Science World Journal Vol. 19(No 3) 2024 https://dx.doi.org/10.4314/swj.v19i3.32
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Full Length Research Article

SWARM INTELLIGENT OPTIMIZATION ALGORITHMS FOR PRECISION GENE SELECTION IN MICROARRAY-BASED CANCER CLASSIFICATION

[1]Inuwa Yakubu Shallangwa, [1]Aminu Ali Ahmad, [*2]Jeremiah Isuwa, and [3]Emmanuel Bulus Yahaya

[1]Department of Computer Science, Gombe State University, Gombe, Nigeria
[2]Department of Computer Science, Federal University of Kashere, Gombe State, Nigeria
[3]Department of Computer Science, Federal College of Horticulture, Dadinkowa, Gombe State, Nigeria

*Corresponding Author Email Address: isuwa.jeremiah@fukashere.edu.ng

ABSTRACT

Cancer Disease remains a global health concern, demanding exploration into its causal factors for early detection and treatment. However, cancer data often presents a high-dimensional challenge for analysis. Selecting only relevant cancer genes can significantly enhance this analysis process. Traditional gene selection techniques such as heuristic methods have been employed over the years but proved infeasible. Thus, Swarm Intelligence algorithms known for their global search capabilities were developed. Nonetheless, the performance of these Swarm Intelligence algorithms is often influenced by their methods of initialization, affecting convergence, solution quality, and overall robustness. Chaos-based initialization methods have shown promise, yet their effectiveness remains underexplored in initializing SI algorithms. This research conducted a comprehensive performance comparison of three Swarm Intelligence algorithms: Particle Swarm Optimization, Salp Swarm Algorithm, and Firefly Algorithm. These algorithms were enhanced by incorporating the logistic chaotic map for initialization, specifically in the context of microarray cancer gene selection tasks. To assess the effectiveness of these enhanced algorithms, two cancer datasets were employed, namely Ovarian and Colon, and utilized two classifiers: the k-nearest neighbor and multilayer perceptron. The results of the study demonstrate that the logistic-chaos firefly algorithm paired with the k-nearest neighbor stands out as a significant performer, achieving an impressive overall accuracy rate of 93.95% while selecting 444 genes. In summary, the proposed logistic-chaos firefly algorithm paired with the k-nearest neighbor approach proves itself as a worthy competitor in gene selection tasks.

Keywords: Swarm Intelligence, Cancer Disease, Microarray Gene Selection, Chaotic Initialization

INTRODUCTION

Microarray gene expression analysis is a powerful tool in bioinformatics for studying the expression levels of thousands of genes simultaneously (Alshareef et al., 2022). The analysis of microarray data can provide valuable insights into biological processes, identify potential biomarkers for diseases, and improve our understanding of complex diseases such as cancer (Alshareef et al., 2022; Nouri-Moghaddam et al., 2021). However, the high dimensionality of microarray data and the large search space of potential gene combinations make it a challenging task to identify a subset of genes that can accurately predict a biological outcome or disease status (Qin et al., 2022).

Traditional gene selection methods have been employed by several researchers over the years, particularly in cancer research (Alomari et al., 2021; Dabba, Tari, Meftali, et al., 2021). One of the simplest yet most computationally intensive methods is the brute force approach (Huda & Banka, 2020). It entails evaluating all possible combinations of genes to find the subset that optimizes a predefined criterion, such as classification accuracy. However, due to its exponential growth in the number of subsets with increasing dimensionality, it is often impractical for high-dimensional datasets (Isuwa et al., 2022; Jeremiah et al., 2022).

Recursive Feature Elimination (RFE), a wrapper method, offers a more practical alternative (Abd-Elnaby et al., 2021). RFE operates iteratively, commencing with all available genes and then systematically removing the least important genes based on a specified criterion, typically their contribution to a predictive model (Sharma & Rani, 2019). The process repeats until a predetermined number of features is reached (Sharma & Rani, 2019). This method is more computationally efficient than brute force while providing a reasonably good subset of genes (Adamu et al., 2021; Alrefai & Ibrahim, 2022).

Forward selection is another wrapper method that initiates with an empty set of genes (Blot et al., 2018; Huda & Banka, 2019). It progressively adds the gene that contributes the most to the model's performance in each iteration. Forward selection continues until a stopping criterion is met, such as a specific number of features or a predefined level of model performance (Chaudhuri & Sahu, 2021). It tends to be more computationally efficient than brute force, making it a practical choice when incrementally selecting features. Backward elimination, similar to forward selection, begins with all available genes. However, it removes the gene with the least impact on model performance during each iteration. Like forward selection, backward elimination is more computationally efficient than brute force and is suitable for systematic feature reduction (Jeremiah et al., 2023).

Swarm Intelligence (SI) such as Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Salp Swarm Algorithms (SSA)(Aljarah et al., 2020), and Cuckoo Search (CS) (Alzaqebah, Briki, et al., 2021), inspired by the collective behavior of social organisms like ants, bees, and birds, has gained significant attention and utility in various fields, especially cancer research (Ayham et al., 2019). The growing adoption of SI methods can be attributed to several compelling factors. Firstly, SI algorithms offer a unique approach to problem-solving. They leverage decentralized decision-making, collaboration, and adaptation, mirroring the robustness and efficiency found in natural systems (Hussain et al., 2018). This paradigm shift has introduced innovative ways to tackle complex optimization and search problems. Secondly, advancements in computational capabilities have made it increasingly feasible to implement and scale SI algorithms (Hussain et al., 2018). Modern computing infrastructure and parallel processing allow researchers and practitioners to harness the power of large ensembles of agents, enhancing the algorithms' effectiveness (Chakraborty et al., 2022). Thirdly, the applicability of SI extends across a wide range of domains beyond cancer research (Hussain et al., 2018). From optimizing complex functions in mathematics and engineering to addressing real-world challenges in robotics, logistics, and data analysis, these algorithms have demonstrated versatility and adaptability (Yang et al., 2022).

SI algorithms, while powerful and versatile in solving complex optimization problems, do have some limitations (Deng et al., 2022). One key challenge is the potential to get trapped in local optima, particularly in multimodal and high-dimensional search spaces such as cancer datasets (Brezočnik et al., 2018; Jeremiah et al., 2023). This limitation can restrict their ability to find the global optimum, leading to suboptimal solutions. Additionally, they might require a significant number of iterations to converge to a satisfactory solution, which can be computationally expensive (Brezočnik et al., 2018). Furthermore, the convergence speed and solution quality of swarm algorithms can be highly dependent on the algorithm's parameters, making their fine-tuning a non-trivial task (Isuwa et al., 2021). This is where chaotic map initialization plays a crucial role.

Chaotic maps introduce an element of randomness and unpredictability into SI algorithms, making them less likely to become stuck in local optima (Assiri, 2021). By injecting controlled chaos into the algorithm's behavior, chaotic maps can enhance the exploration of the search space, helping the algorithm discover diverse and potentially better solutions (Dash et al., 2019; Liu et al., 2022). Chaotic maps also contribute to adaptability, enabling SI algorithms to respond effectively to changes in the optimization landscape (Alshareef et al., 2022; Rupa et al., 2023). Overall, the integration of chaotic maps in SI algorithms addresses some of their shortcomings, making them more robust, efficient, and capable of tackling complex optimization problems effectively.

In light of these considerations, this paper aims to present gene selection methods based on SI by leveraging three of its prevalent algorithms: Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), and Firefly Algorithm (FA). These algorithms will be uniformly enhanced with the chaotic initialization techniques. Furthermore, the research will comprehensively evaluate the performance of the proposed methods, comparing them side by side, in terms of classification accuracy, F1 measure, and the counts of selected features. This evaluation will be conducted using datasets from ovarian and colon cancer with varying feature sizes. Specifically, we:

i. Designed an improved initialization strategy in three SI algorithms (PSO, SSA, and FA) for cancer gene selection by utilizing chaotic maps.
ii. Performed a comparative analysis of the performance of the improved algorithms from (i) with benchmark ovarian and colon cancer datasets to determine the optimal, using KNN and MLP as separate learning algorithms in terms of classification accuracy, F1 measure, and the number of selected features.
iii. Performed further hyperparameter tuning, which includes adjusting the population size and the number of search agents among others, to further enhance performance.

The rest of the paper is structured as follows: In Section 2, a concise review of the field's foundational knowledge and discussions of current literature are presented. Section 3 provides a detailed description of the proposed gene selection methods. Section 4 presents the experimental results along with a thorough analysis. Lastly, Section 5 concludes the paper.

BACKGROUND

This section provides an overview of the field's background knowledge as well as discussions of current literal works.

