E. Freitas, M. D. Gross, S. Ripley and B. Shapiro, "Radionuclide diagnosis and therapy of thyroid cancer: a current status report," *Seminars in nuclear medicine,* vol. 15, no. 2, pp. 106-131, 1985.

[240] D. Gupta, S. Sundaram, A. Khanna, A. E. Hassanien and V. H. C. d. Albuquerque, "Improved diagnosis of Parkinson's disease using an optimized crow search algorithm," *Computers and Electrical Engineering,* vol. 6, pp. 412-424, 2018.

[241] P. Sharma, S. Sundaram, M. Sharma, A. Sharma and D. Gupta, "Diagnosis of Parkinson's disease using modified grey wolf optimization," *Cognitive Systems Research,* vol. 54, pp. 100-115, 2019.

[242] D. Gupta, A. Julka, S. Jain, T. Aggarwal, A. Khanna, N. Arunkumar and V. H. C. d. Albuquerque, "Optimized cuttlefish algorithm for diagnosis of Parkinson's disease," *Cognitive systems research,* vol. 52, pp. 36-48, 2018.

[243] N. Gupta, R. Jain, D. Gupta, A. Khanna and A. Khamparia, "Modified Ant-Lion Optimization algorithm for improved diagnosis of Thyroid disease," in *Cognitive Informatics and Soft Computing* , Singapore, 2020.

LIST OF PUBLICATIONS

- P. Sharma and K. Sharma, “A novel quantum- inspired binary bat algorithm for leukocytes classification in blood smear,” Expert Systems, vol. 39, no. 3, Sep. 2021, doi: https://doi.org/10.1111/exsy.12813.

- P. Sharma and K. Sharma, “Fetal state health monitoring using novel Enhanced Binary Bat Algorithm,” Computers and Electrical Engineering, vol. 101, p. 108035, Jul. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022. 108035.

- P. Sharma and K. Sharma, “Theoretical Development based Nature-Inspired Optimization Algorithms: Applications and Challenges,” 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Nov. 2022, doi: https://doi.org/10.1109/icccis56430.2022. 10037686.

- P. Sharma and K. Sharma, “Optimized Classification of fetal state health using GWO and WOA,” May 2022, doi: https://doi.org/10.1109/setit54465.2022. 9875521.

- Paper title: “Efficient Diagnosis of Thyroid disease using Improvised Moth Flame Optimization algorithm” is accepted in” Computer Modeling in Engineering & Sciences “SCIE in May 2022