Microarray Data

The human body comprises numerous cells, each containing a copy of the genome encoded in Deoxyribonucleic acid (DNA) (Abel et al., 2020; Jain et al., 2018). Advanced DNA microarray technologies are employed to extract information from these cellular samples, generating valuable data. DNA microarray experiments, also referred to as microarray data, are a high-throughput genetic analysis method (Ravindran & Gunavathi, 2023). They aim to simultaneously assess the expression levels of millions of genomic genes, providing insights into the cellular state. As per Kumar & Rath, (2016), the process of acquiring microarray data encompasses four key stages: cell lysis, genetic content separation, identification of relevant genes, and compilation of the identified genes into a list. Figure 1 offers a visual representation of the intricate processes involved in microarray analysis.

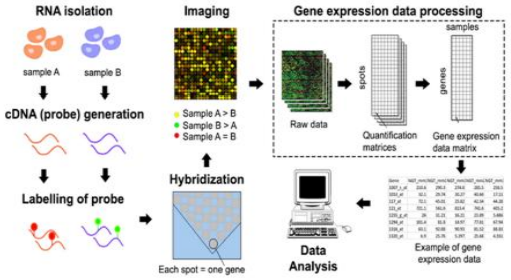

Figure 1: Illustration of the Microarray Analysis Procedure (Hasri et al., 2017)

Microarray data offer valuable insights into various biological processes, including the development of diseases like cancer (Hasri et al., 2017). Nevertheless, the analysis and interpretation of these data pose significant challenges due to the vast number of genes involved. In many cases, the number of genes greatly exceeds the number of available samples, leading to computational complexities and other difficulties (Chaudhuri & Sahu, 2021). The gene expression data is typically represented as a two-dimensional array denoted as G, with thousands of genes (dimension D) and a limited number of samples (N). G can be mathematically expressed as in Equation 1:

$$G = \{X_{nd} \mid n = 1, 2, 3, \ldots, N,\ d = 1, 2, 3, \ldots, D\} \qquad (1)$$

Microarray Gene Selection

Microarray gene selection is the process of identifying a subset of genes that are particularly relevant to a specific research question from a larger set of genes represented on a microarray (Ravindran & Gunavathi, 2023). The aim is to reduce analysis complexity by focusing on a smaller group of genes that offer the most valuable insights. This selection process primarily involves two approaches: filter-based methods and SI (Chaudhuri & Sahu, 2021).

Filter methods

Filter-based methods employ statistical or computational filters to assess genes based on specific criteria and choose the highest-ranked genes for further examination (Chaudhuri & Sahu, 2021). These gene selection techniques can be categorized into two groups: univariate and multivariate (Isuwa et al., 2023). In univariate approaches, individual features are evaluated to gauge their association with the target disease. Common univariate filter methods include Chi-Square, Mutual Information (MI), and Information Gain (IG). Conversely, multivariate methods can concurrently consider multiple features and assess them as groups, rather than individually, as observed in univariate analysis. Two notable multivariate techniques are the Minimum Redundancy Maximum Relevance (mRmR) (Song et al., 2021), and the Fast Correlation-Based Filter (FCBF) (Deng et al., 2022).

Swarm Intelligence (SI) Algorithms

The essence of SI algorithms lies in their primary focus on populations. These algorithms rely on the collective intelligence and behaviors of groups of individuals, as opposed to the isolated actions of individual agents (Alrefai & Ibrahim, 2022). In essence, they function by evolving a population of candidate solutions, adhering to the mathematical structure specific to the SI algorithm, in pursuit of the optimal solution (Alrefai & Ibrahim, 2022). Each candidate solution within this population represents a potential answer to the optimization problem, which, in this context, pertains to finding an optimal feature subset (Baliarsingh et al., 2019). Population-based algorithms excel in mitigating the risk of getting stuck in local minima due to the interaction and learning among multiple individuals. However, it's worth noting that involving multiple individuals increases the computational load due to the need for multiple evaluations using a machine learning algorithm to assess solution quality (Dabba, Tari, & Meftali, 2021). Khurma et al., (2022) provide a succinct mathematical representation illustrating how the total number of evaluations typically depends on the number of individuals and the number of iterations:

$$Evaluation\ (population\ based)\ =\ N\ *\ T \qquad (2)$$

where N is the population size and T the number of iterations. The general framework of SI algorithms is represented in Figure 2.

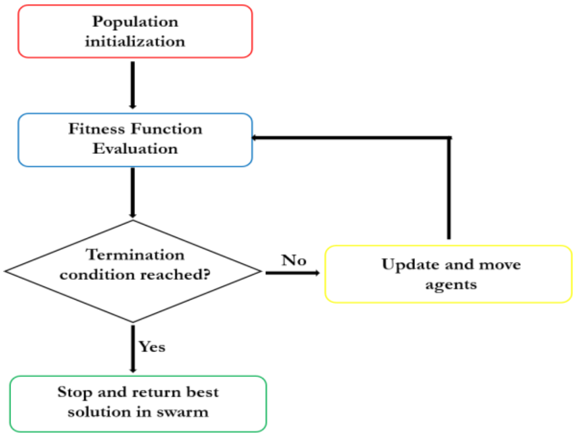
Population initialization
Fitness Function Evaluation
Termination condition reached?
No
Update and move agents
Yes
Stop and return best solution in swarm

Figure 2: General Framework of SI Algorithms

In Figure 2, the algorithm starts by initializing a population of search agents, represented as N and D, where N corresponds to the number of search agents and D signifies the dimension or number of features. This initialization phase also encompasses the configuration of algorithm parameters, including acceleration coefficients and inertia weight for PSO, if applicable. Subsequently, the initialized solutions undergo assessment through a learning algorithm, such as KNN, to identify the most optimal solution.

The termination condition is continually monitored to determine whether the process should persist or conclude (Dabba, Tari, & Meftali, 2021). If the termination condition remains unmet, solutions are updated based on mathematical structures inherent to the specific SI algorithm, to refine them towards the optimum solution. Conversely, if the termination condition is satisfied, the current best solution is returned as the ultimate output.

Particle swarm optimization (PSO)

PSO draws its inspiration from the realm of artificial life research (Chen et al., 2023). The fundamental operation of PSO commences with the random initialization of a swarm comprising $'N'$ particles within a population. Each particle's position within this swarm corresponds to a prospective solution within a D-dimensional search space (Adamu et al., 2021). The process of initializing particle positions is governed by Equation 3.

$$X_{i,d} = L_d^{min} + r_i^d(U_d^{max} - L_d^{min}) \qquad (3)$$

where $i = 1,2,3,\ldots,N\ and\ d = 1,2,\ldots,D$, L_d^{min} represents the lower bound of the search space for d^{th} dimension, U_d^{max} represents the upper bound of the search space for d^{th} dimension, and r_i^d represents a random number in the range [0,1]. The velocity, which plays a crucial role in determining both the speed and direction of a particle's movement within the search space, is initialized following Equation 4.

$$V_{i,d} = [L_d^{min} + r_i^d(U_d^{max} - L_d^{min}) - X_{i,d}]/2 \qquad (4)$$

where $i = 1,2,3,\ldots,N\ and\ d = 1,2,\ldots,D$. A particle's position is subject to modification based on a combination of factors, including its inertia, personal best position, and the swarm's best position (Houssein et al., 2021). In this context, D denotes the dimensionality of the search space, x_{id}^k represents the position of the i^{th}particle along the d^{th}dimension for the k^{th} generation, and V_{id}^k signifies the velocity of the i^{th}particle along the d^{th}dimension for the k^{th} generation. The updates to both velocity and position for each particle within the population adhere to the equations outlined in (5) and (6) respectively.

$$V_{id}^{k+1} = \Omega V_{id}^K + C_1 r_1^k(pbest_{id}^k - x_{id}^k) + C_2 r_2^k(gbest_d^k - x_{id}^k) \qquad (5)$$

$$X_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \qquad (6)$$

$$pbest_i^{t+1} = \begin{cases} X_i^{k+1}\ if\ f(X_i^{k+1}\ better\ than\ pbest_i^k) \\ pbest_i^k\ otherwise \end{cases} \qquad (7)$$

Here, within the equations, $pbest_{id}^k$ and '$gbest_d^k$ denote the particle's personal best position and the swarm's best position within the D-dimensional space for the k^{th} generation, respectively. The symbol "Ω" represents the inertia weight, while "C_1" and "C_2" represent the acceleration constants. These constants play a role in attracting the particles towards 'pbest' and 'gbest' respectively (Alzaqebah, Jawarneh, et al., 2021). When these constants are multiplied by random numbers "r_1" and "r_2" (within the range [0,1]), they introduce controllable stochastic influences on the swarm's velocity (Alzaqebah, Jawarneh, et al., 2021).

Salp swarm optimization (SSA)

Salps, marine organisms characterized by their cylindrical body shape, are known to aggregate in the ocean, intentionally forming groups or chains as they float (Mirjalili et al., 2017). Scientific studies confirm that this collective behavior among Salp swarms enhances their overall mobility and significantly improves their ability to efficiently locate food sources. The SSA, introduced by Mafarja & Mirjalili, (2017) leverages this natural behavior to address various optimization problems. In SSA, the swarm of salps collaborates to create a chain-like structure, facilitating their exploration of the search space in search of target locations or food sources (Mirjalili et al., 2017). This chain formation is mathematically represented by dividing the swarm into two key components: the leader and the followers. The leading Salp consistently occupies the front position, guiding the other followers within the chain (Tubishat et al., 2020).

Let's consider a collection of n Salps denoted as $Y = \{Y_1, Y_2, \ldots, Y_i, \ldots. Y_n\}$, where each Salp is represented by a d-dimensional vector ($Y_i = y_1, y_2, \ldots, y_d$). The target vector or food source is denoted as F_s. The position of the leader Salp is updated according to (8).

$$Y_1 \begin{cases} F_s + \alpha 1((Y_{max} - Y_{min})\alpha 2 + Y_{\min})\ \alpha 3 \geq 0 \\ F_s - \alpha 1\ ((Y_{max} - Y_{min})\alpha 2 + Y_{min})\ \alpha 3 < 0 \end{cases} \qquad (8)$$

In this context, we work with random values denoted as $\alpha 1, \alpha 2,$ and $\alpha 3$, and Y_1 signifies the location of the leading Salp. Y_{max}, and Y_{min} are used to represent the upper and lower boundaries for each Salp, respectively. Within the SSA framework, the equilibrium between exploration and exploitation is controlled by $\alpha 1$, and its value undergoes updates in each cycle according to the equation (9).

$$\alpha 1 = 2e^{-(\frac{4*c_{iter}}{Max_{iter}})^2} \qquad (9)$$

In this context, we work with the variables Max_{iter}, representing the total number of iterations, and c_{iter}, which denotes the current iteration in progress. The positions of the follower Salps, except for Y_1, are improved following Newton's law of motion, as detailed in equation (10).

$$Y_j(i) = \frac{1}{2}at^2 + v_o t \qquad (10)$$

In this scenario, our variable j spans from $2\ to\ n$ where n signifies a certain value. Within this context, $Y_j(i)$ designates the i^{th} dimension of thej^{th} Salp. Furthermore, the initial velocity, time, and acceleration, denoted as v_o, t, and a respectively, are determined through computations based on equation (11).

$$a = \frac{v_{end}}{v_o}\ where\ v = \frac{y - y_o}{t} \qquad (11)$$

In the context of optimization problems, the concept of 'time' corresponds to the count of iterations, with the initial velocity initialized to zero. Consequently, the positions of the follower Salps are adjusted using a modified equation as outlined in (12).

$$Y_j(i) = \frac{1}{2}\left(Y_j(i) + Y_{J-1}(i)\right) \qquad (12)$$

The SSA follows a systematic process. Initially, a random population is generated, and among the Salps, one is identified as the most suitable solution for the problem, akin to a food source (Tubishat, Ja, et al., 2021). Subsequently, the remaining Salps adjust their positions to approach this identified food source. The location of this food source is updated with each iteration (Tubishat, Ja, et al., 2021).

Firefly Algorithm (FA)

The Firefly Algorithm (FA) draws its inspiration from the flashing behavior of fireflies. Developed by Yang, (2009), this algorithm has gained widespread recognition as an effective optimization technique applied across various domains, including science and engineering. The flashing behavior in fireflies serves purposes such as attracting mates or prey, and in the context of the FA, it is harnessed to evaluate potential solutions to optimization problems (Zouache & Ben Abdelaziz, 2018). The algorithm commences by establishing a population of fireflies randomly distributed within the search space. Each firefly is assigned a brightness value based on its fitness, with brighter fireflies signifying superior solutions (Dash et al., 2019). Fireflies are naturally inclined to gravitate towards those that are brighter in the population, and this attraction is governed by the firefly's attractiveness, which is directly linked to its brightness and inversely related to the distance separating two fireflies, mirroring the behavior of light intensity diminishing with distance (Dash et al., 2019). The algorithm concludes when specific stopping criteria, such as reaching a maximum number of iterations or achieving a predefined fitness threshold, are met (Dash et al., 2019).

The development of FA is based on three foundational principles, it is essential to understand these key rules (Jati & Manurung, 2013). Firstly, all fireflies are inherently attracted to each other, irrespective of their gender (Jati & Manurung, 2013). Secondly, their level of attractiveness is contingent upon their brightness, and this attractiveness diminishes as the distance between them increases (Jati & Manurung, 2013). Consequently, it is the less luminous fireflies that tend to gravitate towards their brighter counterparts. Thirdly, a firefly's brightness is influenced by the specific form of the objective functions used (Jati & Manurung, 2013). The Firefly Algorithm operates as a population-based SI algorithm, with each firefly representing a potential solution within the search space. When delving into the FA, two critical aspects warrant consideration: the variance in brightness intensity and the formulation of attractiveness. In the conventional FA, attractiveness primarily relies on brightness, which is intricately linked to the objective function (Jati & Manurung, 2013).

Hence, the brightness of a given firefly denoted as $I(x)$ can be expressed as proportional to $f(x)$ at a specific location x. Similarly, the attractiveness, represented by β, is relative and contingent on the distance, denoted as r_{ij}, between firefly i and j. Consequently, attractiveness varies from one firefly to another solely based on their respective distances. This relationship between light intensity and distance can be mathematically defined using Equation (13).

$$I = I_O e^{-\gamma r^2 ij} \qquad (13)$$

In this context, I_O represents the initial light intensity. The attractiveness, denoted as β, of a firefly is intricately linked to its brightness. This relationship can be quantified as demonstrated in Equation 14:

$$\beta = \beta_O e^{-\gamma r^2 ij} \qquad (14)$$

Here, β_O signifies the attractiveness of the firefly when $r = 0$. The light absorption coefficient, represented as γ, is a constant and remains fixed at 1.0 in the context of FA. To calculate the distance between two fireflies, i, and j, situated at x_i and x_j, respectively, the Cartesian distance is determined using Equation 15:

$$r_{ij} = \left|\left|x_{i\,-}\ x_j\right|\right| = \sqrt{(x_i - x_j)^2 + (y_{i-}y_j)^2} \qquad (15)$$

The motion of firefly i towards a more attractive firefly j is computed using Equation 16, as follows:

$$x_i = x_i + \beta_O e^{-\gamma r^2 ij}(x_{j-}\,x_i)^2 +$$
$$\alpha(rand\ 0.5) \qquad (16)$$

In Equation 15, the second term corresponds to the attraction force, while the third term represents a randomizing factor controlled by α, which falls within the range of [0,1]. The constant β_O is consistently set to 1, and rand signifies a random number within the interval [0,1]. In the FA, the parameter α is configured to introduce variability in the solutions. The parameter γ, which characterizes the degree of attractiveness variation, typically spans values between 0.01 and 100 across various applications.

Related Literary Works

The advent of DNA Microarray technology has empowered scientists to concurrently examine the expression levels of numerous genes. Microarray gene selection, conversely, involves identifying the most pertinent genes while eliminating redundant and inconsequential ones. A critical application of Microarray data analysis lies in cancer classification. Nevertheless, the challenge of dealing with high-dimensional data poses a formidable obstacle when it comes to classifying gene expression. A brief review of the literature about improvements in different single and hybrid SI algorithms for microarray gene selection is presented in this section.

In the realm of cancer classification research, Tang et al., (2005) introduced the Hybrid Particle Swarm Optimization-Support Vector Machine (PSO-SVM) and Artificial Bee Colony-Support Vector Machine (ABC–SVM) methods, aimed at selecting the most informative genes for accurate classification. Among these two innovative approaches, it was observed that the ABC-SVM method outshone the others, achieving an impressive 88% accuracy rate in classifying cancer types. These findings hold significant promise for improving the precision of cancer diagnoses and treatment strategies.

Another noteworthy development in gene expression data analysis comes from Zhang et al., (2018), who identified the Support Vector Machine based on Recursive Feature Elimination and Parameter Optimization (SVM-RFE-PO) as a highly effective feature extraction technique for classifying gene expression data. Furthermore, their research unveiled the potential of the Support Vector Machine based on Recursive Feature Elimination and Particle Swarm Optimization (SVM-RFE-PSO) algorithm in extracting essential genetic information from expression data. This breakthrough holds considerable importance as it aids in unraveling the intricate relationships between genes and cancer, facilitating a deeper understanding of disease mechanisms and enhancing clinical diagnostic accuracy.

Jain et al., (2018) embarked on a comprehensive evaluation of their proposed model using 11 benchmark microarray datasets encompassing various cancer types. Their model outperformed seven other well-established methods, demonstrating superior classification accuracy and gene selection capabilities in most scenarios. Remarkably, it achieved remarkable classification accuracy rates of up to 100% for seven out of the eleven datasets, while utilizing only a small-sized prognostic gene subset (up to 1.5%). These findings underscore the potential of their model in advancing cancer classification research and applications.

Utami & Rustam, (2019) delved into the realm of breast cancer detection using innovative techniques. They employed Particle Swarm Optimization-Support Vector Machine (PSO–SVM) and Artificial Bee Colony- Colony-support vector machine (ABC–SVM) methods to detect breast cancer symptoms. Interestingly, their research indicated that the ABC–SVM method surpassed the PSO–SVM method, boasting an accuracy rate of 88%. Such findings suggest that the ABC–SVM method holds promise for clinical experts seeking to enhance breast cancer stage classification, ultimately leading to more effective treatments and improved patient outcomes.

Additionally, Khurma et al., (2020) introduced multiple binary versions based on the Moth Flame Optimization (MFO) algorithm to address the feature selection (FS) problem in medical datasets. They ingeniously incorporated chaotic maps into their approach to enhance the MFO's performance in the feature space. The results of their study, which featured 23 medical datasets from esteemed repositories, showcased that the chaotic operators had a remarkable impact on improving the standard BMFO's performance when optimizing feature search spaces. This research sets the stage for further exploration of metaheuristic-based wrapper methods, suggesting the possibility of proposing new modification strategies and exploring different metaheuristic algorithms for feature space examination.

On a different note, Wu & Guan, (2007) introduced a novel watermarking algorithm founded on chaotic maps. This unique approach utilized one map to encrypt the embedding position and another to determine the pixel bit for embedding within a host image. An intriguing aspect of this scheme was its dependence solely on a private key for watermark extraction, rendering it a blind watermarking scheme. Extensive simulations confirmed the scheme's robustness against various signal processing operations and geometric attacks, underscoring its potential as a secure and reliable watermarking technique.

The Proposed Method

In this section, the paper provides a comprehensive explanation of the operational principles underlying the logistic map, the gene selection methods earlier proposed, and an overview of the methods' overall architecture.

Principles of the logistic chaotic map

The Logistic chaotic map works by iteratively generating a sequence of numbers based on a nonlinear mathematical function known as the Logistic map (Liu et al., 2022). Here's a simplified explanation of how it works:

i. Initialization: Start with an initial value (seed) between 0 and 1. This initial value serves as the starting point for the chaotic sequence.

ii. Iteration: Apply the Logistic map formula iteratively to generate the sequence of numbers. The formula for the Logistic map is typically expressed as:

$$q_i^{t+1} = aq_i^t(1 - q_i^t)$$

- q_i^{t+1} represents the next value in the sequence
- q_i^t is the current value
- a is a parameter known as the control parameter? It determines the behavior of the chaotic map.

iii. Repeat: Continue iterating the formula, where each new value q_i^{t+1} becomes the current value q_i^t for the next iteration.

iv. Chaotic Behaviour: Depending on the value of the control parameter "a", the Logistic map exhibits chaotic behavior. This means that even small changes in the initial seed or "a", can lead to vastly different sequences of numbers. Chaotic sequences are characterized by their sensitivity to initial conditions, unpredictability, and apparent randomness.

The Proposed Gene Selection Method

Figure 3 illustrates the overarching framework of the proposed methods. In this depiction, an initial cancer gene dataset undergoes preprocessing to prepare it for subsequent analysis. The preprocessed data is subsequently employed as input for a designated SI algorithm, namely PSO, SSA, or FA, and is initialized using the logistic chaotic map. The subsequent steps align with the conventional SI algorithm process, with the notable distinction that we incorporate KNN and MLP separately for performance evaluation in addressing the specific task at hand.

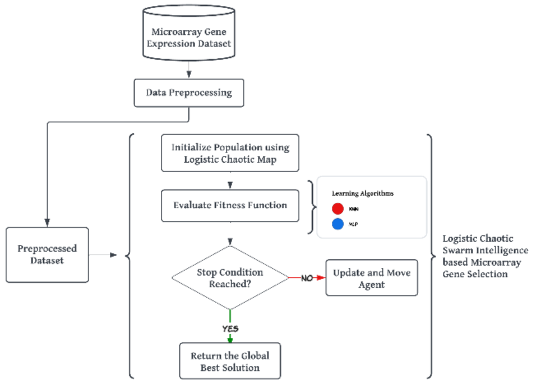
Microarray Gene Expression Dataset
Data Preprocessing
Preprocessed Dataset
Initialize Population using Logistic Chaotic Map
Evaluate Fitness Function
Learning Algorithms
KNN
MLP
Stop Condition Reached?
NO
Update and Move Agent
YES
Return the Global Best Solution
Logistic Chaotic Swarm Intelligence based Microarray Gene Selection

Figure 3: General Architecture of the Proposed Methods

Dataset description

This research evaluate the performance of our developed gene selection techniques using two distinct microarray datasets. Our selected datasets are the Colon and Ovarian cancer datasets, both widely employed in previous research as standard benchmarks for evaluating competing algorithms. These datasets are publicly accessible on the website https://figshare.com/. A summary of the dataset characteristics is presented in Table 1, which also includes additional notations (x, y) to indicate the distribution of instances in each class. In this notation, $'x'$ signifies the number of instances in the positive class (tumor or malignant), while $'y'$ represents the number of instances in the negative class (normal or benign).

Table 1: Overview of the Microarray Cancer Datasets used

Datasets	Instances	Features	Class
Colon Cancer	62	2000	2 (40,22)
Ovarian Cancer	253	15154	2 (162,91)

Learning Algorithms

Classifiers are essential in microarray gene selection for evaluating and optimizing the selection of genes to distinguish between different sample categories, such as benign and malignant groups. They enable the assessment of gene subsets' performance, ranking of genes based on their relevance, and systematic exploration of different feature combinations. Classifiers also help ensure model generalization to unseen data and are employed in optimization processes to identify the most informative gene subset while minimizing dimensionality.

Thus, in this study, two distinct learning algorithms were utilized: KNN (K-Nearest Neighbors) machine learning algorithms and MLP (Multilayer Perceptron) to assess and compare their performance in the context of microarray gene selection. This evaluation aims to test the effectiveness of the developed gene selection methods.

KNN and MLP have been selected individually based on their respective strengths and suitability for the task. KNN is chosen for its simplicity and ability to handle nonlinear data patterns effectively (Adamu et al., 2021; Isuwa et al., 2022). It's a non-parametric algorithm that doesn't make strong assumptions about the underlying data distribution, making it suitable for gene expression data with complex relationships (Tubishat, Ja'afar, et al., 2021).

On the other hand, MLP, or Artificial Neural Network, is selected due to its capacity to model intricate and nonlinear relationships within data (Pantic et al., 2023). ANNs can learn from large datasets and capture hidden patterns, making them well-suited for gene expression analysis where the interactions between genes can be highly intricate (Woldseth et al., 2022).

Result Presentation and Analysis

Experimental setup

All experiments in this study are carried out using the Python programming language within a Jupyter Notebook integrated development environment. The experiments are executed on a computer equipped with an Intel(R) Core TM i5-5300U CPU running at 2.30GHz and 8.00 GB of RAM.

Logistic Chaotic Map

The Logistic chaotic map finds frequent application in a range of optimization and search algorithms, spanning domains like image encryption (Liu et al., 2022), multimedia security (Rupa et al., 2023), microarray cancer analysis (Nouri-Moghaddam et al., 2021), and more. It stands out as a favorable choice for several reasons. Firstly, it exhibits strong chaotic properties, ensuring unpredictability and a broad exploration of the search space, which is crucial in optimization tasks (Adamu et al., 2021; Yang et al., 2022). Secondly, its simple mathematical formulation makes it computationally efficient, reducing the algorithm's computational burden (Da Silva & Gertrudes, 2022).

Moreover, the Logistic map offers a balance between ergodicity and sensitivity, striking a harmonious trade-off between thorough exploration of the solution space and quick convergence to optimal or near-optimal solutions (Rupa et al., 2023). These qualities collectively render the Logistic chaotic map a reliable and efficient choice in various optimization and search algorithms, contributing to its widespread adoption. Other chaotic maps include the Circle, Chebyshev, Gauss, Sine, and Piecewise among others (Arora & Singh, 2017).

Therefore, this work utilized the logistic chaotic map to enhance the initialization approach in PSO, SSA, and FA. This involves substituting the conventional random variables with the logistic chaotic map, contributing to improved initialization strategies in these algorithms.

Fitness Function

In pursuit of the overarching objective of gene selection, which aims to optimize classification accuracy while minimizing the number of chosen features, it is imperative to establish a method for quantifying the merit of potential solutions. This quantification is achieved through a fitness function, a mathematical construct employed to evaluate the excellence of solutions in the context of SI optimization problems (Naseri & Hasheminejad, 2019). The fitness function serves as a gauge of how effectively a particular solution aligns with the optimization objectives of the task at hand problems (Naseri & Hasheminejad, 2019). It assigns a fitness score to each candidate solution, thereby guiding the optimization algorithm toward the discovery of superior solutions. In light of these considerations, Equation 17 defines the fitness function employed in this study for all the SI algorithms under examination.

$$fitness\ function = \alpha \Delta_R(D) + \beta \frac{|Y|}{|T|} \qquad (17)$$

In this context, $\Delta_R(D)$ denotes the error rate of the classifier, with $|Y|$ representing the dimensionality of the chosen gene subset and $|T|$ signifying the total number of genes in the dataset. The parameter 'α' takes on values within the range [0,1] and determines the impact of the classifier's error rate. Conversely, $'\beta'$ is equivalent to $(1-\alpha)$ and signifies the level of importance assigned to gene reduction.

Transfer Function

In this study, the three SI algorithms have been transformed into a unified single-objective framework and discretized to optimize the balance between classification accuracy and the count of selected genes. To accomplish this, a Transfer Function (TF) is employed to convert continuous values into binary representations, distinguishing between selected and unselected genes. Among the options within the S-shape family, the sigmoid function is selected as the threshold function (Kalra et al., 2022).

This choice is motivated by its continuous and differentiable properties, clear probabilistic interpretation, and robustness in handling noise and outliers, as observed by (Norfadzlia et al., 2022). Utilizing the sigmoid function serves to guide the behavior of the search agents within the swarm, and its mathematical representation is as Equation (18) and (19):

$$S(x_i^d) = \frac{1}{1+e^{-x_i^d}} \qquad (18)$$

$$x_i^d = \begin{cases} 1, if\ S(x_i^d) > \alpha, \\ 0,\ otherwise \end{cases} \qquad (19)$$

Here, α is a random variable drawn from a uniform distribution between 0 and 1, x belongs to the set of real numbers and represents a potential solution, and d signifies the continuous value of the gene at a specific moment.

Evaluation Metrics

The assessment of the proposed gene selection methods encompasses the following performance metrics:

i. Mean classification accuracy: This metric determines the average classification accuracy by executing the algorithm $'P'$ times and averaging the results.

ii. F1 score: The F1 Score assesses a model's performance by considering both precision and recall, providing a balanced evaluation of its predictive abilities.

iii. Mean number of selected genes: This metric calculates the average count of selected genes after running the algorithm $'P'$ number of times.

In addition to these evaluation measures, the research also presents the standard deviation of the competing methods and conducts the T-test

(William, 1908). These supplementary analyses aim to demonstrate the stability of the competing methods and evaluate potential statistical differences among them.

Parameters and Settings

The performance of SI algorithms relies on hyperparameters. These hyperparameters or configurations are set to customize the algorithm for a specific problem. The choice of appropriate hyperparameter values can significantly impact the quality of the algorithm's solutions. For instance, in addition to factors like swarm size and the number of generations, the PSO algorithm involves settings like acceleration coefficients, which influence how particles move toward their personal or global best positions. These settings are crucial for ensuring effective exploration of the search space and convergence to optimal solutions.

Table 2 provides a comprehensive list of both general and algorithm-specific hyperparameters used in this study, along with their respective values. Algorithm-specific hyperparameters were determined following the work of Too, (2021), while general hyperparameters like swarm size, maximum iterations, fitness function, K value in KNN, number of hidden layers and neurons in MLP, and the number of runs (P) were intuitively selected.

Table 2: Parameters and Settings Utilized

	Hyperparameters	Parameter values
General	Swarm Size	20
	Number of generation/iterations	20
	Train-Test Split	70-30
	$'\alpha'$	0.99
	$'\beta'$	0.01
	Dimension (D)	Gene count in datasets
	K value in KNN	5
	Number of runs (P)	10
	Number of hidden layers (MLP)	32
	Number of neurons (MLP)	8
	Momentum (MLP)	0.90
	Solver (MLP)	Adam
	a (Logistic map)	3.7
	q_i (Logistic map)	0.9
PSO	C_1	2
	C_2	2
	W	0.9
SSA	None	
FA	Alpha	2
	Beta	3.5
	Gamma	4
	Theta	0.97

RESULTS AND DISCUSSIONS

As mentioned in earlier sections of this study, several metrics, including classification accuracy, F1-measure, standard deviation (SD), and the count of selected genes were employed to assess and compare the performance of the competing (proposed) algorithms. It is worth noting that, in all experiments, the study initially performed gene selection by choosing the top-performing 1000 genes using the Mutual Information (MI) statistical tool. This approach aligns with the research of various authors, such as Almugren & Alshamlan, (2019a, 2019b), who have demonstrated that employing a filter method like MI for initial gene selection yields improved performance and reduces computational time compared to using all available genes.

Table 3: Results Obtained from using the Colon Cancer Dataset

Additionally, the outcomes of the experiments presented in this section are reported in the following format: $x \pm y$, where $'x'$ denotes the average classification accuracy or F1-measure (expressed as a percentage), and $' \pm y'$ indicates the SD value. Furthermore, noteworthy results, representing the best performance within each group, are denoted using boldface and underline formatting.

While the gene selection process is often considered a multiobjective optimization challenge, taking into account both classification accuracy/F1 measure and the number of selected genes simultaneously, this study places a higher emphasis on classification accuracy. This emphasis stems from the crucial role of accurate predictive models in applications critical to human life, such as healthcare.

Experimental I: Results and Discussions from the Colon Cancer Dataset

Table 3 provides the results obtained by employing the Colon cancer dataset, featuring statistics such as the count of selected genes, classification accuracy, F1-score, and SD. These findings stem from experiments conducted with both the KNN and MLP learning algorithms. The table is structured to facilitate a vertical reading, with accuracy and its corresponding SD for both KNN and MLP recorded in the table's first section. Meanwhile, the latter portion of the table contains the F1 measure and its associated SD for both KNN and MLP. Notably, only a single count of selected genes is documented for both experiments since genes selected by a specific logistic-chaotic SI algorithm are employed for final classification using both the KNN and MLP learning algorithms.

Analysis of the first section of Table 3, which pertains to accuracy, reveals that the KNN machine learning algorithm consistently outperforms the MLP algorithm in terms of classification accuracy across all three competing algorithms. This superiority of KNN could be attributed to its capacity to capture intricate data relationships, especially when decision boundaries exhibit non-linearity. Additionally, KNN's robustness in handling noisy data may contribute to its superior performance.

Turning attention to the latter section of the table concerning F1-Measure, it is evident once more that KNN excels significantly when compared to the MLP machine learning algorithm across all competing algorithms.

Based on the findings presented in Table 3, it is evident that the KNN algorithm outperforms the MLP algorithm. As a result, KNN has been chosen for further comprehensive comparison and analysis.

Dataset	Algorithms	Accuracy & Standard Deviation (SD)		F1_Measure & Standard Deviation (SD)		No.
		KNN	MLP	KNN	MLP	Genes
Colon Cancer	LC-PSO	**72.10±2.5408**	57.89±4.2989	**81.93±1.3332**	71.09±3.8603	533.60
	LC-SSA	**68.95±1.6634**	62.31±9.9603	**80.28±0.8728**	71.96±3.6583	586.10
	LC-FA	**73.68±0.0000**	59.47±7.8650	**82.76±0.0000**	72.09±5.7557	**476.20**

Table 4 provides a comprehensive overview of the results achieved exclusively through the use of the KNN learning algorithm. The table is designed for vertical interpretation as in Table 3, with the first section showcasing the performance of competing algorithms in terms of accuracy, while the subsequent section shows their performance concerning the F1 measure.

Observing the results, it becomes apparent that, among the three competing algorithms, the Logistic Chaotic-Firefly Algorithm-KNN (LC-FA-KNN) excels across all metrics. This includes accuracy, registering an impressive 73.68%, the F1 measure, with a remarkable 82.76%, and the selection of the least number of genes, totaling 476.20. The superior performance of LC-FA-KNN can be attributed to the unique qualities of the Logistic Chaotic-Firefly Algorithm (LC-FA), which tends to explore the solution space by guiding fireflies towards superior solutions while also accommodating local search around the current best solutions. Additionally, the FA demonstrates robustness against noise and local optima, thanks to its adept balance between exploration and exploitation, enabling it to escape local optima by maintaining population diversity.

Table 4: Comparison of Competing Algorithms Based on Accuracy, F1 Measure, SD, and Number of Selected Genes, using the Colon Cancer Dataset.

Dataset	Algorithms	Accuracy & Standard Deviation (SD)	F1_Measure & Standard Deviation (SD)	No. Features
		KNN	KNN	
Colon Cancer	LC-PSO	72.10±2.5408	81.93±1.3332	533.60
	LC-SSA	**68.95±1.6634**	**80.28±0.8728**	**586.10**
	LC-FA	**73.68±0.0000**	**82.76±0.0000**	**476.20**

Subsequently, a T-test was conducted on the accuracy and F1 measure results of the competing algorithms to assess for statistical distinctions. In this analysis, a p-value below 0.05 indicates the presence of a statistically significant difference among the competing algorithms. Conversely, if the p-value exceeds 0.05 (highlighted in bold), it suggests the absence of a significant difference.

Tables 5 and 6 present the results of the T-test for accuracy and the F1 measure, respectively. From both tables, the p-values calculated for the comparisons between LC-PSO Vs. LC-SSA, as well as LC-PSO Vs. LC-FA, are notably less than 0.05. This signifies the existence of a statistically significant difference between the compared algorithms. However, it's noteworthy that the p-value derived from the comparison between LC-SSA Vs. LC-FA exceeds 0.05. This suggests a lesser degree of statistical significance between the compared algorithms, implying a higher likelihood that the observed results could be attributed to random variation.

Note that the empty cells within the tables indicate redundancy in the comparisons or the unnecessary evaluation of an algorithm against itself.

Table 5: Statistical Analysis Result of Accuracy using The T-Test on the Competing Algorithms Using the Colon Cancer Dataset (p_values > 0.05 are bolded)

	Competing Algorithms			
		LC-PSO	LC-SSA	LC-FA
Competing Algorithms	LC-PSO	-	0.00481102818	**0.08112618885**
	LC-SSA	-	-	8.53805E-06
	LC-FA	-	-	-

Table 6: Statistical Analysis Result of F1 Measure Using the T-Test on the Competing Algorithms using the Colon Cancer Dataset (p_values > 0.05 are bolded)

	Competing Algorithms			
		LC-PSO	LC-SSA	LC-FA
Competing Algorithms	LC-PSO	-	0.004811028	**0.081126189**
	LC-SSA	-	-	8.53805E-06
	LC-FA	-	-	-

https://dx.doi.org/10.4314/swj.v19i3.32

Experimental II: Results and Discussion from the Ovarian Cancer Dataset

Table 7 displays the outcomes of experiments conducted with the Ovarian cancer dataset. Similar to Table 3, various aspects are assessed, including classification accuracy, the F1 measure, the count of selected genes, and the standard deviation (SD). The final classification task is performed using the KNN and MLP algorithms, applied to the gene subset chosen by the specific logistic-chaotic SI algorithm.

Furthermore, the table is organized for vertical reading, with the initial section presenting results related to classification accuracy and its corresponding SD, while the latter section provides the F1 measure and its corresponding SD. As indicated in the prior section, only a single count of the selected genes is presented. This is due to the utilization of the KNN and MLP algorithms to evaluate the quality of genes chosen by the specific algorithm.

Table 7: Results Obtained from using the Ovarian Cancer Dataset

Dataset	Algorithms	Accuracy & Standard Deviation (SD)		F1_Measure & Standard Deviation (SD)		No. Features
		KNN	MLP	KNN	MLP	
Ovarian Cancer	LC-PSO	**91.98±0.9707**	72.90±3.5761	**87.83±1.3998**	57.65±4.7354	603.60
	LC-SSA	**90.66±0.4174**	72.76±2.7783	**85.92±0.9269**	56.88±4.6657	798.70
	LC-FA	**92.24±0.9699**	70.00±12.5863	**88.28±1.7298**	58.77±6.1446	**502.90**

Analyzing the results presented in the first portion of Table 7, it is apparent that the KNN machine learning algorithm once again demonstrates significant superiority over the MLP in all of the considered algorithms. As previously stated in the earlier sections, this enhanced performance of the KNN algorithm can be attributed to various factors, including its proficiency in handling noisy data and its simplicity, among other attributes. Furthermore, a comparison focusing on the F1 measure reaffirms the dominance of the KNN machine learning algorithm, consistently outperforming the MLP across all competing algorithms. Based on the findings presented in Table 7, it is evident that the KNN algorithm outperforms the MLP algorithm. As a result, KNN has been chosen for further comprehensive comparison and analysis.

Table 8 provides a more comprehensive and in-depth examination of the classification accuracy results, the F1 Measure, the SD, and the count of selected genes when utilizing the Ovarian cancer dataset. As previously mentioned, results are exclusively presented for the KNN machine learning algorithm due to its consistent and pronounced superiority over the MLP algorithm. The table maintains a vertical format for ease of interpretation, with the first section illustrating the performance of the three algorithms concerning classification accuracy, while the latter section presents their performance concerning the F1 measure.

Table 8: Comparison of Competing Algorithms Based on Accuracy, F1 Measure, SD, and Number of Selected Genes, using the Ovarian Cancer Dataset.

Dataset	Algorithms	Accuracy & Standard Deviation (SD)	F1_Measure & Standard Deviation (SD)	No. Features
		KNN	KNN	
Ovarian Cancer	**LC-PSO**	91.98±0.9707	87.83±1.3998	603.60
	LC-SSA	90.66±0.4174	85.92±0.9269	798.70
	LC-FA	**92.24±0.9699**	**88.28±1.7298**	**502.90**

Observing the results, it is again clear that among the three competing algorithms, the Logistic Chaotic-Firefly Algorithm-KNN (LC-FA-KNN) consistently outperforms across all metrics. This includes an impressive accuracy rate of 92.24%, an F1 measure score of 88.28%, and the selection of the fewest genes, totaling 502.90.

Following this, a T-test was carried out to evaluate potential statistical differences in the accuracy and F1 measure results among the competing algorithms. Tables 9 and 10 display the outcomes of the T-test for accuracy and the F1 measure, respectively. Both tables reveal that the p-values computed for the comparisons between LC-PSO Vs. LC-SSA and LC-SSA Vs. LC-FA are notably less than 0.05. This indicates the presence of a statistically significant difference between these compared algorithms. However, it's worth noting that the p-value resulting from the comparison between LC-PSO Vs. LC-FA exceeds 0.05. This implies a lesser degree of statistical significance in the comparison between these algorithms, suggesting a higher possibility that the observed results may be attributable to random variation.

Table 9: Statistical Analysis Result of Accuracy using the T-Test on the Competing Algorithm using the Ovarian Cancer Dataset (p_values > 0.05 are bolded)

		Competing Algorithms		
		LC-PSO	LC-SSA	LC-FA
Competing Algorithms	LC-PSO	-	0.001882407	**0.55202282**
	LC-SSA	-	-	0.00046087
	LC-FA	-	-	-

Table 10: Statistical Analysis Result of F1 Measure using the T-Test on the Competing Algorithm using the Ovarian Cancer Dataset (p_values > 0.05 are bolded)

		Competing Algorithms		
		LC-PSO	LC-SSA	LC-FA
Competing Algorithms	LC-PSO	-	0.002483421	**0.529106824**
	LC-SSA	-	-	0.001974217
	LC-FA	-	-	-

In conclusion, combining the logistic map for chaotic initialization with the K-nearest neighbors (KNN) learning algorithm yields improved performance in a firefly optimization process for several reasons. Firstly, the logistic map introduces diversity into the initial solutions, fostering exploration of a broader solution space and aiding in escaping local optima due to its chaotic behavior. This diversity is especially valuable in optimization tasks where the search landscape is complex and multifaceted. Secondly, the KNN algorithm, known for its robustness in handling noisy and complex datasets, guides the optimization process by making informed decisions based on the proximity of solutions. By considering the similarity of solutions in the dataset, KNN can effectively steer the algorithm toward promising regions in the solution space, enhancing the overall convergence of the optimization process. The combination of chaotic initialization and KNN's intelligent guidance leverage their complementary strengths, ultimately leading to improved optimization outcomes. However, the effectiveness of this approach may vary depending on problem-specific characteristics and the careful tuning of algorithm parameters.

Finally, based on the No-Free Lunch theorem, which states that no one optimization algorithm performs best on all datasets, the LC-FA-KNN can be seen as a worthy competitor for high dimensional gene selection problems.

Experimental III: Hyperparameter Tuning of the LC-FA-KNN

Hyperparameter tuning in SI algorithms is crucial to optimize their performance on specific problem domains. These hyperparameters significantly influence the algorithms' convergence speed, exploration-exploitation balance, and overall effectiveness. Tuning hyperparameters involves finding the best combination of settings that adapt the algorithm to the problem at hand.

Different problems may require different hyperparameter configurations to achieve optimal results. Hyperparameter tuning is essential because using inappropriate or default values can lead to suboptimal performance, slow convergence, or even failure to find a solution. Through systematic tuning, SI algorithms can be tailored to the specific characteristics of the optimization problem, improving their efficiency and the quality of solutions they produce.

In light of the preceding experiments, the top-performing algorithm, LC-FA-KNN, is chosen for further investigation. In this experiment, our focus will be specifically on assessing the impact of parameter tuning on its overall performance. Concentration is solely on classification accuracy, the number of selected genes, and the Ovarian dataset, conducting a rigorous analysis to determine how adjustments to key parameters of the algorithm may influence its effectiveness. The parameters above are fine-tuned using their respective values as shown in Table 11. Also, note that the selection of these values is solely based on our prevailing knowledge of the domain. Afterward, a statistical test is conducted on the results to check for statistical significance.

Table 11: Parameters and Values used in Tuning of LC-FA-KNN

	Hyperparameters	Parameter values
Test #1	Swarm Size	30
	Number of generation/iterations	50
Test #1	Swarm Size	50
	Number of generation/iterations	100

Table 12 presents the outcomes derived from experiments conducted with two parameter configurations, specifically (30,50) and (50,100). These experiments highlight the impact of adjusting the swarm size and the number of generations on the performance of LC-FA-KNN. This impact is substantiated by a notable 1.71% improvement in accuracy and an 11.5% reduction in the count of selected genes.

Importantly, it is worth noting that augmenting both the swarm size and the number of generations, transitioning from 30 to 50 and 50 to 100, respectively, directly contributes to enhanced performance. However, this improvement is accompanied by increased computational costs, as the number of search agents and generations escalates, consequently amplifying the number of fitness evaluations required. In future work, a comprehensive computational cost assessment alongside an in-depth examination of classification performance to furnish a more robust report can be conducted

.

Table 12: Accuracy and Number of Genes Result from Fine-Tuning of LC-FA-KNN Parameters

Dataset	Algorithm	Accuracy & Standard Deviation (SD) For (30,50)	# Features	Accuracy & Standard Deviation (SD) For (50,100)	# Features
Ovarian Cancer	LC-FA-KNN	93.16±0.5500	490.40	**93.95±0.6800**	**444.57**

Moreover, it is worth mentioning that while the 0.84% increment from (50,100) to (30,50), i.e., from 93.16 to 93.95, might initially appear to be possibly due to chance, a statistical test on the results was conducted and obtained a p-value of 0.0110. This p-value is below the critical threshold of 0.05, signifying a statistically significant difference between the results, as demonstrated in Table 13.

Table 13: Statistical Analysis Result of F1 Measure using the T-Test on LC-FA-KNN (30,50) Versus LC-FA-KNN (50,100)

Dataset	LC-FA-KNN (30,50) Vs LC-FA-KNN (50,100)
Ovarian Cancer	0.0110132

This experiment underscores the significance of hyperparameter tuning and its capacity to exert a substantial influence on the overall performance of optimization algorithms. In forthcoming research, an effort can be made to undertake further hyperparameter tuning to attain even more favorable results.

Conclusion and Future Research Directions

This research conducted a comprehensive performance comparison of three SI algorithms: Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), and Firefly Algorithm (FA). These algorithms were enhanced by incorporating the logistic chaotic map for initialization within the context of microarray cancer gene selection tasks. The study utilized two cancer datasets, specifically the Ovarian and Colon datasets. The study evaluated their performance using two machine learning algorithms: K-nearest neighbor (KNN) and Multilayer Perceptron (MLP). Significantly, KNN consistently outperformed MLP across all metrics. The findings highlight the remarkable effectiveness of the logistic-chaos Firefly Algorithm when paired with KNN, denoted as LS-FA-KNN. Impressively, LS-FA-KNN consistently demonstrated superior performance among its peers, further improved through parameter tuning, achieving an exceptional overall accuracy rate of 93.95% while selecting 444 genes. This approach firmly establishes itself as a significant performer in gene selection tasks. One significant limitation of this study is its computational cost. In future investigations, there is the need to assess the computational expenses associated with these algorithms, particularly in light of their scalability challenges. As both population size and iteration count increase, the number of fitness evaluations grows substantially, leading to higher computational costs. While the study employed the mutual information filter method for preliminary gene selection, there is a clear need to explore strategies to mitigate these computational burdens. Additionally, it is imperative to explore the potential of other chaotic maps in conjunction with the SI algorithms considered in this research. This approach aligns with the 'No-one-size-fits-all' concept, acknowledging the influence of dataset characteristics and application domains on optimization algorithm performance. Lastly, further exploration of alternative machine learning algorithms, such as logistic regression and random forest, is warranted to uncover potential enhancements in performance.

REFERENCES

Abd-Elnaby, M., Alfonse, M., & Roushdy, M. (2021). Classification of breast cancer using microarray gene expression data: A survey. Journal of Biomedical Informatics, 117(March), 103764.

Abel, H. J., Larson, D. E., Regier, A. A., Chiang, C., Das, I., Kanchi, K. L., Layer, R. M., Neale, B. M., Salerno, W. J., Reeves, C., Buyske, S., Abecasis, G. R., Appelbaum, E., Baker, J., Banks, E., Bernier, R. A., Bloom, T., Boehnke, M., Boerwinkle, E., … Hall, I. M. (2020). Mapping and characterization of structural variation in 17,795 human genomes. Nature 2020 583:7814, 583(7814), 83–89.

Adamu, A., Abdullahi, M., Junaidu, S. B., & Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. Machine Learning with Applications, 6, 100108.

Aljarah, I., Habib, M., Faris, H., Al-Madi, N., Heidari, A. A., Mafarja, M., Elaziz, M. A., & Mirjalili, S. (2020). A dynamic locality multi-objective salp swarm algorithm for feature selection. Computers and Industrial Engineering, 147, 106628.

Almugren, N., & Alshamlan, H. (2019a). FF-SVM: New FireFly-based Gene Selection Algorithm for Microarray Cancer Classification. 2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2019.

Almugren, N., & Alshamlan, H. M. (2019b). New bio-marker gene discovery algorithms for cancer gene expression profile. IEEE Access, 7, 136907–136913.

Alomari, O. A., Makhadmeh, S. N., Al-Betar, M. A., Alyasseri, Z. A. A., Doush, I. A., Abasi, A. K., Awadallah, M. A., & Zitar, R. A. (2021). Gene selection for microarray data classification based on Gray Wolf Optimizer enhanced with TRIZ-inspired operators. Knowledge-Based Systems, 223, 107034.

Alrefai, N., & Ibrahim, O. (2022). Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets. Neural Computing and Applications 2022, 1–16.

Alshareef, A. M., Alsini, R., Alsieni, M., Alrowais, F., Marzouk, R., Abunadi, I., & Nemri, N. (2022). Optimal Deep Learning Enabled Prostate Cancer Detection Using Microarray Gene Expression. Journal of Healthcare Engineering, 2022.

Alzaqebah, M., Briki, K., Alrefai, N., Brini, S., Jawarneh, S., Alsmadi, M. K., Mohammad, R. M. A., ALmarashdeh, I., Alghamdi, F. A., Aldhafferi, N., & Alqahtani, A. (2021). Memory based cuckoo search algorithm for feature selection of gene expression dataset. Informatics in Medicine Unlocked, 24, 100572.

Alzaqebah, M., Jawarneh, S., Mohammad, R. M. A., Alsmadi, M. K., Al-marashdeh, I., Ahmed, E. A. E., Alrefai, N., & Alghamdi, F. A. (2021). Hybrid feature selection method based on particle swarm optimization and adaptive local search method.

Arora, S., & Singh, S. (2017). An improved butterfly optimization algorithm with chaos. Journal of Intelligent and Fuzzy Systems, 32(1), 1079–1088.

Assiri, A. S. (2021). On the performance improvement of butterfly optimization approaches for global optimization and Feature Selection. In PLoS ONE (Vol. 16, Issue 1 January).

Ayham, M., Alhafedh, A., & Qasim, O. S. (2019). Two-Stage Gene Selection in Microarray Dataset Using Fuzzy Mutual Information and Binary Particle Swarm Optimization. January.

Baliarsingh, S. K., Ding, W., Vipsita, S., & Bakshi, S. (2019). A memetic algorithm using emperor penguin and social engineering optimization for medical data classification. Applied Soft Computing Journal, 85, 105773.

Blot, A., Kessaci, M. É., & Jourdan, L. (2018). Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation. Journal of Heuristics, 24(6), 853–877.

Brezočnik, L., Fister, I., & Podgorelec, V. (2018). Swarm intelligence algorithms for feature selection: A review. Applied Sciences (Switzerland), 8(9).

Chakraborty, C., Kishor, A., & Rodrigues, J. J. P. C. (2022). Novel Enhanced-Grey Wolf Optimization hybrid machine learning technique for biomedical data computation. Computers and Electrical Engineering, 99, 107778.

Chaudhuri, A., & Sahu, T. P. (2021). A hybrid feature selection method based on Binary Jaya algorithm for micro-array data classification. Computers and Electrical Engineering, 90(November 2020), 106963.

Chen, Y., Liu, J., Zhu, J., & Wang, Z. (2023). An improved binary particle swarm optimization combing V-shaped and U-shaped transfer function. Evolutionary Intelligence, 1–14.

Da Silva, S. R., & Gertrudes, J. C. (2022). Chaotic Genetic Bee Colony: Combining Chaos Theory and Genetic Bee Algorithm for Feature Selection in Microarray Cancer Classification. GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference, 296–299.

Dabba, A., Tari, A., & Meftali, S. (2021). Hybridization of Moth flame optimization algorithm and quantum computing for gene selection in microarray data. Journal of Ambient Intelligence and Humanized Computing, 12(2), 2731–2750.

Dabba, A., Tari, A., Meftali, S., & Mokhtari, R. (2021). Gene selection and classification of microarray data method based on mutual information and moth flame algorithm. Expert Systems with Applications, 166, 114012.

Dash, S., Thulasiram, R. K., & Thulasiraman, P. (2019). Modified firefly algorithm with chaos theory for feature selection: A predictive model for medical data. International Journal of Swarm Intelligence Research, 10(2), 1–20.

Deng, X., Li, M., Wang, L., & Wan, Q. (2022). RFCBF: Enhance the Performance and Stability of Fast Correlation-Based Filter. International Journal of Computational Intelligence and Applications, 21(2).

Hasri, N. N. M., Wen, N. H., Howe, C. W., Mohamad, M. S., Deris, S., & Kasim, S. (2017). Improved support vector machine using multiple SVM-RFE for cancer classification. International Journal on Advanced Science, Engineering and Information Technology, 7(4-2 Special Issue), 1589–1594.

Houssein, E. H., Gad, A. G., Hussain, K., & Nagaratnam, P. (2021). Major Advances in Particle Swarm Optimization : Theory , Analysis , and Application. Swarm and Evolutionary Computation, 63(February), 100868.

Huda, R. K., & Banka, H. (2019). Efficient feature selection and classification algorithm based on PSO and rough sets. Neural Computing and Applications, 31(8), 4287–4303.

Huda, R. K., & Banka, H. (2020). New efficient initialization and updating mechanisms in PSO for feature selection and classification. Neural Computing and Applications, 32(8), 3283–3294.

Hussain, K., Najib, M., Salleh, M., Cheng, S., & Shi, Y. (2018). Metaheuristic research : a comprehensive survey. Artificial Intelligence Review.

Isuwa, J., Abdullahi, M., & Abdulrahim, A. (2022). Hybrid particle swarm optimization with sequential one point flipping algorithm for feature selection. July, 1–18.

Isuwa, J., Abdullahi, M., Ali, Y. S., Kim, J., Hassan, I. H., & Buba, J. R. (2023). Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent Developments and An Exploratory Study Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent. Egyptian Informatics Journal.

Isuwa, J., Mohammed, A., Yusuf, S. A., Idris, M. N., & Garko, B. S. (2021). Hybridization of Metaheuristic Algorithms with Local Search Methods for Optimizing the Feature Selection Process : A Survey. Iccait.

Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification. Applied Soft Computing, 62, 203–215.

Jati, G. K., & Manurung, R. (2013). 13 Discrete Firefly Algorithm for Traveling Salesman Problem : A New Movement Scheme. In Swarm Intelligence and Bio-Inspired Computation. Elsevier Inc.

Jeremiah, I., Abdullahi, M., Yusuf, S. A., & Hassan, I. H. (2023). Towards an Improved Particle Swarm Optimization for Feature Selection : A Survey. Sule Lamido University Journal of Science & Technology, 6(1), 59–73.

Jeremiah, I., Abdullahi, M., Yusuf, S. A., Nuruddeen Idris, M., Garko, B. S., & Yusuf Haruna, M. (2022). Integrating Local Search Methods in Metaheuristic Algorithms for Combinatorial Optimization: The Traveling Salesman Problem and its Variants. 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), 1–5.

Kalra, M., Kumar, V., Kaur, M., Idris, S. A., Öztürk, Ş., & Alshazly, H. (2022). A novel binary emperor penguin optimizer for feature selection tasks. Computers, Materials and Continua, 70(3), 6239–6255.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1942–1948.

Khurma, R. A., Aljarah, I., & Sharieh, A. (2020). An efficient moth flame optimization algorithm using chaotic maps for feature selection in the medical applications. ICPRAM 2020 - Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, January, 175–182.

Khurma, R. A., Aljarah, I., Sharieh, A., Elaziz, M. A., Damaševičius, R., & Krilavičius, T. (2022). A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem. Mathematics, 10(3), 1–45.

Kumar, M., & Rath, S. K. (2016). Feature Selection and Classification of Microarray Data Using Machine Learning Techniques. In Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology. Elsevier Inc.

Liu, L., Wei, Z. X., & Xiang, H. (2022). A novel image encryption algorithm based on compound-coupled logistic chaotic map. Multimedia Tools and Applications, 81(14), 19999–20019.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. 0, 1–11.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 114, 163–191.

Naseri, A., & Hasheminejad, S. M. H. (2019). An unsupervised gene selection method based on multiobjective ant colony optimization. International Journal of Artificial Intelligence, 17(2), 1–22.

Norfadzlia, M. Y., Muda, A. K., Pratama, S. F., Carbo-Dorca, R., & Abraham, A. (2022). Improved swarm intelligence algorithms with time-varying modified Sigmoid transfer function for Amphetamine-type stimulants drug classification. Chemometrics and Intelligent Laboratory Systems, 226.

Nouri-Moghaddam, B., Ghazanfari, M., & Fathian, M. (2021). A novel bio-inspired hybrid multi-filter wrapper gene selection method with ensemble classifier for microarray data. Neural Computing and Applications 2021 35:16, 35(16), 11531–11561.

Pantic, I., Paunovic, J., Cumic, J., Valjarevic, S., Petroianu, G. A., & Corridon, P. R. (2023). Artificial neural networks in contemporary toxicology research. Chemico-Biological Interactions, 369, 110269.

Qin, X., Zhang, S., Yin, D., Chen, D., & Dong, X. (2022). Two-stage feature selection for classification of gene expression data based on an improved Salp Swarm Algorithm. 19(July), 13747–13781. https://doi.org/10.3934/mbe.2022641

Ravindran, U., & Gunavathi, C. (2023). A survey on gene expression data analysis using deep learning methods for cancer diagnosis. Progress in Biophysics and Molecular Biology, 177, 1–13.

Rupa, C., Harshitha, M., Srivastava, G., Gadekallu, T. R., & Maddikunta, P. K. R. (2023). Securing Multimedia Using a Deep Learning Based Chaotic Logistic Map. IEEE Journal of Biomedical and Health Informatics, 27(3), 1154–1162.

Sharma, A., & Rani, R. (2019). C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. Computer Methods and Programs in Biomedicine, 178, 219–235.

Song, X. fang, Zhang, Y., Gong, D. wei, & Sun, X. yan. (2021). Feature selection using bare-bones particle swarm optimization with mutual information. Pattern Recognition, 112, 107804. https://doi.org/10.1016/j.patcog.2020.107804

Tang, E. K., Suganthan, P. N., & Yao, X. (2005). Feature selection for microarray data using least squares SVM and particle swarm optimization. Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB '05, 2005(January 2005).

Too, J. (2021). Jx-WFST : Wrapper Feature Selection Toolbox. Github Project. https://github.com/JingweiToo/Wrapper-Feature-Selection-Toolbox-Python

Tubishat, M., Idris, N., Shuib, L., Abushariah, M. A. M., & Mirjalili, S. (2020). Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Systems with Applications, 145, 113122.

Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic Salp swarm algorithm for feature selection. Expert Systems with Applications, 164.

Tubishat, M., Ja, S., Alswaitti, M., Mirjalili, S., Idris, N., Akmar, M., & Shah, M. (2021). Dynamic Salp swarm algorithm for feature selection. Expert Systems With Applications, 164(June 2020), 113873.

Utami, D. A., & Rustam, Z. (2019). Gene selection in cancer classification using hybrid method based on Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) feature selection and support vector machine. AIP Conference Proceedings, 2168.

William, S. G. (1908). On student's 1908 article "the probable error of a mean." Journal of the American Statistical Association, 103(481), 1–7.

Woldseth, R. V., Aage, N., Bærentzen, J. A., & Sigmund, O. (2022). On the use of artificial neural networks in topology optimisation. Structural and Multidisciplinary Optimization 2022 65:10, 65(10), 1–36.

Wu, X., & Guan, Z. H. (2007). A novel digital watermark algorithm based on chaotic maps. Physics Letters, Section A: General, Atomic and Solid State Physics, 365(5–6), 403–406.

Yang, J., Liu, Z., Zhang, X., & Hu, G. (2022). Elite Chaotic Manta Ray Algorithm Integrated with Chaotic Initialization and Opposition-Based Learning. Mathematics, 10(16).

Yang, X. (2009). Firefly Algorithms for Multimodal Optimization. 169–178.

Zhang, Y., Deng, Q., Liang, W., & Zou, X. (2018). An Efficient Feature Selection Strategy Based on Multiple Support Vector Machine Technology with Gene Expression Data. BioMed Research International, 2018.

Zouache, D., & Ben Abdelaziz, F. (2018). A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. Computers and Industrial Engineering, 115(October 2017), 26–36.